Topology Optimization with Geometric Cleanability Requirements

Giele, R.J.P.

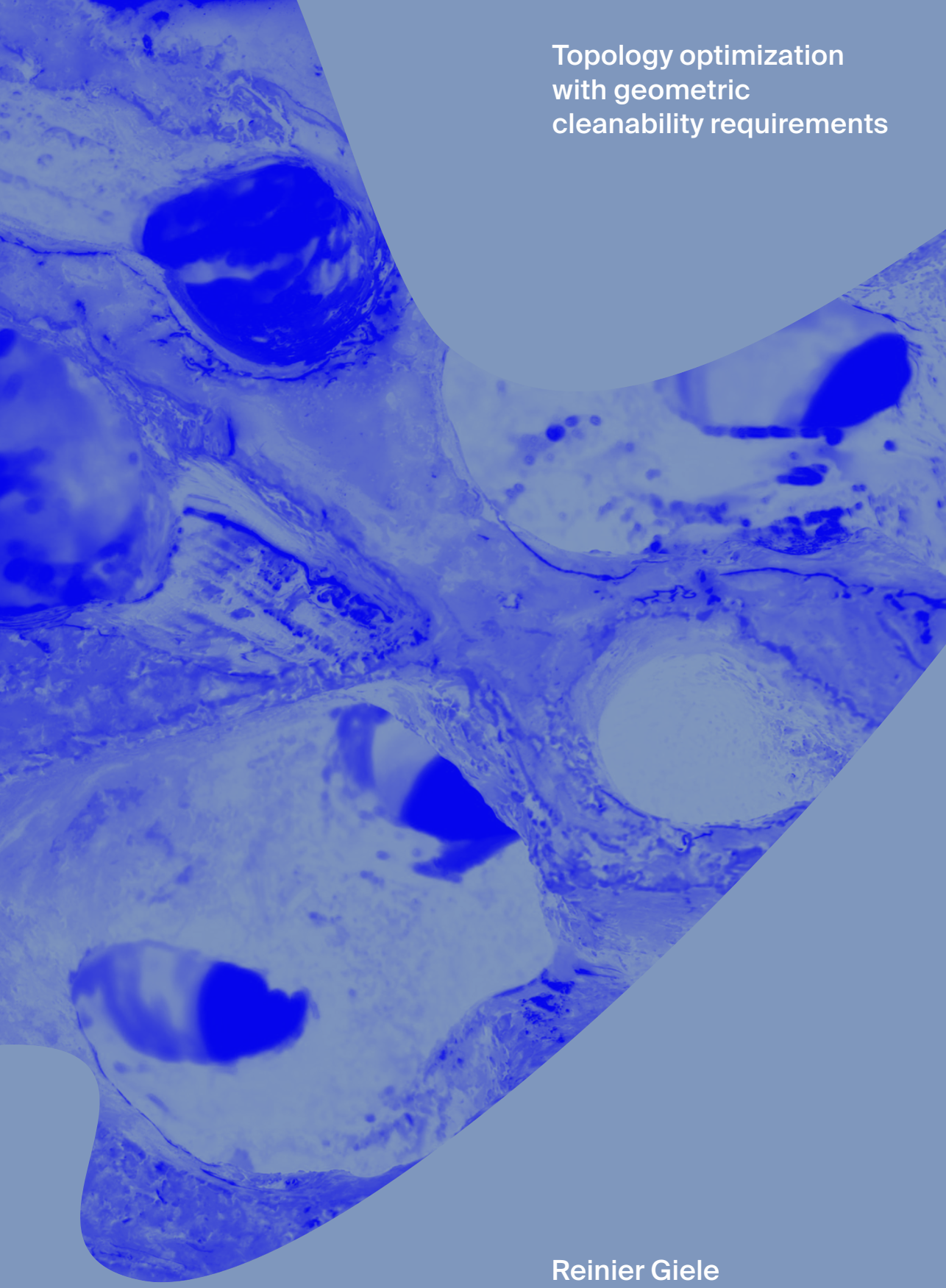
DOI
10.4233/uuid:01d0b8ad-6ee9-4be8-a509-a6e7885ecd1e

Publication date
2025

Document Version
Final published version

Citation (APA)
Giele, R. J. P. (2025). *Topology Optimization with Geometric Cleanability Requirements*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:01d0b8ad-6ee9-4be8-a509-a6e7885ecd1e

Topology optimization with geometric cleanability requirements



Reinier Giele

TOPOLOGY OPTIMIZATION WITH GEOMETRIC CLEANABILITY REQUIREMENTS

TOPOLOGY OPTIMIZATION WITH GEOMETRIC CLEANABILITY REQUIREMENTS

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Friday 21 November 2025 at 12::30 o'clock

by

Reinier Jan Pieter GIELE

Master of Science in Aerospace Engineering, Delft University of Technology, the Netherlands
Master of Science in Wind Energy, Technical University of Denmark, Denmark
born in Delft, the Netherlands

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. ir. M. Langelaar,	Delft University of Technology, *promotor*
Dr. C. Ayas,	Delft University of Technology, *promotor*
Independent members:	
Prof. dr. ir. P. Breedveld	Delft University of Technology
Prof. dr. ir. M. A. N. Hendriks	Delft University of Technology / Norwegian University of Science and Technology, Norway
Prof. dr. A. Pirondi	University of Parma, Italy
Dr. J. V. Carstensen	Massachusetts Institute of Technology, United States of America
Dr. C. S. Andreasen	Technical University of Denmark

An electronic copy of this dissertation is available at
https://repository.tudelft.nl/.

CONTENTS

SUMMARY

This thesis presents several novel methods developed for imposing geometric constraints for cleanability in density based topology optimization (TO). Design for cleanability is a field of growing relevance, and requirements for cleanibility should be taken into account during the design optimization instead of using post-optimization design modifications. Thus far, methods to enable an integral treatments of these requirements in TO have been lacking.

The first method presented ensures that a cleaning fluid can passively run off the surfaces of a part under the effect of gravity alone, which assures drainability. Drainability in a design is obtained by a novel geometric filter that forces upward facing surfaces to have a minimum runoff angle while preventing occurrence of pockets without sink holes.

Secondly, a method is introduced to ensure designs where the entire surface can be cleaned by jetting a cleaning fluid. This is a widespread way of cleaning surfaces in industrial applications. To incorporate the requirement that the entire surface of a part can be accessed by at least one fluid jet, two formulations are considered: one based on jet trajectory lines and another based on front propagation for detecting the access field of individual jets available at predefined positions. The final jettable design is obtained by incorporating a novel filter to TO that ensures the entire surface is accessible by the superposition of the access fields of all jets.

Next, building on the previous method, simultaneous optimization of component design and the jet positions is established. Consistent sensitivities are formulated for the jet positions to enable computationally efficient gradient based optimization. This approach increases the design freedom of the part layout leading to better structural performance. This method is presented as a general framework suitable for any geometric requirement. To illustrate this, in addition to jetting, geometric requirement examples related to additive manufacturing and milling are given. A unique characteristic of the filter in this general framework is that it is disconnected from the finite element mesh, and hence naturally compatible with unstructured meshes.

Finally, building on the concepts developed in the aforementioned method, a new feature mapping TO approach is presented with highly flexible feature shapes, parametrized by NURBS (non-uniform rational B-splines). Similar to existing feature mapping methods, multiple features can be combined to create a final component. The key advantage of the proposed flexible feature representation is that fewer features are needed than in traditional feature mapping techniques. This allows better control over the final topology and outer surface. Regulations on feature shapes must be applied to prevent artifacts such as intersection of features and influence of initialization. However, the proposed method exhibits good control over the component shape, and thus bears great potential

for ensuring cleanability.

Every proposed method is demonstrated and investigated through various numerical examples. All methods succeed in ensuring the targeted geometric requirements. However further improvements are possible, e.g. by modeling the cleaning procedure in a more detailed fashion for closer resemblance to reality, or by simultaneously taking into account several geometric requirements during the optimization.

1

INTRODUCTION

1

Developments in computational design methods and manufacturing techniques offer exiting new opportunities for designers in industry. At the same time, further research is needed for how to best utilize them. This thesis aims to contribute to this process, with particular attention to the aspect of design for cleanability: a vital requirement in many industries that has received remarkably little attention thus far in scientific research. Before defining the research topics addressed in this thesis, first topology optimization and the general concept of cleanability are introduced.

1.1. TOPOLOGY OPTIMIZATION

Topology optimization (TO) is a computational design method to determine the geometric layout of structural components with superior performance. TO is an iterative procedure. In every iteration, the components performance is determined numerically with finite element analysis, and the permitted amount of material is distributed in the most beneficial manner within the design domain, allowing for significant reduction of weight and cost. TO comprises i) geometric representation, ii) performance simulation, and iii) algorithmic optimization. In this thesis some basic TO knowledge is assumed, for a comprehensible TO introduction the reader is referred to Bendsøe and Sigmund [1], or educational articles reviewed by Wang *et al.* [2]. An example of a TO result is shown in Figure 1.1.

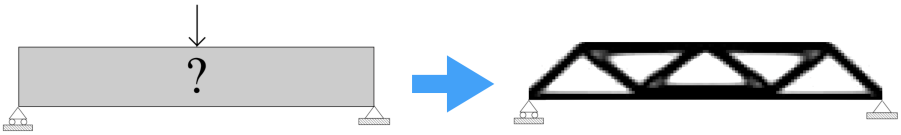


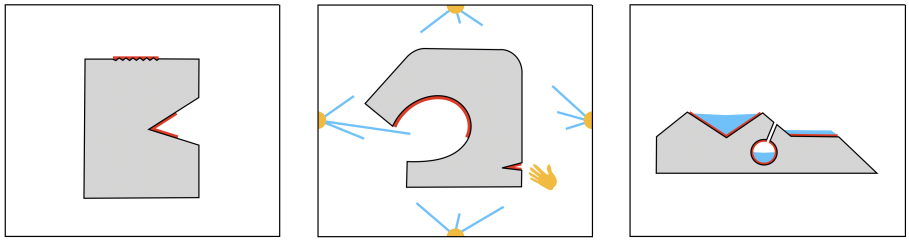
Figure 1.1: Topology optimization (left) in a simply supported design domain under a point load illustrated in and (right) the corresponding optimal distribution of material for maximum stiffness when only 35% of design domain is permitted to consist of solid material.

Additive manufacturing (AM) is a manufacturing method where a part is realized in a layer-by-layer manner on previously manufactured part of the component. This is a promising method, with almost no material waste and the great form freedom it can accommodate. TO and AM are synergistic, complementing each other in strengths and weaknesses. Generally speaking, free-form characteristics of TO can result in design shapes with superior strengths, yet complicated, costly, and even sometimes impossible to manufacture with traditional manufacturing methods. On the other hand, AM can realize most complex design shapes, yet still has some manufacturing constraints of its own which require due account. A strong harmony between the two relies on taking into account limitations of AM during the TO design process. Commonly referred to as design for AM has recently been a flourishing active research field, and an overview is given in Bayat *et al.* [3].

1.2. CLEANABILITY

For several industries, such as food processing, semiconductor production, space, and medical, a specific key requirement is that components should be clean. Thus, design methods for products that are also *cleanable* can have significantly increased impact.

Cleanability of a component design requires considerations of three different stages: before, during, and after the act of cleaning, as illustrated in Figure 1.2. "Before cleaning" entails that contaminants do not easily adhere on surfaces. That means e.g. that smooth surfaces and rounded corners are preferred over rough surfaces and sharp corners. "During cleaning" entails easy access of the cleaning tool/agent to the entire surface. That requires e.g. altering a design such that the outer surface has no unreachable region for fluid jets or cleaning tools. "After cleaning" entails that cleaning fluids and contaminants dissolved or separated from the part surfaces easily leave the component. That means e.g. preventing pockets in the design that can trap fluids and ensuring minimum runoff angles for upward facing surfaces, since preferably this happens passively due to gravity alone.



(a) Before cleaning, e.g. sharp inner corners or rough surfaces.

(b) During cleaning, e.g. inaccessible regions for cleaning tools or jets.

(c) After cleaning, e.g. unfavorable runoff angles and pockets.

Figure 1.2: Design for cleanability should encompass different stages: (a) before, (b) during, and (c) after the cleaning. Uncleanable design features are marked in red.

To the best of the author's knowledge, no research has been done on cleanability in the context of TO. This implies cleanability requirements of the component has not been taken into account during the optimization. TO often leads to more complex designs, which generally does not promote cleanability. Therefore, structurally superior TO designs currently have to be altered posteriori to adhere to cleanability requirements, likely undoing the TO benefits. Providing TO methods that account for cleanability therefore has the potential to make significant impact.

1

1.3. THIS THESIS

This thesis aims to close the gap between TO and the design of components with cleanability requirements. We will explore various novel additions to conventional density based TO, to account for cleanability during the optimization process.

In the remainder of this chapter, the identification of the specific research topics will be described. This identification is an unchartered territory, since little scientific research has been done on cleanability. Cleanability of components is often examined in a case-specific industrial setting, rather than general. Therefore, a first requirement for the research focus, is that methods are applicable/useful in a broad range of applications. A second requirement for the research topics, is that the methods proposed should be addressed in combination with TO. This is why we focus on the overall component design, rather than e.g. its surface roughness. Several relevant research topics related to component design are listed below.

A first research topic that was identified, is how to eliminate sharp inner corners, as displayed in Figure 1.3. Preventing this requires geometric control over inner corners, which from a TO perspective, is fairly well established. For instance, general length scale control is discussed in detail in Wang *et al.* [4]. Length scale control can be applied to inner corners using the 'open' filter from Sigmund [5]. Furthermore, curvature control was studied in Bartz *et al.* [6]. Consequently, this topic is therefore not addressed in this work.

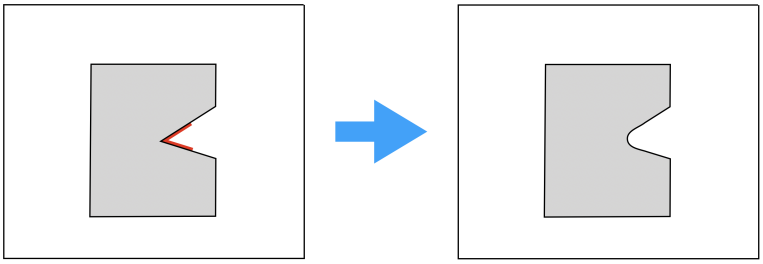


Figure 1.3: A component with (left) sharp inner corners where contaminants are difficult to remove and (right) rounding of the sharp corner to improve cleanability.

A second topic is (cleaning) fluids ability to leave the component after the cleaning. This is always possible with effort, such as vacuum cleaning or shaking, however it is desirable for this to happen passively under the sole effect of gravity. This is referred to as *drainability*. For the component's design, this implies 1) upward facing surfaces should have a minimum runoff angle, and 2) no pockets should exist where the fluids are trapped. This is depicted in Figure 1.4 and will be covered in Chapter 2. In here, we also point out several similarities between design for drainability and design for additive manufacturing.

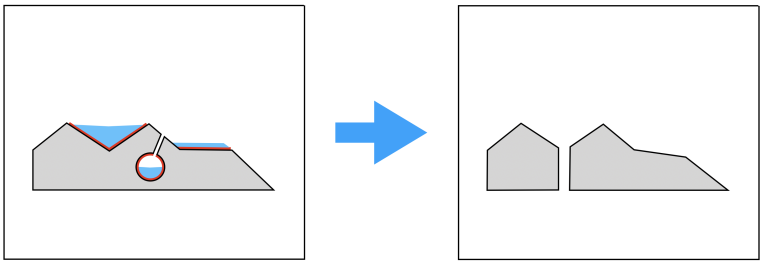


Figure 1.4: Easy removal of cleaning fluids. A component with (left) shallow runoff angles and pockets where fluids will remain on the component after cleaning and (right) minimum runoff angles satisfied and without pockets.

Thirdly, access of cleaning with fluid jets is considered as a research topic. This requires the entire component's surface to be reachable by at least one jet, as shown in Figure 1.5. Cleaning with jetting is often used in industry, and thus jet accessibility is a frequent design requirement. A method that can ensure this during TO is expected to have broad industrial adoption. This topic is covered in Chapter 3. Also here several similarities with methods related to design for milling and additive manufacturing were identified and investigated.

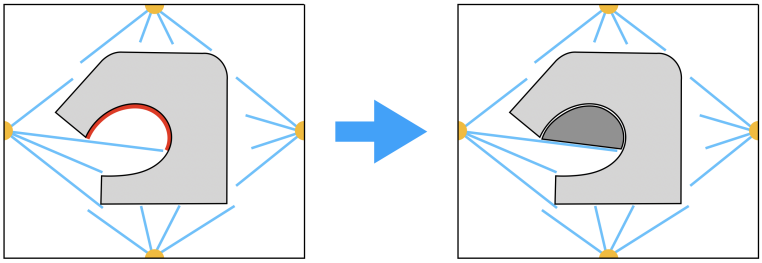


Figure 1.5: Access for jets to reach the entire component's surface. The figure shows a design with unjettable surfaces with red (left) and a design with the entire surface jettable (right).

For the fluid jet access discussed in Chapter 3, jets with a predefined position are considered. This can impose a strong restriction on the design freedom of the component, even though there might actually be some flexibility in the jet positions. Accounting for jet position as an extra design variable freedom leads to an increase in the computational design space, and consequently can improve the structural performance of the design, as shown in Figure 1.6. Simultaneous optimization of structure and jet positions is considered in Chapter 4. Although this may appear as a small conceptual difference, optimizing jet positions required a fundamental

reformulation of the approach.

Furthermore, a general geometric filtering methodology is proposed in Chapter 4, advocating that all geometric requirements can be handled with a similar philosophy within this framework. To illustrate this aspect of wider applicability, examples are given in design for additive manufacturing and milling.

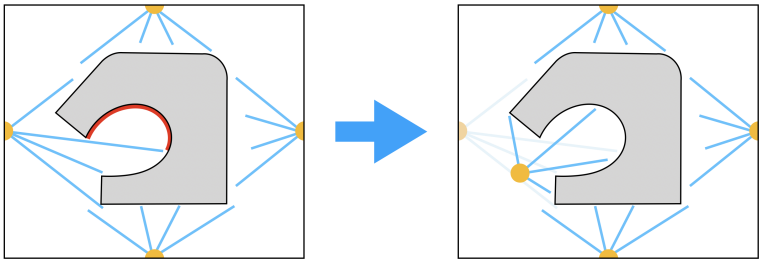


Figure 1.6: Arbitrary jet locations (left) that cannot clean the entire surface of the component and (right) optimized jet locations with complete surface access.

The method proposed in Chapter 4, with sets of points resulting in projections onto the finite element mesh, shows potential to also be applicable in other settings where TO with a degree of shape control is needed. More specifically, a novel feature mapping TO method is developed that is successful in creating features with highly flexible shapes, as illustrated in Figure 1.7.

This feature mapping method allows for design of a component to be represented with much fewer features compared to existing feature based TO schemes e.g. Norato *et al.* [7]. Consequently, control over the surface and the topology are notably improved. The author expects that this improved control over the outer surface will eventually prove useful for ensuring cleanability, for instance in cases where the cleaning process relies on specifically shaped tool. While the steps for establishing a first version of this method are discussed in Chapter 5, several directions for further research are also provided.

As a final remark, please note that Chapters 2-5 consist of scientific articles. Therefore, they can largely be read independently, although the provided order is most logical. Also there will inevitably be some repetition between the chapters.

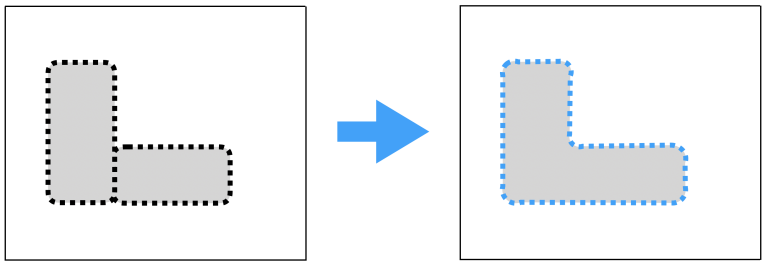


Figure 1.7: Components created from two features with fixed shape (left), and a single feature component with flexible shape (right). Using one feature potentially allows more control over the feature's shape and topology.

2

DRAINING

Topology optimization methods improve the structural performance of components. However, in food processing, medical, high-precision, and other industries' designs should also fulfil the requirement of being cleanable. An important aspect of cleanability is drainability, which entails that fluids can always run of the structure under just gravity. Therefore, taking drainability into account during the optimization process is essential for many applications. This chapter proposes a drainage filter that turns a blueprint design into a drainable design. In a layer-by-layer fashion, the design is adjusted, to ensure fluids can always run down its surface. A smooth minimum and maximum are used in the formulation to allow for consistent sensitivity calculation. To allow for the small runoff angles, typical for practical drainability, a grid refinement is proposed. Moreover, any drainage direction can be considered. The effectiveness of the method is illustrated in 2D and 3D.

This chapter is based on Giele *et al.* [8].

2.1. INTRODUCTION

Topology optimization techniques enable designers to generate structures with superior mechanical performance. The complex designs created with topology optimization can however require intense postprocessing, which can undo the optimization gains. This can be prevented by using constraints reflecting the real practice during the optimization process. Sectors such as the food processing, medical, and high-precision industry, have a design requirement in common for their structural components: they have to be cleanable. The cleanability of a component is directly related to its geometry, as well as the applied cleaning procedures.

One essential aspect of cleanability is to ensure that the cleaning fluid can leave the component. If cleaning fluid remains, dirt can be collected and the component is unfit for usage. Although active methods can be used (reorienting, vibrating, air flow, etc.), often passive methods are preferred. In this context, we focus on the requirement that cleaning fluids can leave the component by running off under gravity: the structure has to be *drainable*, which is illustrated in Figure 2.1. Firstly, drainability entails that the inclination of surfaces should be such that fluids run off under gravity. This implies that upward facing surfaces must make a minimum angle with the horizontal: the *runoff angle*. Our partners have indicated that 10° is a typical guideline for the runoff angle in the food processing industry (R Deckers, personal communication, 12 January, 2021). Secondly, the geometry should be such, that no pockets with stagnant fluids can exist. This way, fluids will be able to run down and leave the structure. Having a method that can ensure drainable designs, by respecting the two above aspects, would greatly improve the applicability of topology optimization in industries where cleanability is important.

(a) Not drainable (b) Drainable

Figure 2.1: A 2D static part in a fixed orientation is considered, with gravity acting downwards. Design (a) is not drainable, because of the pocket and the flat surface area prevent that the fluid (denoted in blue) will run down. Design (b) is drainable, implying that all fluid runs off.

Within structural optimization, to the best knowledge of the authors, no research has been done on creating drainable structures. Yet, there is related research to consider. Firstly, there are approaches focusing on optimizing fluid flow through structures, see Alexandersen and Andreasen [9] for an overview. Given the excessive computational effort requirement which is involved in simulating a fluid flow over a structure, a geometric approach, which typically involves modest computational costs, for evaluating drainability is more appealing. In the context of powder

evacuation in additive manufacturing (AM), the outflow of material from a structure is also considered by Gaynor and Johnson [10]. This method ensures the existence of pathways to evacuate the powder from the structure, which can also be used to provide drainability. The aforementioned authors note that dealing with shallow angles, such as the runoff angles of interest here, is challenging.

Taking inspiration from the latter study, for drainability it is helpful to focus on controlling the void regions in the domain, since there the fluid can reside. In order to ensure that fluid can run off, every void region must be connected to a void region beneath it. The runoff angle defines the maximum horizontal offset. It is assumed that the fluid can always run off at the bottom (and optionally at the sides) of the design domain.

The requirements for drainability have some interesting similarities with those for overhang control used for AM. For a drainable design, void regions need to have sufficient drainage regions below, while for a printable design, solid regions need to have sufficient material below. And where drainability involves a minimum runoff angle, printability is linked to a critical overhang angle. Furthermore, while fluid can drain off at the bottom of the domain, this is also where the printing of solid material can start. These similarities are illustrated in Figure 2.2. Therefore, works in the field of AM overhang control can also serve as inspiration.

Topology optimization approaches for overhang control have been extensively studied, and for a comprehensive overview the reader is referred to Liu *et al.* [11]. Two relevant categories of overhang control methods are: local boundary control and geometrical AM process modelling. The local boundary control methods constrain the angle between the surface normal and the build direction Qian [12] and Allaire *et al.* [13]. These methods require additional care to prevent the creation of saw-tooth like structures. Therefore, the main inspiration for the proposed drainage approach comes from geometrical AM process modelling. Examples of these methods are presented in Gaynor and Guest [14] and Langelaar [15] with a layer-by-layer approach, or in van de Ven *et al.* [16] with a front propagation approach.

For this research, the focus is on a density-based topology optimization approach, in a structured mesh setting. Furthermore, it is preferred to work with a filter instead of adding a constraint term to the objective function, in order to achieve strict enforcement of the drainability requirement. Therefore, the filter from Langelaar [17] is taken as the main inspiration for our proposed drainage method, but we switch the focus to the void. Furthermore, an extra refinement is introduced in order to accommodate the shallow runoff angle requirements.

This chapter presents an approach that ensures drainable designs in 2D and 3D density-based topology optimization. A filter is proposed that converts a given blueprint design into a drainable design. The performance of this drainable design is subsequently evaluated and is optimized. The runoff angle is adjustable by using a refined grid in vertical direction. Draining always works in the direction of gravity, but this direction may not necessarily equal the vertical direction in the topology optimization domain. Therefore, a step is added to accommodate any orientation of the component with respect to gravity. Since the filter procedure adjusts the design

(a) Printable design, with a 45° overhang angle

(b) Drainable design, with a 45° runoff angle

Figure 2.2: Comparison between designs for AM and drainage, where blue represents void and grey represents material. A printable design is determined by placement of the solid and the overhang angle (a). Similarly, a drainable design is determined by the void and the runoff angle (b).

variables to a drainable design in a separate procedure, the proposed method can be used in combination with other methods. As an example, the proposed drainage filter is combined with an AM filter.

The rest of the chapter is organized as follows. Section 2.2 presents the drainage filter method. The optimization problem used for the numerical examples is described in Section 2.3. In Section 2.4, the performance of the drainage design method are demonstrated on two minimum compliance design problems. Finally, the findings are summarized and discussed in Section 2.5.

2.2. METHOD

The discussion in this section is focused on 2D for clarity, the steps required to extend to 3D will be indicated. Section 2.2.1 introduces the method for a runoff angle of 45°. The filtering steps for runoff angle control, drainage direction control, and length scale control, are given in Section 2.2.2 till 2.2.4. The full filter procedure is given in Section 2.2.5.

2.2.1. DRAINAGE FILTER

As stated in the Introduction, in order to facilitate the drainage of fluid, every void region should have some void region below where fluid can run off. Otherwise, either the void region should be transformed into solid so that no fluid can reside there, or a solid region below should be transformed into void so fluid can be drained. In our drainage filter $\mathscr{D}$, we follow the first strategy and the void region without void below in the blueprint design $\mathbf{x}$ is transformed into solid in the drainable design $\hat{\mathbf{x}}$. This process is executed layer-by-layer, sweeping upwards through the domain. This will ensure that undrainable regions are banned from the design. The filter is

defined on a regular Cartesian mesh, with the conventions shown in Figure 2.3. For unstructured meshes, mapping on a structured field is possible, see e.g. Hoffarth *et al.* [18] and Langelaar [19].

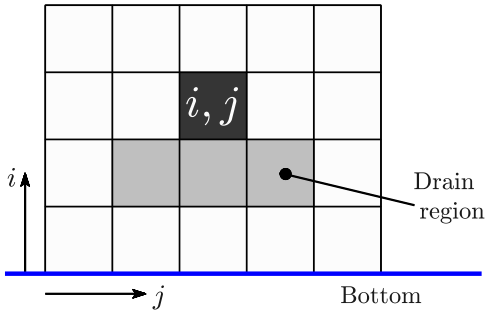
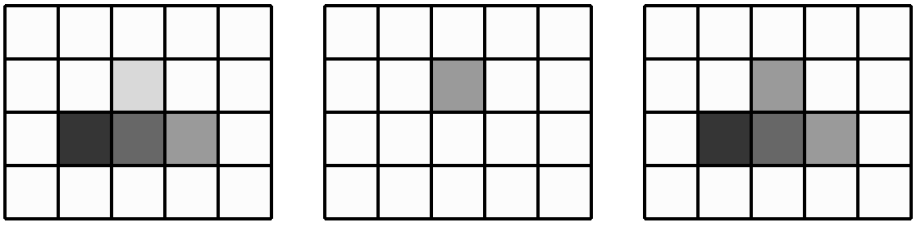


Figure 2.3: The used conventions for 2D, in which the three elements below element (i, j) are the drain region for element (i, j). It is assumed that the bottom of the domain is void, so the fluid can always run off here. For an extension from 2D to 3D, instead of using the 3 elements below, a drain region of 5 elements below element (i, j) is used.

In a continuous density variable setting, an element with blueprint density $x_{(i,j)}$, fulfills the drainability requirement if at least one of the adjacent elements below is has a lower density. Therefore, the lowest density value of the elements in the drain region of element (i, j), denoted with $x^{dr}_{(i,j)}$, is to be found, which is done with a minimum operator. If this density $x^{dr}_{(i,j)}$ is not lower than the density $x_{(i,j)}$, the latter inherits the value of the former, for which a maximum operator is used. This procedure can be seen in Figure 2.4. Note that this bears similarity with the AM filter of Langelaar [17], but drainage requirements lead to a reversed ordering of the min and max operations.



1: Blueprint design x | 2: Minimum of drain region x^{dr} | 3: Filtered design $\hat{x}$, max of (1) and (2)

Figure 2.4: The drainage filtering procedure, which changes the design layer-by-layer from bottom to top, using min and max operations.

Because these steps can change element densities, which themselves are part of the drain region for the row above, this procedure should happen in a layer-by-layer

fashion from bottom to top. The newly obtained drainage field $\hat{\mathbf{x}}$ densities are considered for the drainage region, rather than the blueprint densities $\mathbf{x}$:

$$\begin{aligned} x^{dr}_{(i,j)} &= \min(\hat{x}_{(i-1,j-1)}, \hat{x}_{(i-1,j)}, \hat{x}_{(i-1,j+1)}) \\ \hat{x}_{(i,j)} &= \max(x_{(i,j)}, x^{dr}_{(i,j)}). \end{aligned} \tag{2.1}$$

Because we are considering gradient based topology optimization, the min/max operations have to be replaced with a smooth operation. Material added on top of a structure to ensure drainability, does often not have the most favorable location for an optimization objective. But since it is required to ensure drainability, it should be prevented that the density becomes lower towards the top because of the accumulation of approximation errors. Therefore, it is preferred to over-approximate the density rather than to under-approximate. Appropriate approximation choices are the P-mean (P_{mean}) for the minimum, and the P-norm (P_{norm}) for the maximum:

$$\begin{aligned} P_{mean}(x) &= \left(\frac{1}{N} \sum_{e=1}^{N} (x_e)^{P_m} \right)^{\frac{1}{P_m}} \\ P_{norm}(x, x^{dr}) &= \left(x^{P_n} + (x^{dr})^{P_n} \right)^{\frac{1}{P_n}}. \end{aligned} \tag{2.2}$$

In here, N represents the number of elements in the drain region, i.e. 3 in 2D. The 3D implementation could have a drain region of either 5 or 9 elements, where the former is more conservative and ensures that the runoff angle cannot be violated. In this chapter we therefore use 5 elements, namely all elements in the layer below that share one or more edges with the element on top. A negative P_m should be used to get the minimum. Written out, the approximations used for the drainage filter $\mathscr{D}$ are given by:

$$\begin{aligned} x^{dr}_{(i,j)} &= \left(\left(\hat{x}^{P_m}_{(i-1,j-1)} + \hat{x}^{P_m}_{(i-1,j)} + \hat{x}^{P_m}_{(i-1,j+1)} \right) / 3 \right)^{1/P_m} \\ \hat{x}_{(i,j)} &= \left(x^{P_n}_{(i,j)} + (x^{dr}_{(i,j)})^{P_n} \right)^{1/P_n}. \end{aligned} \tag{2.3}$$

The used aggregation parameters are taken as: $P_m = -40$ and $P_n = 40$. For elements at the sides of the domain, a 'ghost element' is added to represent the outside of the domain. If this is given a density of 0, it implies that fluids can run off at this side of the domain. Furthermore, the blueprint design in the top row of the domain can not be transformed into a drainable design, because this row is not a drainage region itself for the region above the domain. Therefore is should be prevented that this is a solid region, e.g. with a local volume constraint. Finally, if computational overflow occurs because the input of the approximation functions is too small, this can be resolved by introducing a small offset on the input values.

No full derivation of the sensitivity analysis is presented here, as it proceeds along the same lines as described in Langelaar [17]. Only the order of the min/max operations and the used approximation functions differ, which will be given here. The sensitivities from a performance criterion f in row i can be calculated with:

$$\frac{\partial f}{\partial \mathbf{x}_i} = \boldsymbol{\lambda}_i^T \frac{\partial P_{norm,i}}{\partial \mathbf{x}_i}, \tag{2.4}$$

in which $\boldsymbol{\lambda}$ is the multiplier vector, which is computed with the multiplier vector from the row above:

$$\boldsymbol{\lambda}_i^T = \left(\frac{\partial f}{\partial \hat{\mathbf{x}}_i} + \boldsymbol{\lambda}_{i+1}^T \frac{\partial P_{norm,i+1}}{\partial \hat{\mathbf{x}}_i} \right) \frac{\partial P_{norm,i}}{\partial \mathbf{x}_i}. \tag{2.5}$$

The derivatives of P_{norm} and P_{mean} are as follows:

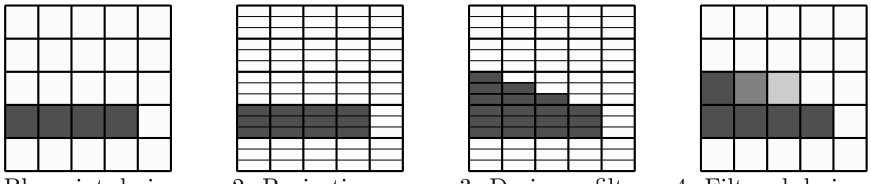
$$\begin{aligned} \frac{\partial P_{norm}(x, x^{dr})}{\partial x} &= \left(x^{P_n} + (x^{dr})^{P_n} \right)^{\frac{1}{P_n}-1} x^{P_n-1} \\ \frac{\partial P_{norm}(x, x^{dr})}{\partial x^{dr}} &= \left(x^{P} + (x^{dr})^{P_n} \right)^{\frac{1}{P_n}-1} (x^{dr})^{P_n-1} \\ \frac{\partial P_{norm}}{\partial \hat{x}} &= \frac{\partial P_{norm}}{\partial x^{dr}} \frac{\partial P_{mean}}{\partial \hat{x}} \\ \frac{\partial P_{mean}(\hat{x}_1, \hat{x}_2, \hat{x}_3)}{\partial \hat{x}} &= \frac{1}{N} \left(\frac{1}{N} \sum_{e=1}^{N} (x_e)^{P_m} \right)^{\frac{1}{P_m}-1} x^{P_m-1}. \end{aligned} \tag{2.6}$$

In here, i and j are left out for readability. Note that the sensitivities are calculated in a layer-by-layer manner from top to bottom, with the topmost multipliers equal to zero.

2.2.2. RUNOFF ANGLE CONTROL

For the method described in Section 2.2.1, the runoff angle can be calculated with $\phi = \text{atan}(\frac{l_y}{l_x})$, where l_x and l_y represent the element size. For a structured grid with $l_y = l_x$, this results in a runoff angle of 45°. The runoff angle can become smaller if $l_y < l_x$.

In the proposed method a smaller runoff angle is obtained by projecting the values of original grid on a grid refined in the drainage direction. The drainage filter is applied on this refined grid, after which the drainable design is projected back to the original grid. This procedure is shown in Figure 2.5. It is important to note that the refined grid is only used for creating the drainable design with the filter and not for the finite element analysis, so no remeshing is performed.



1: Blueprint design x | 2: Projection on refined mesh | 3: Drainage filter | 4: Filtered design $\hat{x}$, projected back

Figure 2.5: Grid refinement procedure for shallow runoff angles. In this example the runoff angle is $\tan^{-1}(1/3) = 18°$.

We choose the refined grid to have size $l_{y,r} = \frac{l_y}{n}$, in which refinement factor n is an integer, so that n refined cells fit exactly in one element of the original grid.

Thus, a refinement of $n = 5$ gives a runoff angle of 11°, which suffices for common drainability requirements in industry.

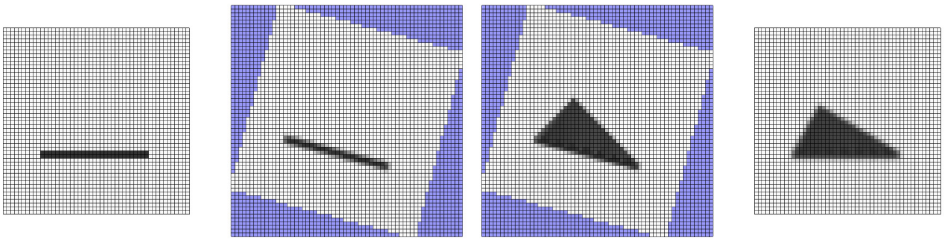
For the forward projection $\mathscr{R}_f$, from original to refined grid, all refined cells inherit their value from the original grid. For the backward projection $\mathscr{R}_b$, from the refined to the original grid, the original elements densities are set to the average value of the corresponding refined elements. For an original grid $\mathbf{x}_o$ and refined grid $\mathbf{x}_r$, these operations can be described as follows:

$$\begin{aligned} x_r &= \mathscr{R}_f(x_o) = x_o \quad \text{for} \quad \Omega(x_r) \subseteq \Omega(x_o) \\ x_o &= \mathscr{R}_b(x_r) = \sum_{i=1}^{n} x_{r(i)}/n \quad \text{for} \quad \Omega(x_r) \subseteq \Omega(x_o), \end{aligned} \tag{2.7}$$

in which Ω represents the (sub)cell volume. Finally, note that for a more specific pick in runoff angles, it would also be possible to do projection on a grid in which the refined cell is not exactly fitted with integer n in an original cell.

2.2.3. DRAINAGE DIRECTION CONTROL

Structural components can have any orientation while being used. However, the gravity that induces the drainage always works downwards. This motivates the need to ensure drainability from multiple specified orientations. This is achieved with a projection on another grid, similar to the approach used for the runoff angle. First, the values of the original grid are projected with a rotation on another grid, which we call the rotation grid. The drainage filter, that creates a design for straight downward drain orientation, is applied on this rotated design field. Finally, the values of the drainable design are projected back to the original grid. This procedure is shown in Figure 2.6.



1: Original design 2: Rotation grid 3: Drain filter 4: Drainable design

Figure 2.6: Rotation procedure for drainage in variable orientations, shown for a rotation 15°. The elements that are not covered by the original grid are represented in blue.

In our application, the rotational projection is performed around the center point of the original mesh. The cell size of the rotation grid is the same as the original grid, but the amount of cells is increased to fully cover all elements of the original grid. The projection is done with a density filter operation with an integrated coordinate

transformation. For an original density field $\mathbf{x}_o$ and a mapped field $\mathbf{x}_m$, the forward mapping filter $\mathscr{M}_f$ and backward mapping filter $\mathscr{M}_b$ are defined as follows:

$$\begin{aligned} x_m &= \mathscr{M}_f(x_o) = \frac{\sum_{o\in N_{m,o}} x_o\, w(\mathbf{c}_o, \mathbf{c}_m)}{\sum_{o\in N_{m,o}} w(\mathbf{c}_o, \mathbf{c}_m)} \\ x_o &= \mathscr{M}_b(x_m) = \frac{\sum_{m\in N_{o,m}} x_m\, w(\mathbf{c}_m, \mathbf{c}_o)}{\sum_{m\in N_{o,m}} w(\mathbf{c}_m, \mathbf{c}_o)}, \end{aligned} \tag{2.8}$$

where $N_{m,o}$ is the neighborhood set of elements within the filter domain for element m, and $w(\mathbf{c}_o, \mathbf{c}_m)$ is the linear weight function between two cells with central coordinates $\mathbf{c}_o$ and $\mathbf{c}_m$:

$$w(\mathbf{c}_m, \mathbf{c}_o) = R_m - \|M(\mathbf{c}_m) - \mathbf{c}_o\|, \tag{2.9}$$

with R_m the specified filter radius, and where the coordinates of $\mathbf{c}_m$ are first rotated by rotation matrix M. The filter radius R_m is picked as small as possible, but big enough to ensure that all elements in the rotation grid that are covered by the original grid have an input. Limited additional smoothing of the design is induced, as is demonstrated in the numerical examples (Section 2.4). The elements in the rotation grid that are not covered by the original grid get assigned a value of 0, which represents drainable space.

2.2.4. LENGTH SCALE CONTROL

Small sinkholes can be created inside the structure, sometimes only the size of one element. In particular in 2D with large runoff angle this was observed. According to the model the fluid can run off through these sinkholes, however for practical usage a minimum length scale on the void is required. The robust method Wang *et al.* [4], Sigmund [20], and Lazarov *et al.* [21] creates a length scale on both the solid and the fluid, and is summarized here.

First, the structural domain density field $\mathbf{x}$ is filtered using the standard density filter $\mathscr{F}$ Bourdin [22] and Bruns and Tortorelli [23], given by:

$$\tilde{x} = \mathscr{F}(x) = \frac{\sum_{l\in N_{k,l}} x_l\, w(\mathbf{c}_l, \mathbf{c}_k)}{\sum_{l\in N_{k,l}} w(\mathbf{c}_l, \mathbf{c}_k)}, \tag{2.10}$$

where $N_{k,l}$ is the neighborhood set of elements within the filter domain for element k, and $w(\mathbf{c}_l, \mathbf{c}_k)$ is the linear weight function between two elements k and l:

$$w(\mathbf{c}_l, \mathbf{c}_k) = R - \|\mathbf{c}_l - \mathbf{c}_k\|, \tag{2.11}$$

with R the specified filter radius. $\mathbf{c}_k$ and $\mathbf{c}_l$ contain the central coordinates of the design elements k and l respectively. Next, a smooth Heaviside projection $\mathscr{H}$ is used:

$$\bar{x} = \mathscr{H}(\mathscr{F}(x)) = \frac{\tanh(\beta\eta) + \tanh(\beta(\tilde{x} - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))}, \tag{2.12}$$

where η is the threshold value, and β the steepness of the projection. Using three threshold values results in *eroded*, *blueprint*, and *dilated* designs, with corresponding

subscripts e,b,d, and threshold values $\eta_e < \eta_b < \eta_d$. The offset parameter $\Delta\eta$ is used to obtain the erode and dilate threshold values: $\eta_e = \eta_b - \Delta\eta$ and $\eta_d = \eta_b + \Delta\eta$. By basing the optimization on the worst case performance of the three designs, robustness against boundary variations is achieved.

This added robustness might result in a bigger length scale than initially intended for the structure. However in general, robustness is in line with creating designs for cleanability, which requires avoiding tiny design features. The relationship between the parameters and the imposed length scale is explained in Trillet *et al.* [24]. Finally, note that the robust filter step is mostly needed for problems where material is scarce or drainage solutions are limited, which is often the case in 2D, especially for high run-off angles. In these cases, the robust formulation helps to prevent designs in which the optimizer exploits intermediate densities. In more realistic cases, i.e. 3D and with lower runoff angles, this is not required as the optimizer has more freedom to create a drainable design. Therefore, in 3D applications only the density filter $\mathscr{F}$ is used and the Heaviside filter step $\mathscr{H}$ can be left out.

2.2.5. FULL FILTER PROCEDURE

All the steps in the drainage filter procedure are shown in Figure 2.7. Only the drainage filter $\mathscr{D}$ is essential, all other filters are optional and depend on the specific problem requirements. Refinement filter $\mathscr{R}$ is needed for runoff angles smaller than 45°, which holds for most practical cases. Rotation filter $\mathscr{M}$ is needed for drainage orientations other than straight downwards. Heaviside filter $\mathscr{H}$ in combination with the robustness offset is effective in preventing small sinkholes, which may otherwise occur for complex problems. This step is done before the drainage filter in order to achieve a bigger variance in the input variables of the smooth min/max approximations. This also means that the subsequent steps are done for all of the eroded, blueprint, and dilated designs.

2.3. OPTIMIZATION FORMULATION

The optimization problem is formulated based on the design variable field $\mathbf{x}$. The drainable density field $\hat{\mathbf{x}}$ is obtained by performing the drainage filter steps on design field $\mathbf{x}$, as outlined above and shown in Figure 2.7. As the focus is on the demonstration of the performance of the drainability filter, numerical examples will involve minimum compliance problems with a global volume constraint, defined as:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}}: \quad & C(\hat{\mathbf{x}}) = \mathbf{u}^T \mathbf{K}(\hat{\mathbf{x}})\mathbf{u} \\ \text{subject to}: \quad & \mathbf{K}(\hat{\mathbf{x}})\mathbf{u} - \mathbf{p} = \mathbf{0} \\ & \frac{V(\hat{\mathbf{x}})}{V^*} - 1 \leq 0 \\ & 0 \leq x_i \leq 1 \quad \text{for} \quad i = 1...N_{el}, \end{aligned} \tag{2.13}$$

in which $\mathbf{K}(\hat{\mathbf{x}})$ is the stiffness matrix, $\mathbf{p}$ are the nodal loads, and $\mathbf{u}$ are the resulting nodal displacements. $V(\hat{\mathbf{x}})$ is the volume of the current design, and V^* is the maximum allowed volume. N_{el} is the number of elements in the domain.

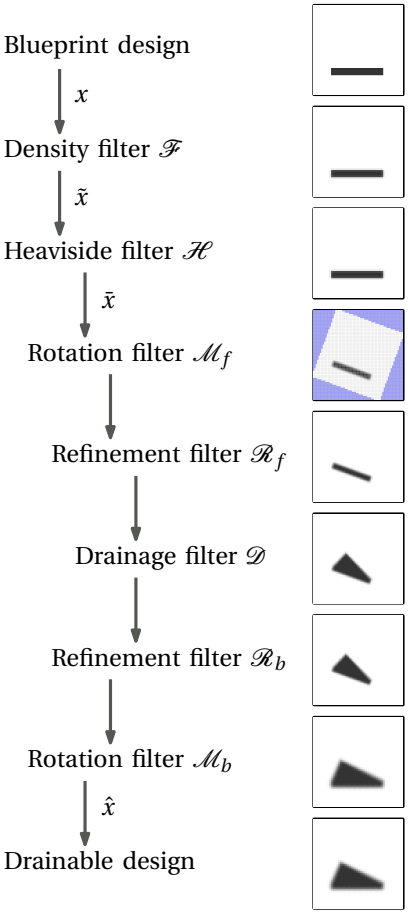


Figure 2.7: Schematic representation of the full drainage filter procedure, where the filter on the left results in the design on the right. Only the drainage filter $\mathscr{D}$ is essential.

As stated in Section 2.2.5, for minimum length scale control the robust approach will be used in the several problems. The simplified robust optimization formulation Lazarov *et al.* [21] is used, which utilizes only the eroded and the dilated design in the optimization problem, and which is given by:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}}: \quad & C(\hat{\mathbf{x}}) = \mathbf{u}_e^T \mathbf{K}_e(\hat{\mathbf{x}}) \mathbf{u}_e \\ \text{subject to}: \quad & \mathbf{K}_e(\hat{\mathbf{x}}) \mathbf{u}_e - \mathbf{p} = \mathbf{0} \\ & \frac{V_d(\hat{\mathbf{x}})}{V^*} - 1 \leq 0 \\ & 0 \leq x_i \leq 1 \quad \text{for} \quad i = 1...N_{el}. \end{aligned} \tag{2.14}$$

Here $\mathbf{K}_e$ and $\mathbf{u}_e$ come from the eroded realization, while V_d comes from the dilated realization. To ensure that the optimized design meets the target volume fraction for the blueprint design V_b^{target}, the volume continuation scheme presented in Wang *et al.* [4] is used. Thus, the allowed volume fraction V^* is updated every 20th iteration as $V^* = \frac{V_d(\hat{\mathbf{x}})}{V_b(\hat{\mathbf{x}})} V_b^{\text{target}}$, where $V_b(\hat{\mathbf{x}})$ refers to the blueprint volume fraction. A β continuation is performed every 15 iterations, initially by increasing the value with 1.0, and after this is higher than 7 the value is multiplied with 1.2.

Classical isoparametric elements are used, with 4-node quadrilateral elements with bilinear shape functions in 2D, and 8-node hexahedral elements with trilinear shape functions in 3D. The modified solid isotropic material with penalization (SIMP) interpolation scheme Sigmund [5] is used for the mapping in each Element i between the drainable density and the Young's modulus, i.e.:

$$E(\hat{x}_i) = E_{\min} + (\hat{x}_i)^p (E_{\max} - E_{\min}). \tag{2.15}$$

In here, $E_{\min}$ and $E_{\max}$ are the lower and upper bounds of the Young's modulus, for which the used values are 10^{-9} and 1.0, respectively. A Poissons ratio of $\nu = 0.3$ is used. The penalization power p is 3.0. Note that the SIMP penalty is useful for eliminating big areas of intermediate densities, but it cannot suppress the intermediate density used for the subelement slope on top of the structure (Figure 2.5).

The 2D problem is solved by the 88 line MATLAB code Andreassen *et al.* [25] augmented with the MMA optimizer of Svanberg [26]. The 3D problem is implemented as an extension to the open-source framework for large-scale topology optimization Aage *et al.* [27] based on PETSc, with the MMA implementation of Aage and Lazarov [28]. Unless mentioned otherwise, all cases have been run for 250 iterations with default optimizer settings. This ensured sufficient convergence of the designs. A full overview of the used parameters is given in Table 2.1.

Table 2.1: Summary of used parameters

Parameter	Value
SIMP power p	3.0
Poissons ratio ν	0.3
$E_{\min}$	10^{-9}
$E_{\max}$	1
P_m	-40
P_n	40
Iterations	250
Rotation filter radius R_m	1.5 l_x
Filter radius R	$3.0l_x$–$4.5l_x$
Threshold η_b	0.5
Offset parameter $\Delta\eta$	0.075
Heaviside β_{final}	12

2.4. NUMERICAL EXAMPLES

To illustrate the performance and characteristics of the drainage filter, the method will be subjected to several tests. Firstly, in Section 2.4.1 the method is tested in 2D to most clearly illustrate its behaviour. Secondly, a simple 3D case is used in Section 2.4.2. Thirdly, the method is tested in 3D in combination with other constraints and methods to address more complex use cases in Section 2.4.3.

2.4.1. 2D CASE

Firstly, the performance is tested in 2D for a cantilever beam problem. The load and boundary conditions can be seen in Figure 2.8. The locations of the loading and clamped area are placed more inwards in the domain, so there the fluid have space on the side to run off, while the structure is not cut off by the domain for any drainage direction. Also sufficient space is provided in the top region of the design domain, which is needed with large run-off angles. The discretization is 150 × 150 elements, the filter radius is 4.5 elements. The robust method was used, in which robust threshold offset $\Delta\eta$ is 0.075, and the Heaviside β is incremented from 1 till 12.

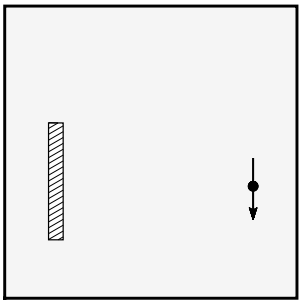


Figure 2.8: Load and boundary conditions for the 2D problem. The load is applied 15% from the right edge, and 40% from the bottom. The fully clamped region is located 15% from the left edge, 40% from the bottom, and has a width of 2% and a height of 40%.

The results of the 2D tests are presented in Figure 2.9. It can be seen that the structure created without the drainage filter contains horizontal areas on the top surface as well as internal voids. This does not meet the criteria of a drainable design. Drainable structures are created by using the drainage filter, as shown in Figure 2.9(b)-2.9(f). Drainability is often obtained by creating a triangular 'roof' on top of the cantilever beam. The fluid can run off on either side of this roof. The filter also bans internal voids from the designs. For a 45° runoff angle, much material is needed to create the roof, at the expense of the lower part of the structure and the compliance performance. Yet, a drainable design is ensured by the strict enforcement of the drainability requirement of the filter. Towards the top of the structure, the elements remain fully solid, which confirms the effectiveness of

the applied smooth min/max operators from Section 2.2.1. For an 11° runoff angle (i.e. $n = 5$), the impact of the drainage requirement is less severe and the structure is allowed to look more similar to the reference design, which is beneficial for the compliance. The smooth slope of the runoff angle has to be represented on a discretized field, which at the surface leads to a gradual increase of the density for every n elements sidewards. In Figure 2.9(d) it can be seen that the filter also works for lower volume fractions, which is usually harder to solve because the structure has to comply with the drainability constraint while there are only limited material resources. Finally, in Figure 2.9(e) and 2.9(f) the structure can be seen for when a different drainage orientation is enforced, more specific a 150° and 270° angle respectively. An 18° runoff angle is used, to make the 'roof' clearly visible, which can now be recognized at the right-bottom and left-side of the structure, respectively. It is visible that the extra filter step does add a blur to the edges of the structure. One could add an extra final Heaviside to mitigate this effect.

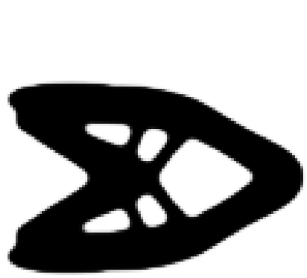
The history of the objective for the experiments in Figure 2.9(a) and 2.9(c), i.e. cases without and with a representative drainability requirement, is plotted in Figure 2.10. The history plot shows that the convergence is mostly smooth, except for some jumps at the continuation of the Heaviside β parameter. The rate of convergence without and with drainage filter is comparable. When using unrealistic high run-off angles (e.g. 45°) a stronger design restriction is imposed, and convergence can be slower than in the reference case.

A final study with the 2D problem concerns the influence of the runoff angle on the compliance. In Figure 2.11 the compliance is plotted for different runoff angles, ranging from 45° ($n = 1$) down to 8° ($n = 7$). As seen before, high runoff angles lead to drastic design changes and therefore significant increases in compliance. In contrast, for runoff angles between 18° and 8°, the compliance only changes 3% at an increase of 8% compared to the reference design. Such angles are more representative for realistic drainability requirements, and only a minor impact on component performance is observed.

2.4.2. 3D CASE

Secondly, the drainage filter is investigated in 3D, also for a cantilever beam problem. The orientation of the load is such that the largest area has to be made drainable, as this poses the biggest challenge for the method. The boundary conditions can be seen in Figure 2.12. The clamped nodes and loaded nodes are 10% from the domain edges in y-direction, to leave space for the fluid to run off within the domain, without any boundary effects. The problem is solved on a mesh of $192 \times 144 \times 96$ elements, for a volume fraction of 10%, and a filter radius of 3 elements, and the drainage orientation is in the z-direction. No robust method was used, and also no Heaviside filter.

Different runoff angles where tested for this problem. The results of the 3D tests can be seen in Figure 2.13. It can be seen in Figure 2.13(a) that if no drainage filter is employed, the design that is created is not drainable. Not all upward facing surface areas have a minimum runoff angle, and there is a pocket in the middle where fluid could reside. By using the drainage filter, the structure becomes drainable, as seen



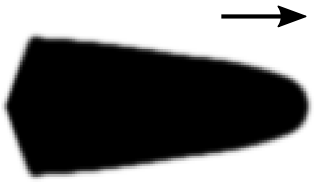
(a) No drainage filter, 20% volume. $C_{ref} := C$.

(b) 45° runoff angle, 20% volume, 300 iterations. $C = 2.43 \times C_{ref}$.

(c) 11° runoff angle, 20% volume. $C = 1.09 \times C_{ref}$.

(d) 11° runoff angle, 15% volume. $C = 1.52 \times C_{ref}$.

(e) 18° runoff angle, 20% volume, 150° orientation. $C = 1.13 \times C_{ref}$.



(f) 18° runoff angle, 20% volume, 270° orientation. $C = 1.92 \times C_{ref}$.

Figure 2.9: Results of 2D cantilever beam problem. The arrow indicates drainage direction other than downwards.

in Figure 2.13(b) and 2.13(c) for runoff angles of 45° and 11° respectively. Material is added on top to create a 'ridge', outer parts of the structure are lowered, and sink holes appear in the middle of the structure, allowing for fluid drainage. For an 11°

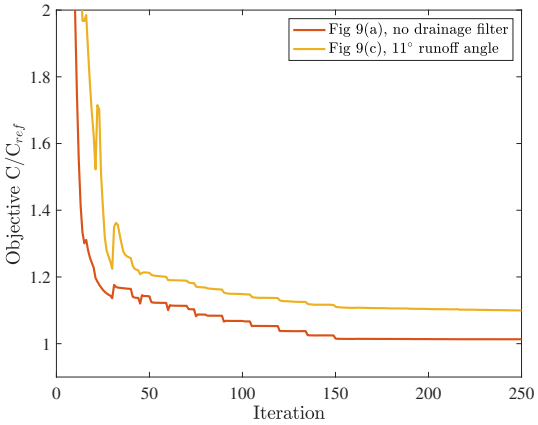


Figure 2.10: Evolution of the objective function for the cases depicted in Figure 2.9(a) and (c).

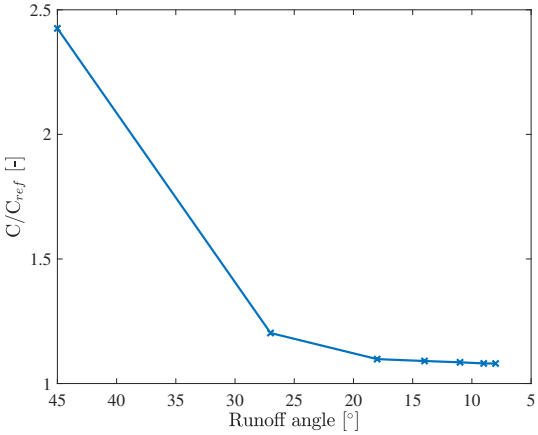


Figure 2.11: The compliance for different runoff angles, for 20% volume, and C_{ref} taken from Figure 2.9(a).

runoff angle, these changes in the design are relatively small, because in this 3D case there are multiple options where the fluid can be send off to. This results in only a small increase in the compliance of the structure, and a much smaller increase than seen for the 2D case.

The smooth isovolume projections in Figure 2.13(e) and 2.13(f) give a representation of the realization of the final design. This shows that no flat areas are present on the structure. The top of the ridge of the structure is the only area with a runoff angle smaller than 10°. The other upward facing surfaces that are part of the ridge are between 10° and 20°. Finally, some mesh dependency can be seen, which is caused by using 5 elements below for the drainage region. This assumes that fluid

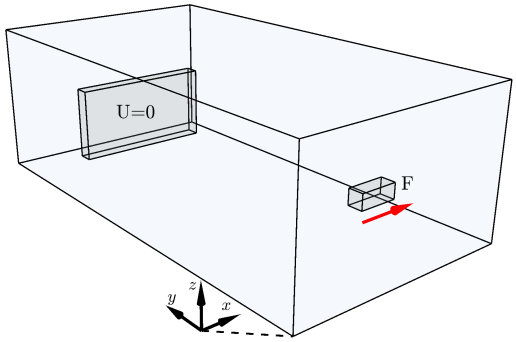


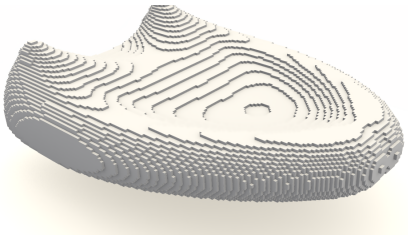
Figure 2.12: Boundary conditions for the 3D cantilever beam problem. The problem has dimensions L_x=1.5, L_y=2, and L_z=1. The load is applied in x-direction at: $x \in$[0.73,0.77], $y \in$[0.19,0.21], and $z \in$[0.39,0.41]. The fully clamped part is located at: $x \in$[0.375,1.125], $y \in$[1,60,1.80], and $z \in$[0.2,0.6].

can only run off in line with the mesh and drainage in the not orthogonal directions has to happen through a combination of these 4 directions. It can be seen that the surfaces that are not in line with the mesh can have a slightly higher runoff angle than the surfaces that are in line with the mesh.

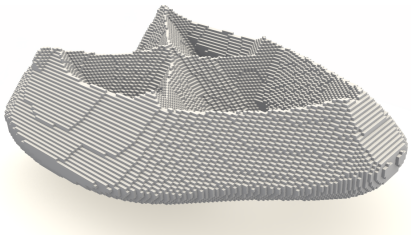
2.4.3. 3D CASES WITH EXTRA CONSTRAINTS

Thirdly, the performance of the drainage filter is tested, while more practical constraints are added to the problem. Case A is a problem where a loading is located more upward in the design domain, which means that not a 'roof' or ridge can just be built on top of the structure. For this, the MBB load case is used, for which the boundary conditions are shown in Figure 2.14. It is important to prevent material from being all in the top row of the domain. This was done by adding 4 rows to the top of the domain, and setting a local volume constraint for this area, taken into account the effect of the density filter radius. The material here is highly penalized with $\frac{V_{\text{top}}}{10^{-6}} - 1 \leq 0$, so the elements here should become void. The problem is solved on a mesh of 192 × 64 × 68 elements, for a volume fraction of 28.2% (which is equal to 30% for the relevant domain volume), and a filter radius of 3 elements. No robust method was used, and also no Heaviside filter.

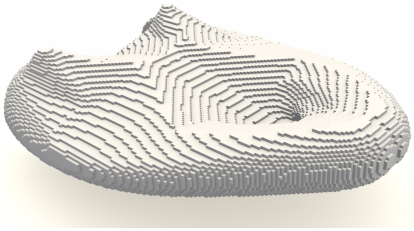
Case B applies a drainage filter in one orientation in combination with a second drainage filter in another orientation. This can ensure the drainability of a component without fixed orientation. The same 3D cantilever beam problem is used, as previously shown in Figure 2.12, again with a mesh of 192 × 144 × 96 elements, for a volume fraction of 15%, and a filter radius of 3 elements. No robust method was used. The drainage direction of the first filter is 10° around the y-axis, the drainage direction of the second filter is 170° around the y-axis. In order to make



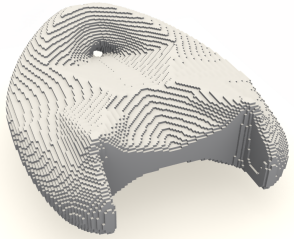
(a) No drainage filter. $C_{ref} := C$.



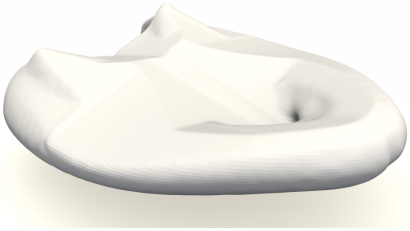
(b) 45° runoff angle. $C = 1.08 \times C_{ref}$.



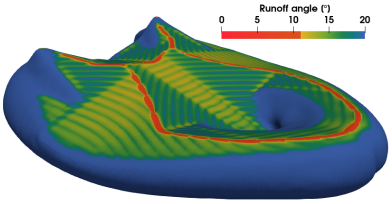
(c) 11° runoff angle. $C = 1.01 \times C_{ref}$.



(d) 11° runoff angle, seen from the back.



(e) 11° runoff angle.



(f) 11° runoff angle, with the actual runoff angle at the surface portrayed.

Figure 2.13: Results of the 3D cantilever beam problem. (a)-(d) is projected with an element 0.5 density threshold value. For (e)-(f) The cell data has been projected to point data, after which an isovolume with threshold 0.5 is created.

the effect more visible, a refinement of 3 is used, which gives a runoff angle of 18°.

Case C applies the drainage filter in combination with an AM filter. The AM filter method described in Langelaar [17] is used. The same mesh as for Case B is used, only the volume fraction is increased to 25% to allow the structure to connect to the baseplate. Again, a refinement of 3 is used.

The results of Case A, the 3D MBB problem, can be seen in Figure 2.15. Without drainage filter, material forms a flat surface at the top of the allowed domain. Applying the drainage filter makes the design drainable, while again obeying the local volume constraint. The top of the structure still looks like a ridge, but this

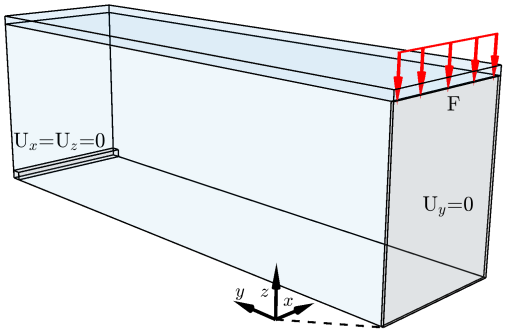


Figure 2.14: Boundary conditions for the 3D MBB problem. The problem has dimensions L_x=1, L_y=3, and L_z=1.0625, where the top 0.0625 is involved in the local volume constraint. There is a symmetry boundary condition at the $x = 0$ plane, with a line load on top. The simply supported part is located at: $x \in [0,1]$, $y \in [2.98,3.0]$, and $z \in [0,0.02]$.

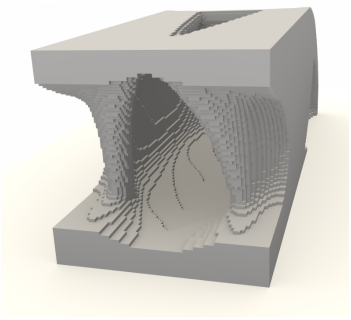
ridge does not enter the constrained area. Also, it is visible that the fluid can run off of the structure on the sides of the domain.

The results of Case B, the 3D beam problem with multiple drainage filters, are presented in Figure 2.16. It can be seen that the 'ridge' is not in the middle anymore but skewed towards one side. Another 'ridge' is created at the bottom of the structure, so the fluid would also run off from this side. Regarding Case C, in Figure 2.17 it can be seen that the structure is both printable and drainable. The challenge for this case is that one filter results in material being added at the bottom, while the other filter results in material being added on top. Yet, both design requirements are fulfilled. This shows that the drainage filter can work well in combination with other complex filters.

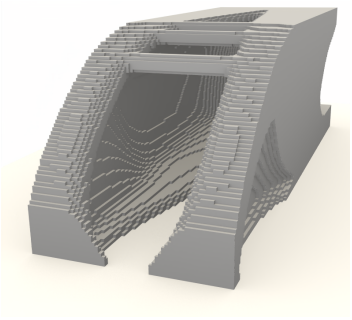
2.5. CONCLUSION

A new filter for density-based topology optimization is proposed that guarantees drainability of a structure through its geometrical design. In a layerwise process, a given blueprint design is transformed into a drainable design, which decreases the need for postprocessing. To the authors' knowledge, it is the first filter to address cleanability, and specifically drainability, of designs generated by topology optimization.

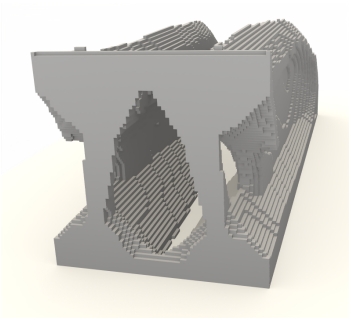
The proposed method has several benefits. Firstly, the method performs well in creating drainable designs. Fluids are ensured to run off the obtained structures, because the drainability definition embedded in the filter eliminates all undrainable regions. The filter is robust, and also works for problems where less material is



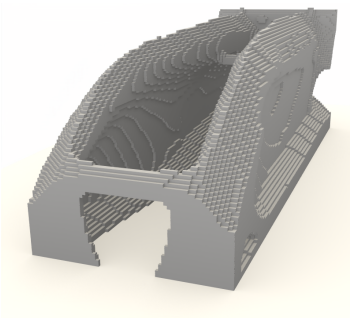
(a) No drainage filter. $C_{ref} := C$.



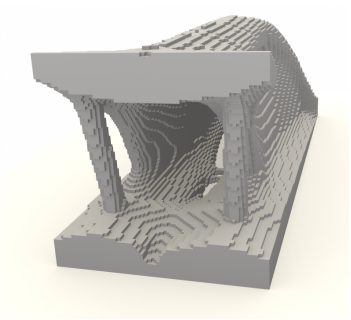
(b) No drainage filter, rear view.



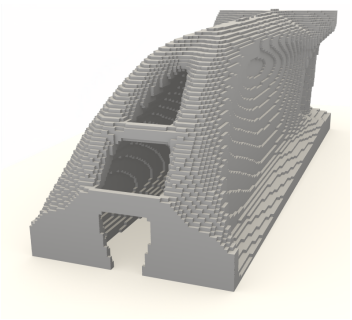
(c) 45° runoff angle. $C = 1.35 \times C_{ref}$.



(d) 45° runoff angle, rear view.



(e) 11° runoff angle. $C = 1.10 \times C_{ref}$.

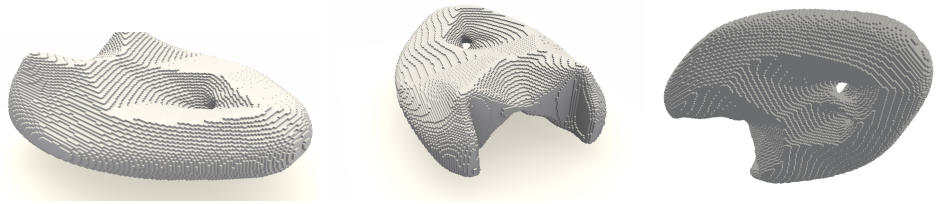


(f) 11° runoff angle, rear view.

Figure 2.15: Results of Case A, the 3D MBB problem, with a local volume constraint at the top of the domain, and with an element 0.5 density threshold projection.

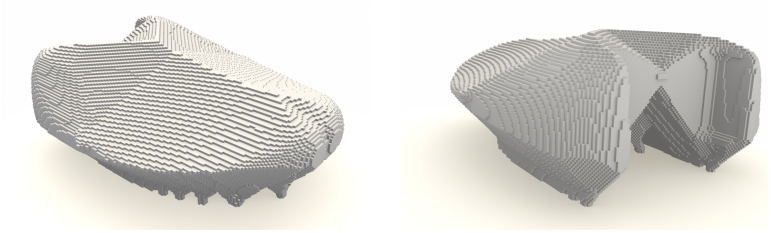
available.

Secondly, the refinement method succeeds in handling shallow runoff angles, which is important for the performance of the components. This can help to create components for which the final geometry is only slightly changed compared



(a) Front view (b) Rear view (c) View from below

Figure 2.16: Results of Case B, the 3D cantilever beam problem, with two drainage filters in different orientations, obtained by 0.5 density threshold projection.



(a) Front view (b) Rear view

Figure 2.17: Results of Case C, the 3D cantilever beam problem, with one drainage filter and one AM filter, obtained by 0.5 density threshold projection.

to the optimized component without drainability considerations. Consequently, the increase in compliance remains minor. Similarly, the rotation filter adds to the applicability of the filter by ensuring that components can be drainable in any chosen orientation relative to the gravity direction.

Thirdly, the proposed layer-by-layer filter is a simple method, which is is easy to implement. The filter itself and the handling of the sensitivities only consist of simple low-cost operations, so that the computational cost of both the filter and its sensitivity analysis is small compared to the finite element analysis. The filter works stand-alone without the need to adjust other parts of the topology optimization code. To facilitate its use and further developments, a 2D implementation is provided with this chapter. Combining it with an overhang filter was successfully demonstrated.

However, the proposed method also has a few limitations. Firstly, the used definition of drainability might be too strict for some applications, as closed internal voids are banned from the filtered design. Secondly, the 2D version of the drainage method often requires the robust formulation, which is needed to prevent small sinkholes. Thirdly, the discrete grid refinement procedure, allows only for some specific runoff angles. Finally, some mesh dependency exists in the 3D implementation where drainage regions of 5 elements are considered. But since the

proposed method builds on a discrete overhang filter, these limitations if necessary can likely be resolved by taking inspiration from a mesh-independent overhang scheme van de Ven *et al.* [e.g. 16].

Drainability is an important aspect of design for cleanability, certainly not the only one. Additional requirements for hygienic design are the topic of future investigations. This is expected to introduce the benefits of topology optimization to new application domains in e.g. biomedical, food processing, and space industries.

3

JETTING

Topology optimization methods are used to design high performance structural components that often have complex geometric layouts. In several industries, components are required to be cleanable, and for this research cleaning by jetting is considered. Thus, being able to ensure jet access on the entire surface of a structure is of interest in topology optimization. In this chapter, a jetting filter is proposed, that turns a blueprint design into a jet accessible design. Two methods are considered to find an access field for each jet. These individual jet access fields are then combined into a total access field, to obtain a cleanable design. Consistent sensitivity analysis is used and the additional computational cost of the jetting filter is modest compared to the finite element analysis. The performance of the two methods is demonstrated with 2D and 3D numerical examples for mechanical and thermal topology optimization problems.

This chapter is based on Giele *et al.* [29].

3.1. INTRODUCTION

Topology optimization (TO) is a computational design method for determining the geometric layout of a part for specific superior characteristics. Since TO does not rely on a specific initial design concept, it can systematically generate innovative geometric layouts, which could have been missed in the traditional design process. With increasing availability of TO techniques, their potential for industrial adoption also increases. In industry, often specific application requirements apply for the designs, for example manufacturability criteria. If manufacturing requirements are not considered during the TO, post-processing of the optimized design might be needed, which may counterweigh the gains achieved in performance by the TO. Consequently, in order to exploit the full potential of TO enabling wider industrial adoption, the relevant specific design requirements should be accounted for in the TO process.

In applications such as food processing, cleanroom equipment space, and medical instrumentation, it is often essential that components are *cleanable*. Cleaning can be performed through scrubbing and polishing, or through jetting a pressurized fluid. In this chapter, the focus is on the latter given its prevalence in the mentioned industries.

In *jetting*, a pressurized cleaning agent such as air, water, or a cleaning liquid, is blasted towards the component, to remove contaminants from its surface. It is important that every surface that can be contaminated is completely cleanable. Assuming cleaning is only ensured through directly blasting the cleaning medium, i.e. ignoring reflected/redirected sprays, all surfaces should be directly accessible/reachable by a jet. In this chapter, we assume the jet source to be a point source that can aim at any direction from a fixed position, although the presented method can also be adapted to jets with specific subsets of directions. The accessibility of a point on the component's surface depends on the overall geometric layout of the component. Thus, during TO it is essential to ensure cleanability by accounting for jet access as a design requirement. Integrating jet accessibility into TO would greatly improve the applicability of TO in numerous industrial applications.

To the best of our knowledge, no TO method considering jet accessibility currently exists. However, related research is available in the literature. Firstly, in the field of cleanability, Li *et al.* [30] presented a waterjet cleaning accessibility analysis. Campana *et al.* [31] quantified a component's cleanability using computational fluid dynamics. However, these studies do not involve TO. In our previous work, Giele *et al.* [8], cleanability is considered for TO, with the goal of ensuring drainability. However this did not involve jet accessibility requirements.

The more general topic of *accessibility* has been more widely studied in the field of TO. Chen *et al.* [32] developed a visibility map for TO and used it for ensuring manufacturability. This method considers for each element a visibility map on a sphere around the object, and compares this map to the visual capacity characteristics of manufacturing processes. This method's integration into TO however leads to a large number of constraints, and only cases where viewing directions align with structured mesh have been presented.

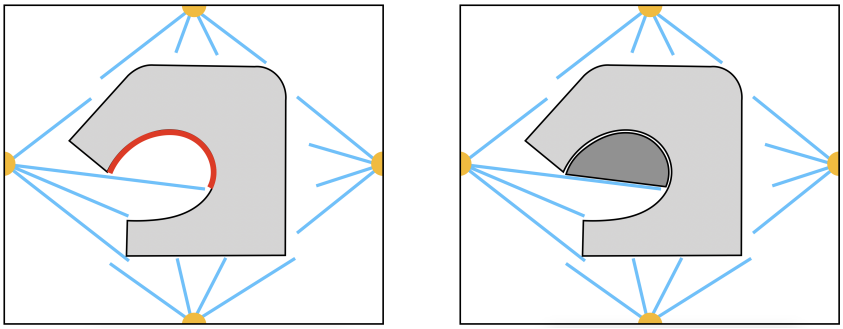
Accessibility is also considered in relation to manufacturing constraints. In milling,

except the case of slot milling, a straight line of access aligned with the tool orientation to the component's surface must exist. All material removed by the tool is then also accessible by a jet following the same trajectory, and thus, a millable part will also be cleanable when the corresponding jet positions and orientations can be chosen. TO filters for multi-axis milling are proposed by e.g. Langelaar [33], Mirzendehdel *et al.* [34] and Høghøj and Träff [35]. However, the tool translates with respect to the workpiece in milling, whereas in jetting, it is common to have jets mounted at a limited number of fixed positions, or a jet operator that can stand at certain positions relative to the part to be cleaned. From these fixed positions, a continuous range of jet orientations can be realized. The component's surface in Figure 3.1a would be entirely accessible with multi-axis milling, while this is not the case with jetting from the four indicated jet positions. Therefore, jet accessibility cannot be represented by existing milling filters and requires a separate consideration.

For casting and molding the component is constrained to be created inside molds, which are later disassembled. So called undercut void regions prevent direct release and require the use of inserts, increasing complexity and cost. Thus, all mould space should be directly accessible in the parting direction. TO filters for casting and molding are for example presented respectively by Gersborg and Andreasen [36] and by Yoon and Ha [37]. However, similar to milling filters, the parting paths for the voids are parallel to each other, while in jetting the access is considered from a point source, and these existing methods therefore can not be applied for jet accessibility purposes.

In this study, a density based topology optimization method ensuring jet access for predefined jet positions is presented. The method is formulated as a filter, which ensures an input blueprint design to become jet-accessible. First, for each jet, an access field is created by the novel filter. We present and compare two methods for this step, one based on trajectory lines and another based on front propagation. Multiple access fields, one from each jet, are then combined into a total access field for all jets. Finally, this total access field is turned into a cleanable design with density field values in the range [0, 1]. As the entire surface area is accessible by at least one jet, the final design will not have internal voids. Therefore, next to the filter, a suggestion is given on how to apply the filter merely as an outer enclosure, in which internal holes are allowed. Two extra steps are introduced in the TO process to improve convergence and stability of the optimization process. The jetting constraint is activated gradually, and numerical stability is promoted by using the unfiltered design in the mechanical/thermal analysis and the filtered design in the volume determination. All steps needed for the proposed jetting filter are differentiable, allowing for gradient-based optimization.

This chapter is organised as follows. In Section 3.2, the method is presented. Section 3.3 gives the numerical examples in 2D for with both ways of formulating the jet access filter, so that their performances can be compared. Next, for the approach with the most potential, 3D numerical examples are presented. The discussion and conclusions are given in Section 3.4 and 3.5, respectively.



(a) Original design (b) Adjusted design

Figure 3.1: Schematic illustration of the jet accessibility concept. Inaccessible surfaces are not allowed, thus inaccessible void area should be turned into solid. The structure is represented with light grey, the jets and jetting fluids are represented by the orange and blue, respectively. In (a) the inaccessible surface is marked in red, in (b) the inaccessible region is turned into solid represented by dark grey.

3.2. METHOD

In this section, the jet access filter is presented. For clarity it is explained in 2D, subsequently its extension to 3D is given. Section 3.2.1 presents the general procedure and the overall structure of the filter. The individual steps of the filter are detailed in the subsequent subsections. Two different methods are presented to obtain a field that determines access for a single jet in Sections 3.2.2 and 3.2.3. Next, in Section 3.2.4 combining the access fields of multiple jets is discussed. In Section 3.2.5, how to obtain an accessible density field as output is shown. Section 3.2.6 focuses on the sensitivity analysis. Finally, in Section 3.2.7 a suggestion is given on how to use the filter to create an optimized jettable enclosure, so that internal holes are allowed to remain in the optimized structure.

3.2.1. GENERAL PROCEDURE

Our aim is to have a filter for density based TO, on a structured grid, that produces jettable designs. We assume the jets have fixed positions and can blast in any direction. A jettable design entails the entire outer surface of the structure to be accessible from at least one jet. *Access* implies all points comprising the outer surface can be reached with a straight line emanating from at least one of the jets, without passing through any part of the structure.

Our jetting filter converts a blueprint design field $\mathbf{x}$ into a jettable design $\check{\mathbf{x}}$. This concept is schematically illustrated in Figure 3.1. Contaminants can adhere at inaccessible parts of the surface, which can be prevented by filling of inaccessible void regions. The filter consists of several consecutive simple steps, where each step is a differentiable operation so that the sensitivities can be calculated easily

with the chain rule. The whole procedure is illustrated in Figure 3.2. Each step of this procedure is an independent operation, so other implementations with similar functionality can replace the ones presented in this chapter.

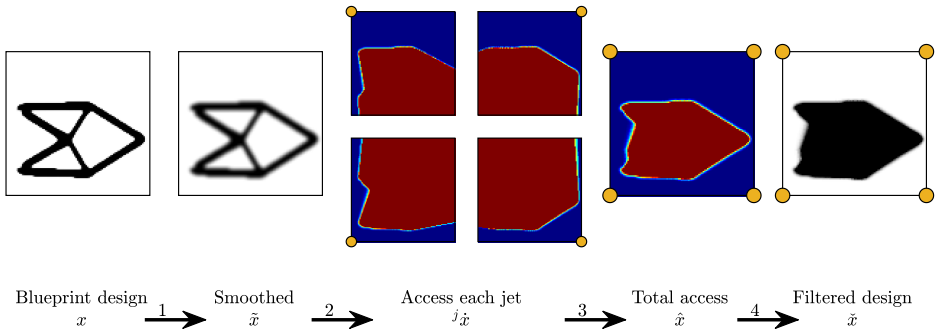


Figure 3.2: Procedure from blueprint design x to cleanable design $\check{x}$. To illustrate the process 4 jets are used, positioned in the outer corners. The steps are indicated by the arrows. The values in the individual and total access fields ${}^{j}\dot{x}$ and $\hat{x}$ are not ensured to be in the range [0, 1] and are therefore represented in another color scheme.

The first step is smoothing of the blueprint design, using the convolution filter $\mathscr{F}$ from Bruns and Tortorelli [23] and Bourdin [22]. This is a standard step in density-based topology optimization to impose a length scale and prevent checkerboarding, and is not further explained. It is defined here as:

$$\tilde{\mathbf{x}} = \mathscr{F}(\mathbf{x}). \tag{3.1}$$

In the second step, an access field ${}^{j}\dot{x}$ is generated for each jet j. A point in the domain is accessible as long as no solid appears in a straight line connecting the point of interest and the jet source. This can be checked by emanating straight lines from the jet source into the domain, and keeping track of the element density values encountered along the way. The principle to convert any design into a valid jettable design, is that in the filtered design, an element further downstream (away from the jet) can not have a density lower than the elements encountered earlier. Two methods are proposed for executing this step, explained in Sections 3.2.2 and 3.2.3. Both methods have advantages and disadvantages in theory and implementation. The results are compared in Section 3.3.

The third step is to combine the access fields of individual jets into a total access field $\hat{x}$. In this step, if a point is accessible by at least one jet, it is considered accessible. Note that the values in the access fields ${}^{j}\dot{x}$ and $\hat{x}$ are not ensured to be in the range [0, 1]. This step is elaborated in Section 3.2.4. The fourth step is to turn the total access field into a density field $\check{x}$. The transformation step for intermediate accessible elements can be done strictly, or more approximately. This is discussed in Section 3.2.5.

3.2.2. JET TRAJECTORY METHOD

The first method to obtain an access field for a single jet is denoted the jet trajectory method. This method is inspired by part of the milling filter presented by Langelaar [33]. The goal is to check the accessibility of all locations in the domain for a jet, by analyzing the smoothed blueprint densities that are encountered on the way from the jet source to the location of interest. In this method, trajectory lines are defined from the source into the domain, along which the total density is calculated between selected points on the line and the jet source. This procedure consists of 3 substeps: i) determine the trajectory lines emanating from a jet and its representation on a structured grid, ii) determine the accessibility along each line, iii) combine the information of multiple lines for the same element.

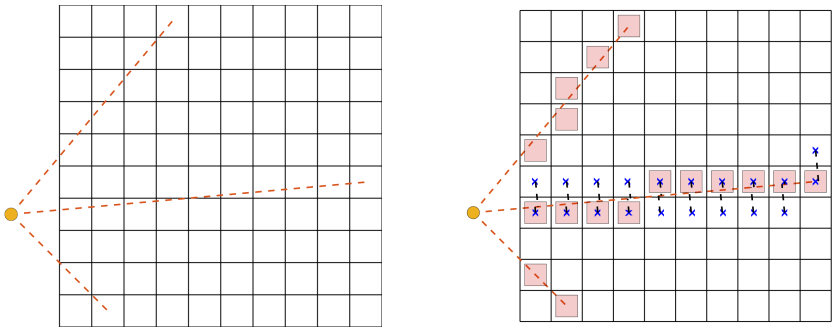
In the first substep, the origin of each line, i.e. the jet location, is connected to the centers of all elements comprising the domain boundaries. Each trajectory line is then represented by a set of elements on the discretized domain that form an approximate representation of the line. For this purpose, the Bresenham line algorithm (Bresenham [38]) is used, for simplicity. For a line l, the starting point position vector $\mathbf{L}_a^l$ and the end point position vector $\mathbf{L}_b^l$ are both assumed to be located in the center of an element for simplicity. The position of $\mathbf{L}_b^l$ with respect to $\mathbf{L}_a^l$ is $\Delta\mathbf{L}^l = \mathbf{L}_b^l - \mathbf{L}_a^l$, which determines the slope of the line. Bresenham defines multiple octants of line slope orientations, and for lines with $|\Delta L_x^l| > |\Delta L_y^l|$ one element is selected in every column, for lines with $|\Delta L_y^l| > |\Delta L_x^l|$ one element is selected in every row. Here, ΔL_x^l and ΔL_y^l are the x and y components of the vector $\Delta\mathbf{L}^l$. This is illustrated in Figure 3.3. Consider the line aiming at the right boundary in Figure 3.3b, where the elements that best describe the line are highlighted by pink. These were selected by choosing one element from every column, of which the element centre is closest to the line. With simple geometric rules, the element center within each column that is closest to a line is determined. The set of elements which are selected for jet j and line l in downstream order, is referred to as ${}^jA^l$, and the total number of elements in the set is ${}^jn^l$.

In the second substep, the accessibility along each line is calculated, computing the cumulative sum of element densities on each trajectory line from start to end. Similar to the multi-axis milling filter by Langelaar [33] the cumulative sum is preferred over a series of smooth maximum operations, as the former adds less nonlinearity for the optimization. This however implies that access field values can exceed 1. This substep is illustrated in Figure 3.4 and can be written as:

$$\begin{aligned} {}^j\ddot{x}_\alpha^l &= {}^j\tilde{x}_\alpha^l, && \text{for } \alpha = {}^jA_1^l, \\ {}^j\ddot{x}_\alpha^l &= {}^j\ddot{x}_\beta^l + {}^j\tilde{x}_\alpha^l, && \text{with } \alpha = {}^jA_i^l \text{ and } \beta = {}^jA_{i-1}^l, \\ & && \text{for } 2 \le i \le {}^jn^l, \end{aligned} \tag{3.2}$$

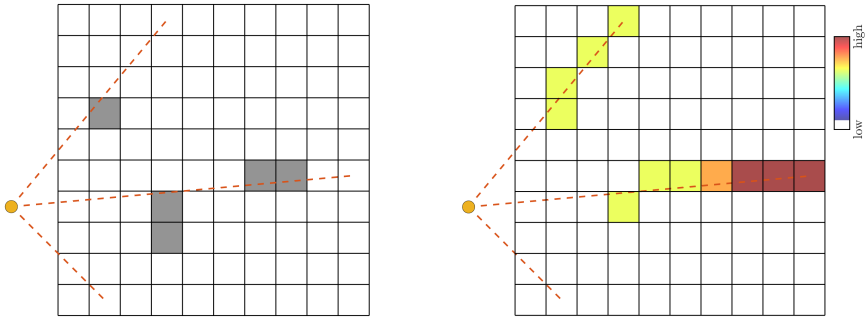
in which $\tilde{x}$ is the input element density which is taken for all elements in ${}^jA^l$ up till element α, and ${}^j\ddot{x}_\alpha^l$ is the element summation value for line l in element α.

In the third substep, multiple lines going through the same element have to be handled consistently. Especially in the vicinity of the jet, many lines go through the



(a) Three trajectory lines (b) Selected elements

Figure 3.3: Trajectory lines are used to check accessibility. In (a) three example lines are shown with each an origin point and an end point. In (b) it is shown that the closest elements along the trajectory line are used to form an approximation of the line on the discretized grid. The pink squares indicate the elements that are selected to describe each line, the blue points around the line aiming at the right boundary show how the distance from an element center to the line can be calculated.



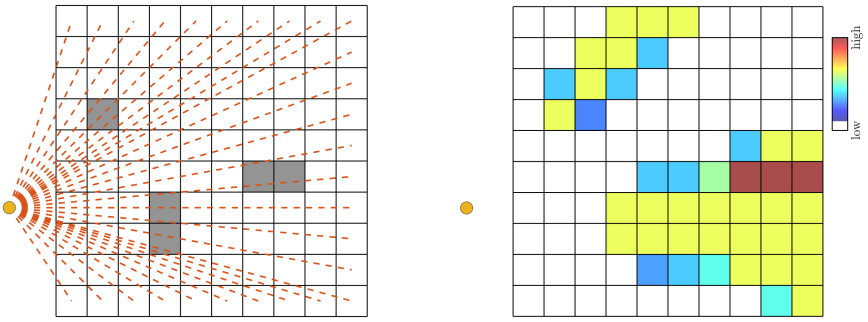
(a) Three trajectory lines, and the input density field (b) The obtained access field

Figure 3.4: The accessibility along three trajectory lines is measured, by cumulative summation of the encountered densities.

same elements. It is necessary to define an unambiguous elemental access value for this situation. In this work, the average is taken, for simplicity and linearity:

$$ {}^{j}\dot{x}_e = \frac{\sum_{l=1}^{\ddot{n}_e} {}^{j}\ddot{x}_e^l}{\ddot{n}_e}, \tag{3.3} $$

where ${}^{j}\ddot{x}_e^l$ is the contribution in element e of line l, and $\ddot{n}_e$ are the total number of line contributions for this element. The result can be seen in Figure 3.5.



(a) Original density field (b) Access field

Figure 3.5: Graphical representation of all trajectory lines covering all elements. The accessibility along all trajectory lines is combined by averaging cumulative densities in elements contained in multiple lines. The average cumulative density values can be much higher than 1, and are therefore represented in a contour plot.

The advantage of the presented jet trajectory approach is its simplicity. The outcome is a directionally summed density field, which is a measure for accessibility, as unobstructed trajectories maintain a zero value. Its extension to 3D requires no adjustments apart from accounting for the third dimension in the trajectory mapping procedure. Also a parallel computation is possible, either by handling parts of the trajectory line by separate processors and later add summation values of upstream parts in the line, or by processing different lines on different processors. However, since the operation is cheap, sequential calculation is not computationally costly.

3.2.3. FRONT PROPAGATION METHOD

The second method to obtain an access field is through front propagation. This method is inspired by the additive manufacturing filter presented in van de Ven *et al.* [16], where anisotropic front propagation is used to distinguish supported and unsupported regions of a part upon addition of layers during additive manufacturing. Here, the goal is to check the accessibility of all locations in the domain, by analyzing the smoothed blueprint densities that are encountered on the way from the jet source to the location of interest. In this method, this is done by propagating a front from the jetting source into the design domain, and calculating the arrival time field. The propagation speed is isotropic but is reduced linearly with the local element density $\tilde{x}$. The time delay compared to an unobstructed reference time field then indicates accessibility. This procedure consists of 3 substeps: i) calculate the reference arrival time field, ii) calculate the density-dependent arrival time field, iii) compute the delay caused by the density field as a measure of accessibility. An example of this approach can be seen in Figure 3.6.

In the first substep, the front propagation arrival time field T_1 is calculated,

which is independent of the density field. This field will serve as a reference, to determine the delay caused by the densities. Front propagation with an isotropic speed function is used, which is given by:

$$|\nabla T_1| F = 1, \tag{3.4}$$

where T_1 is the resulting arrival time function, F is the propagation speed, an initial time $T_1 = 0$ is set at the jet location. In our implementation, the Fast Marching Method is used, presented in Sethian [39].

In the second substep, the front propagation arrival time field T_2 is calculated, which depends on the filtered blueprint densities $\tilde{x}$. The front propagation equation now includes that the front moves slower through solid elements than through void elements:

$$|\nabla T_2| F(\tilde{x}) = 1. \tag{3.5}$$

The slowdown is based on the local density value, and on a lower bound for speed parameter to prevent infinite arrival times. The speed through an element F_e is calculated with:

$$F_e(\tilde{x}_e) = F_{\min} + (1 - \tilde{x}_e)(1 - F_{\min}), \tag{3.6}$$

where $F_{\min} < 1$ is the minimum speed parameter, and $\tilde{x}_e$ is the element density.

In the third substep, the delay field $\tau(\mathbf{x})$ is obtained. This field serves as the measure of accessibility. The delay is simply calculated by subtracting the first arrival time field T_1 from the second arrival time field $T_2(\mathbf{x})$, and is directly used as access field $\dot{x}$:

$$^{j}\dot{x}(\tilde{x}) = \tau(\tilde{x}) = T_2(\tilde{x}) - T_1. \tag{3.7}$$

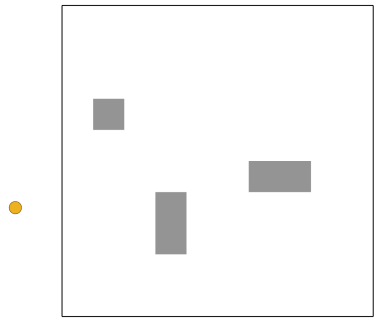
Any point for which the time delay $\dot{x} > 0$ cannot be reached by the jet in a straight line, see Figure 3.6d.

A potential advantage of the presented approach is in the wake that is visible in the delay field. The gradient in the delay field could provide useful information to the optimizer regarding the proximity of an accessible region. The downside of the approach is that the front propagation on a discrete grid shows a discretization effect, which can result in delays also in areas outside the direct wake of the solid area (see also Figure 3.6e). Also, an extra parameter $F_{\min}$ is introduced. The effect of this $F_{\min}$ parameter is visualized in Figure 3.6f, which was created using a smaller $F_{\min}$ than used for the other figures. This can lead to an increase in the delay in and behind solid regions.

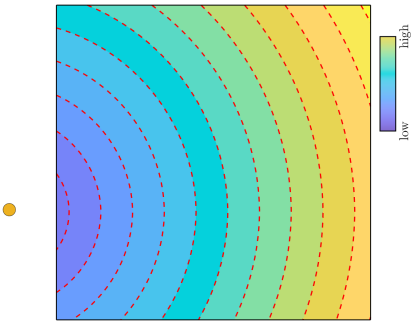
For jet locations outside the domain, one has to use initial arrival times on the boundary surfaces nearest to the jet position, which can be computed simply by Euclidian distance. The computational effort of this method is usually comparable to the jet trajectory method. An extension to 3D is trivial. For a parallel implementation, the reader is referred to e.g. Herrmann [40] or Yang and Stern [41].

3.2.4. COMBINING MULTIPLE JETS

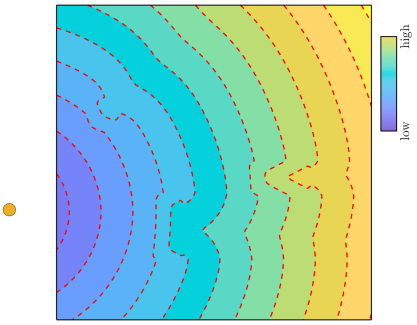
By now the access field $^{j}\dot{\mathbf{x}}$ can be calculated for each jet separately, either as a directional cumulative density field or as a time delay field. Next, these access fields



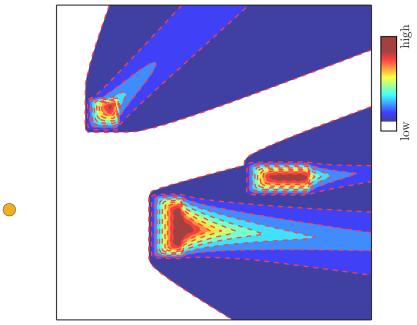
(a) Original density field



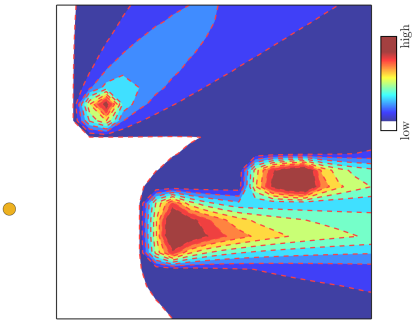
(b) Arrival time field independent of density



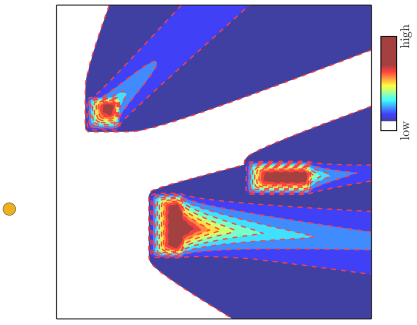
(c) Arrival time field dependent on density



(d) Arrival time delay field



(e) Arrival time delay field, on a 20×20 element mesh



(f) Arrival time delay field, with $F_{\min} = 0.0001$

Figure 3.6: Graphical representation of the front propagation approach. For a clear visualization these figures, except (e), have been obtained on a 100×100 element mesh. Figures (c), (d), and (e), were created with $F_{\min} = 0.1$

for the separate jets have to be combined into one total access field, $\hat{\mathbf{x}}$. Here the lowest access value is relevant for each element, since a single jet access is sufficient for cleanability. This can be obtained in a differentiable manner by applying the

P-norm smooth minimum operator over the access fields of the different jets:

$$\hat{x}_e(x) = \left(\sum_{j=1}^{N} ({}^j\dot{x}_e)^{P_1} \right)^{\frac{1}{P_1}}. \tag{3.8}$$

Here, $P_1 < 0$ is the aggregation parameter, and N is the number of jets. The total access fields for a simple density distribution and two jets can be seen in Figure 3.7.

3.2.5. DESIGN OUTPUT

Finally, the total access field has to be converted to a density field. The access field can have values on the order of the maximum number of elements in all directions. This has to be converted back to the density range of [0, 1]. Several functions can be used for this projection. In this work a P-norm minimum function is used, involving the total access field and the maximum density value of 1:

$$\check{x}_e(x) = \left(1 + \hat{x}_e^{P_2}\right)^{\frac{1}{P_2}}. \tag{3.9}$$

Again, P_2 is the aggregation parameter which is a negative value. The resulting relation between the accessibility values and the density values is shown in Figure 3.8.

3.2.6. SENSITIVITY ANALYSIS

Most of the steps in the filter procedure have straight forward sensitivity operations, such as the density filter in Step 1, and the P-norms in Step 3 and 4. This section will focus on the sensitivities of Step 2 only for the jet trajectory method. For the sensitivity analysis of the front propagation method, the reader is referred to van de Ven *et al.* [16].

For this an elementwise sensitivity ${}^j\dot{s}_e$ for a general function g is assumed, so that ${}^j\dot{s}_e = \frac{\partial g}{\partial {}^j\dot{x}_e}$. For the jet trajectory method, by going back the steps taken in Section 3.2.2 the input sensitivities are first divided by the number of line contributions for $\ddot{n}_e$:

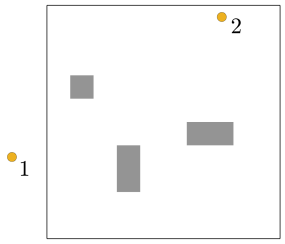
$${}^j\ddot{s}_e = \frac{{}^j\dot{s}_e}{\ddot{n}_e}. \tag{3.10}$$

Next, the cumulative sum of element sensitivities on each trajectory line ${}^j\grave{s}_i^l$ is computed from the end of line to the start of the line:

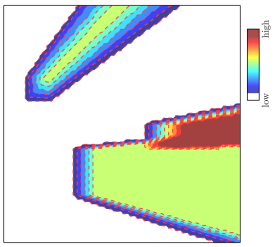
$$\begin{aligned} {}^j\grave{s}_\alpha^l &= \ddot{s}_\alpha^l, && \text{with } \alpha = {}^jA_i^l, \quad \text{for } i = {}^jn^l \\ {}^j\grave{s}_\alpha^l &= {}^j\grave{s}_\beta^l + \ddot{s}_\alpha^l, && \text{with } \alpha = {}^jA_i^l \text{ and } \beta = {}^jA_{i+1}^l, \quad \text{for } 1 \le i \le {}^jn^l - 1. \end{aligned} \tag{3.11}$$

The final output sensitivities $\tilde{\mathbf{s}}$ are obtained by summing up the contributions of all lines and all jets:

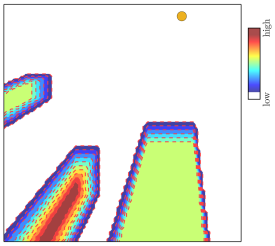
$$\tilde{s}_e = \sum_{j=1}^{N} \sum_{l=1}^{\ddot{n}_e} {}^j\grave{s}_e^l. \tag{3.12}$$



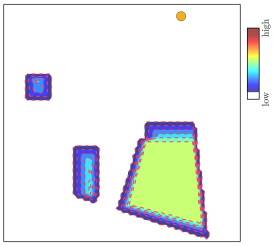
(a) Initial design field, with two jets



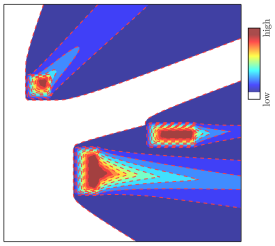
(b) Access field of jet 1



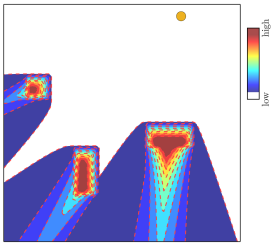
(c) Access field of jet 2



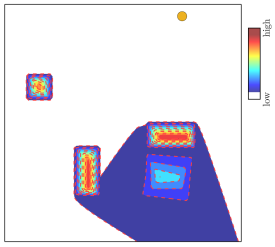
(d) Total access field



(e) Access field of jet 1



(f) Access field of jet 2



(g) Total access field

Figure 3.7: Graphical representation of the combining of separate access fields into a total access field. For (b) and (c) the jet trajectory method is used, for (e) and (f) the front propagation is used. In (d) and (g), the smooth minimum fields are shown.

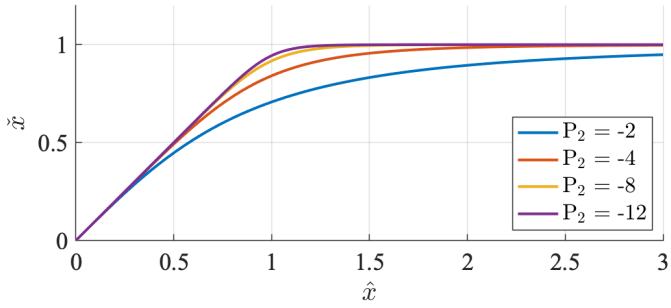


Figure 3.8: The relation between the access field $\hat{x}$ and the jet access density field $\check{x}$, for different negative values of P_2.

3.2.7. JETTABLE ENCLOSURE

The jetting method introduced above does not allow output density fields with internal voids, as these cannot be accessed from outside. This may be overly restrictive for some applications. In this section, a modification is suggested to allow internal voids in a structure with a jettable outer enclosure. This jettable enclosure is part of the TO problem and is load-bearing, and within the enclosure internal voids are allowed. Since this is not the focus of the chapter, only a short description is given here.

To obtain a jettable enclosure, the jetting filter can be combined with existing methods for creating a coating. The full procedure is shown in Figure 3.9. The first steps are the same as in Figure 3.2. Step 1: a smoothed design is created, with a density filter. Step 2: a jettable design with values in the range [0, 1] is obtained. Step 3: the jettable design field is transformed into an enclosure, using the erosion-based interface identification method from Luo *et al.* [42]. That is, after applying a second density filter to create intermediate density boundaries, and applying two Heaviside projections with different threshold values, the eroded design is subtracted from the intermediate design. The thickness of the enclosure can be controlled by the radius of the second smoothing filter. Step 4: the jettable enclosure is added to the smoothed design from Step 1. This can for example be done with a field summation and applying a smooth maximum with a value of 1. Also, it can help to set a void constraint on the boundary of the domain, to prevent the enclosure from adhering to the boundary.

3.3. NUMERICAL EXAMPLES

In this section, the numerical examples are presented and results are shown. Section 3.3.1 presents the optimization formulation, the two extra procedures that improve convergence and stability of optimization, and the used parameters. A 2D mechanical and a 2D thermal optimization problem are described in Section 3.3.2.

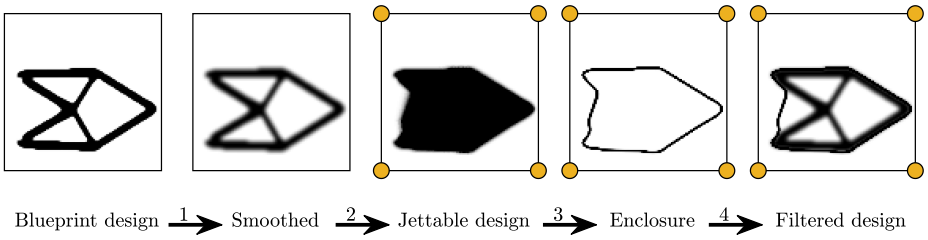


Figure 3.9: Procedure to use the jet access method to only add a jettable enclosure.

In Section 3.3.3, an example is given of the enclosure method from Section 3.2.7. Two 3D mechanical problems are presented in Section 3.3.4.

3.3.1. OPTIMIZATION FORMULATION

For the numerical examples, two optimization problems are considered: a mechanical and a thermal compliance problem, both with a volume constraint. The design variables are the blueprint design field $\mathbf{x}$, and the final filtered design field is denoted by $\tilde{\mathbf{x}}$:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}}: \quad & C(\tilde{\mathbf{x}}) = \mathbf{u}^T \mathbf{K}(\tilde{\mathbf{x}})\mathbf{u} \\ \text{subject to}: \quad & \mathbf{K}(\tilde{\mathbf{x}})\mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \frac{V(\tilde{\mathbf{x}})}{V^*} - 1 \leq 0 \\ & 0 \leq x_e \leq 1 \quad \text{for} \quad e = 1...N_{\text{el}}. \end{aligned} \tag{3.13}$$

In here $\mathbf{K}$, $\mathbf{u}$ and $\mathbf{f}$ denote the finite element system stiffness (/conductivity) matrix, displacement (/temperature) vector and mechanical (/thermal) load vector, for the mechanical and thermal problem respectively. The objective is compliance C, the current design volume is V, the maximum allowed volume is V^*, and the number of elements in the domain is N_{el}.

The proposed filter steps described in Section 3.2 can sometimes suffer from undesirable convergence behaviour, because the sensitivities are most pronounced at the edge of the structure. Due to this, the optimization behaves more like shape optimization rather than topology optimization. If a volume constraint is used, material is removed with high priority, until the volume constraint is satisfied, which increases the chances of ending up in an inferior local optimum. In our approach, the convergence is improved by gradually activating the jetting filter, similar to van de Ven *et al.* [16]. This is done by mixing the density filtered design and the jettable design:

$$\bar{x}_e(x) = (1-\eta)\tilde{x}_e + \eta\check{x}_e, \tag{3.14}$$

where $\eta \in [0,1]$ is the scaling parameter. In this chapter η is continuously increased from 0 to 1 in the first 25 optimization iterations.

To promote numerical stability, designs should not be defined only by a blueprint density field consisting only of an outer contour, as a small change in boundary density value then can result in a large change in the overall design. To prevent boundary-only layouts, we propose to use the jet-filtered design for the volume evaluation, while the smoothed blueprint design is used for the finite element analysis. This encourages the design variables on the inside to increase, as this does hardly influence the volume, but it does increase physical performance. Note that this approach may be less effective for problems where the objective is not monotonically varying with the design variables, but this falls outside the scope of this study.

For the mapping in each element e between the smoothed density and the Young's modulus in $\mathbf{K}$ in Equation 3.13 the modified SIMP interpolation scheme proposed by Sigmund [5] is used, i.e.:

$$E(\tilde{x}_e) = E_{\min} + \tilde{x}_e^p (E_{\max} - E_{\min}), \tag{3.15}$$

with penalization exponent $p = 3.0$, minimum and maximum Young's moduli $E_{\min}$ and $E_{\max}$. For the mechanical problem, $E_{\min} = 10^{-9}$, for the thermal problem $E_{\min} = 10^{-3}$, while for both problems $E_{\max} = 1$. For the finite element analysis, 4-node quadrilateral elements with bilinear shape functions in 2D, and 8-node hexahedral elements with trilinear shape functions in 3D are used.

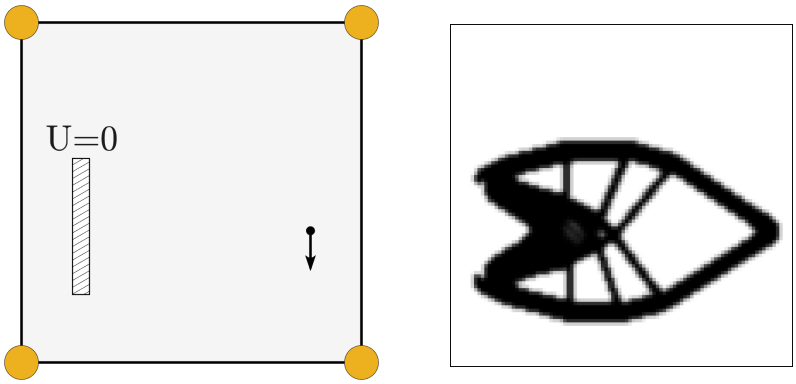
The 2D problem is implemented as an extension to the 88 line MATLAB code by Andreassen *et al.* [25], supplemented with the MMA optimizer of Svanberg [26]. The 3D problem is implemented as an extension to the PETSc code by Aage *et al.* [27]. The optimization is terminated after 250 iterations, by which desired level of convergence was always reached. The density filter radius is 1.5 element length l. The standard MMA values are used. An overview of the used parameters is given in Table 3.1.

Table 3.1: Summary of used parameter values

Parameter	Value
Filter radius R	$1.5l$
SIMP exponent p	3.0
E_{min}	10^{-3} / 10^{-9}
E_{max}	1
Poisson's ratio ν	0.3
P_1	-2
P_2	-4
Number of iterations	250
Front Propagation minimum speed $F_{\min}$	0.1

3.3.2. 2D PROBLEMS

The first numerical example considers a 2D cantilever beam problem. This is a simple problem, with predictable optimization behaviour. The boundary conditions can be seen in Figure 3.10a, as well as the jet positions. A volume constraint of $V^* = 0.2$ is used, and the design domain is discretized by 100×100 elements. For comparison, the result of the standard optimization without the jetting filter is shown in Figure 3.10b. As can be seen, there are many unjettable interior void regions present.



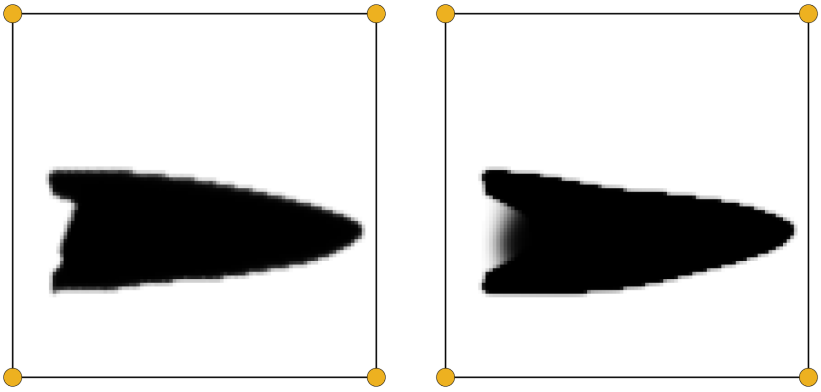
(a) Problem loading. (b) Reference design $C := C_{\text{ref}}$

Figure 3.10: The 2D cantilever beam compliance problem. In (a) the load and boundary conditions are shown with the 4 jet positions in orange. The load is applied 5% from the right edge, and 40% from the bottom. The fully clamped region is located 10% from the left edge, 40% from the bottom, and has a width of 2% and a height of 30%. In (b) the optimization result without jetting filter is shown.

The results for the optimization with the jetting filter are presented in Figure 3.11. As can be seen, no unjettable void regions exist, and the full outer surface area can be reached by at least one of the four jets. However, the compliance is approximately 70 to 80% higher than the reference design. The jet trajectory method and the front propagation method result in similar topologies, however the front propagation shows a more prevalent intermediate density region near the surfaces that are parallel to the jetting direction.

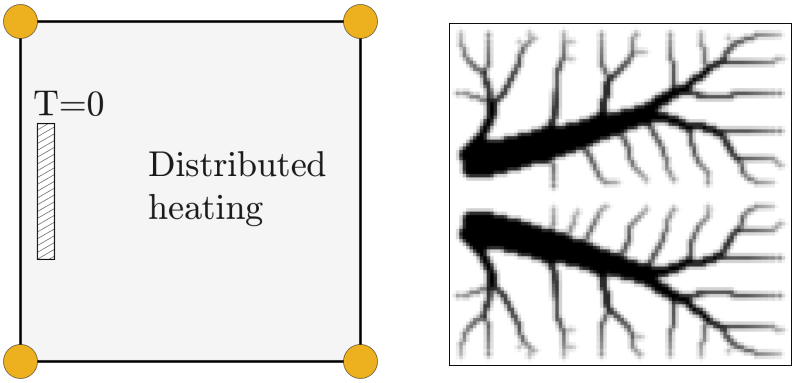
To test the robustness of the method, a heat conduction problem is also studied. The boundary conditions can be seen in Figure 3.12a, as well as the positions of the jets. Again, the maximum allowed volume is $V^* = 0.2$, and a discretization of 100×100 elements. The reference design without jetting filter is shown in Figure 3.12b. As can be seen, the design would be very hard to clean by jets in a 2D setting, with many small branches in an organic layout.

The results for the optimization with the jetting filter are presented in Figure 3.13.



(a) Jet trajectory method. $C = 1.696C_{\text{ref}}$ (b) Front propagation method. $C = 1.817C_{\text{ref}}$

Figure 3.11: The jet-filtered results of the 2D cantilever beam compliance problem, obtained with (a) the jet trajectory method and (b) the front propagation method.

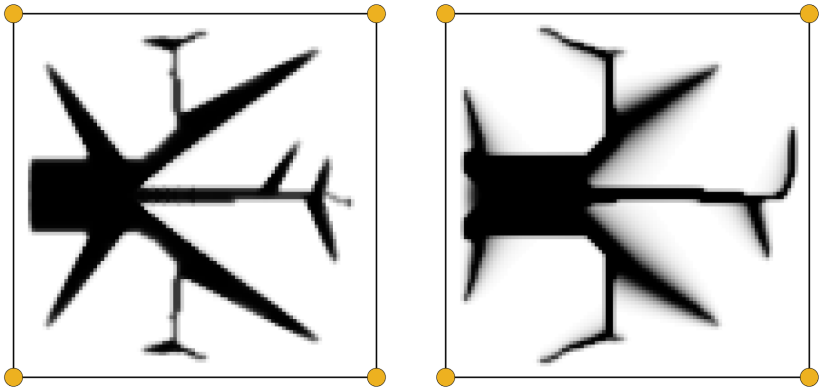


(a) Problem loading. (b) Reference design $C := C_{\text{ref}}$

Figure 3.12: The 2D heat conduction problem. In (a) the load and boundary conditions are shown with the 4 jet positions in orange. The heat sink region is located 2.5% from the left edge, 50% from the bottom, and has a width of 5% and a height of 20%. In (b) the optimization result without jetting filter is shown.

The final designs differ significantly from the reference design, and are indeed jettable. However a price is paid in the objective values, which increase a factor 2.916 respectively 3.985 compared to the reference design. The designs become unsymmetrical during the optimization process for numerical reasons, no symmetry

was enforced on the design. All branches are aimed at one of the jets, so their sides can still be cleaned. Also the difference in performance between the two methods is more clearly visible. Where the jet trajectory method produces clear outlines, the front propagation method has a final design with more prevalent grey regions.



(a) Jet trajectory method. $C = 2.916C_{\text{ref}}$

(b) Front propagation method. $C = 3.985C_{\text{ref}}$

Figure 3.13: The jet-filtered results of the 2D heat conduction problem, obtained with (a) the jet trajectory method and (b) the front propagation method.

Based on the numerical examples above, the performance of the two methods can be evaluated. Firstly, the jet trajectory method is strict in handling encountered solid elements, as the method computes the cumulative sum of element densities. Also the selection of targeted elements is strict, because of the Bresenham line method. Component boundaries which are in line with the jetting direction are clearly defined. On the other hand, the front propagation can be less strict with encountered solid elements, as there is a nonzero minimum speed in solid densities. Also, void regions next to the wake of solids can turn grey, because of discretization effects. These two side effects decrease the strictness, but can help with convergence. For both test problems, better objective values were obtained for jettable designs generated with the trajectory method.

Because a clearly defined boundary is preferred, we will continue this section with the jet trajectory method. For a visual validation of the performance of the method, Figure 3.14 was created to show the accessibility of the boundary for each jet. The void elements are colored based on the jet they are considered reachable by. Extra lines are drawn to indicate that the long branches of the structure are parallel to a jetting direction, so that the whole surface is in fact jet accessible. However it should also be noted that these ‘parallel surfaces’ are on a discretized mesh, which is why some elements on the boundary of the structure seem inaccessible. This is all within a range of one element length, and is caused by the discretization.

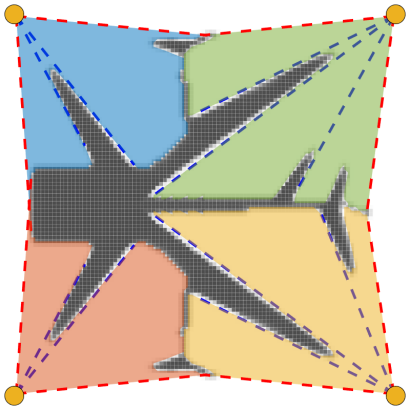


Figure 3.14: Visual representation of what area can be reached by each jet, with the design from Figure 3.13a.

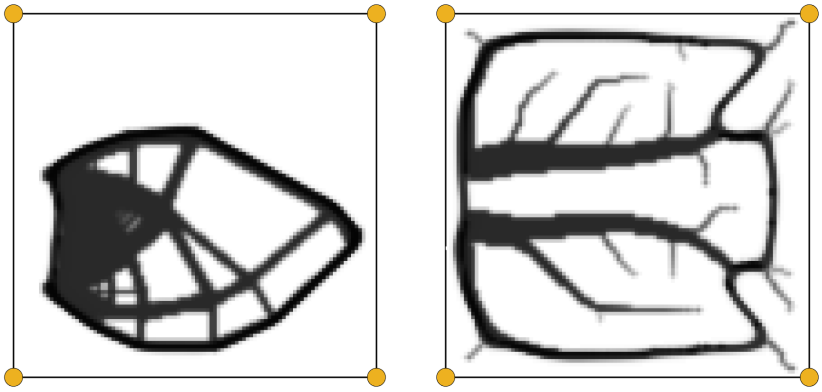
3.3.3. 2D ENCLOSURE PROBLEM

The suggested modification of the presented filter that can create a jettable structure with internal voids, described in Section 3.2.7, was also tested on both 2D problems. The implementation of filter steps to create the density field $\check{\mathbf{x}}$ is performed as described in Section 3.3.1. For the creation of the enclosure field, a density filter with a radius of 8 elements is used, a Heaviside with $\beta = 7$ and $\eta = 0.5$ for the normal field, and a Heaviside with $\beta = 7$ and $\eta = 0.6$ for the eroded field. The final filtered design is obtained by taking an element-wise P-norm smooth minimum with value 1 and the element-wise sum of the smoothed field and the enclosure field, with P-norm parameter $p = -4$. As suggested, a void constraint is used on the boundary of the domain. The results can be seen in Figure 3.15.

As can be seen, the final designs have a jettable enclosure as well as internal voids. Because the designs are less restricted by the filter procedure compared to not allowing internal voids, the impact on the objective value is also smaller, and for these two tests this increased approximately 30% compared to the reference design. For the simple beam problem where the reference design is already nearly jettable, all the extra filter steps in the enclosure procedure make the optimization process unnecessary complex and nonlinear. For the more complex heat problem this solution would be harder to find by an engineer. Note that in both cases the structural/thermal contribution of the enclosure is included in the optimization problem, in contrast to adding an enclosure as a post-processing step.

3.3.4. 3D MECHANICAL PROBLEMS

For the 3D problems, we focus on the jet trajectory method as this was found to perform best in the 2D tests. To investigate its effectiveness in a 3D setting, two tests are performed. The first example considers a simple cantilever beam. The boundary conditions can be seen in Figure 3.16a, as well as the positions of the jets. A volume



(a) Test 1. $C = 1.277 C_{\text{ref}}$, with C_{ref} taken from Figure 3.10b

(b) Test 2. $C = 1.319 C_{\text{ref}}$, with C_{ref} taken from Figure 3.12b

Figure 3.15: The results of the 2D problems, with the jet trajectory method and the enclosure method.

constraint of $V^* = 0.10$ is used, and a discretization of $144 \times 192 \times 96$ elements. For comparison, the result of the optimization without the jetting filter is shown in Figure 3.16b. As can be seen, the reference design without jetting filter creates a flat cantilever beam. Since all four available jets are located at the bottom of the domain, the top of the cantilever beam would not be accessible by any of the jets. In the result with the jetting filter, visible in Figure 3.16c and 3.16d, an additional structure is created on top of the cantilever beam. This ensures that the new top of the structure is now accessible by at least one jet. Even though the added structure contributes marginally to reduce the compliance, this part is still added to comply with the jetting requirements. In spite of the visually large design change, adding the jetting requirement comes at a cost of 38% for the objective.

The accessibility evaluation in Figures 3.16e and 3.16f are created by taking the 0.5 density as a threshold value as solid. This design is subsequently expanded with 1 element in all directions. For all the elements in this shell it is checked if these can be reached in a straight line from any jet. These lines can go through the shell but cannot intersect the original structure. This shell is needed because when the structure is represented in a discretized manner it is possible that even accessible boundary elements are not accessible. A shell with a thickness of one element length shows the accessibility without the discretization effect. It can be seen that the reference design contains a large red surface that is not accessible by jets. The design with jetting filter in 3.16f on the other hand is fully jet accessible.

The second 3D example considers the MBB problem. The boundary conditions can be seen in Figure 3.17a, as well as the positions of the jets. A thin local volume constraint is added to the top of the domain. The jetting filter can add material to the blueprint design up till the boundary of the domain, but for the filter its not problematic if a solid boundary is not accessible. Adding a local volume constraint

gives sensitivity information for the top of the domain. A volume constraint of $V^* = 0.25$ of the relevant domain is used, and a discretization of $64 \times 192 \times 72$ elements. For comparison, the result of the optimization without the jetting filter is shown in Figure 3.17b and 3.17c. Big parts of inside of the structure cannot be reached by any jet.

The resulting structure optimized with the jetting filter is visible in Figures 3.17d-f. This design does comply with the jetting requirements. The two walls in the reference design are combined in one wall in the middle of the domain. In the previous 3D test case a feature was added on top of the structure to make all surfaces accessible, in this test case the main structure is lowered compared to the reference design, so that the jets can still reach the highest surfaces. As a result, the compliance increased by 39%. Again, an accessibility evaluation is visible in Figures 3.17g and 3.17h.

3.4. DISCUSSION

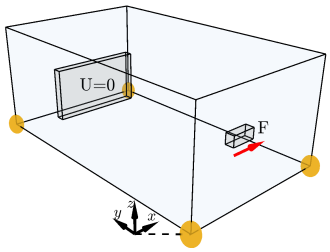
While effective, the proposed method does have several limitations. First, several assumptions are made on the jet, which can differ from specific industrial situations. In the proposed method the jet is stationary. This is a conservative assumption, since there may be cleaning scenario's where jets can be repositioned and moved. In future work we will investigate ways to take this into account. Another assumption on the jet is that it can aim in any direction. It is however quite easy to restrict this, by only consider a subset of the current jet trajectory lines. Secondly, several assumptions are made on the spray of the jet. The distance between the jet and the surface is considered irrelevant. Effectiveness of cleaning may however depend on the relative distance and angle between jet and surface, but this is not considered. Also the deflection of a jet on the surface and any secondary spray is not taken into account. Finally, to make better predictions of the jet access performance of a structure in an industrial setting, one could do a full computational fluid dynamics analysis and perform physical testing.

3.5. CONCLUSION

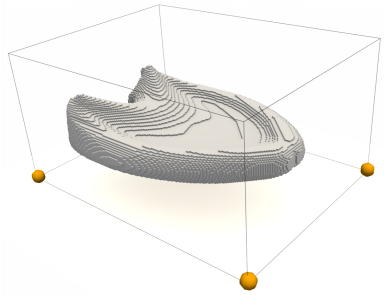
A filter that ensures access for pressure washer jet cleaning has been proposed for density-based topology optimization. After computing an access field of each jet, these are combined into a total access field from which a jettable design field is obtained. To the best of our knowledge this is the first method to include cleanability by jetting in a topology optimization problem, and it was found to be effective in 2D and 3D numerical examples. The proposed method has several advantages. First, it is ensured that always a jettable design is obtained, because the filter turns a blueprint design into a jettable design. Because all steps are differentiable, the optimization process is performed with consistent sensitivities. Next, the method is easy to implement. The filter works as a stand alone procedure and can be used in combination with other design requirements in density based TO. Finally, no significant computational costs are added to the optimization process. This depends

on several factors, such as implementation or the number of jets. For the 3D cases, with 4 jets and a non-optimized implementation, the jetting procedures increased the computation time with about 10%.

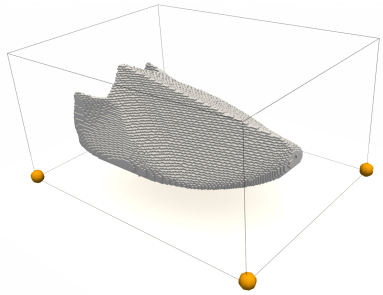
On a final note, the method does have potential to be used in other application settings. Instead of applying the filter for cleanability purposes, there is the possible usage for other applications, such as requirements for visibility and inspection, post-processing of surfaces in 3D printing, or nondestructive testing.



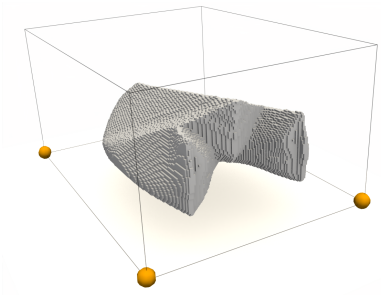
(a) Load and boundary conditions. The problem has dimensions L_x=1.5, L_y=2, and L_z=1. The load is applied in x-direction at: $x \in [0.73, 0.77]$, $y \in [0.19, 0.21]$, and $z \in [0.39, 0.41]$. The fully clamped part is located at: $x \in [0.375, 1.125]$, $y \in [1,80, 1.90]$, and $z \in [0.2, 0.6]$. The four jets are located in all of the lower corners.



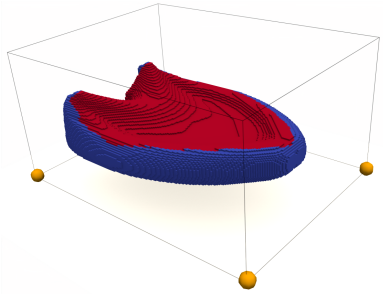
(b) Reference design without jetting filter, frontview. $C := C_{\text{ref}}$



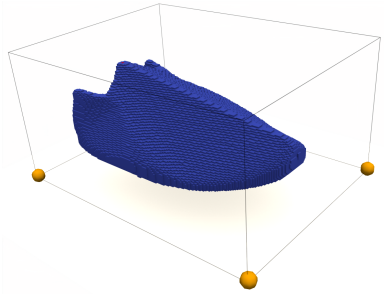
(c) Result with jetting filter, frontview. $C = 1.378 C_{\text{ref}}$



(d) Result with jetting filter, backview.

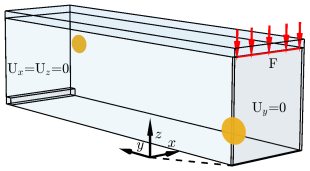


(e) Reference design, unaccessible boundary is red accessible boundary is blue.

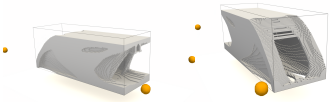


(f) Result with jetting filter, unaccessible boundary is red accessible boundary is blue.

Figure 3.16: The 3D cantilever beam compliance problem optimization results, without and with the jetting filter with the jet trajectory method. Projected with a 0.5 density threshold value.

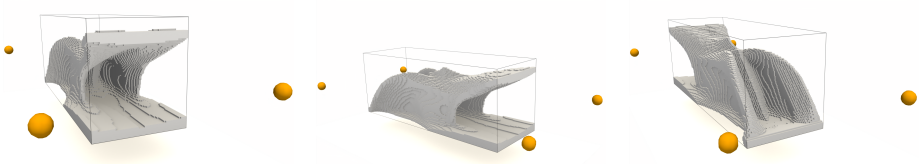


(a) Boundary conditions for the 3D bridge problem. The problem has dimensions L$_x$=1, L$_y$=3, and L$_z$=1.0625, where the top 0.0625 is involved in the local volume constraint. There is a symmetry boundary condition at the $x = 0$ plane, with a line load on top. The simply supported part is located at: $x \in [0,1]$, $y \in [2.98,3.0]$, and $z \in [0,0.02]$. Four jets are used, located at x=-0.5|1.5, y=-0.75|3.75, and z=0.5.



(b) Reference design without jetting filter. $C := C_{\text{ref}}$

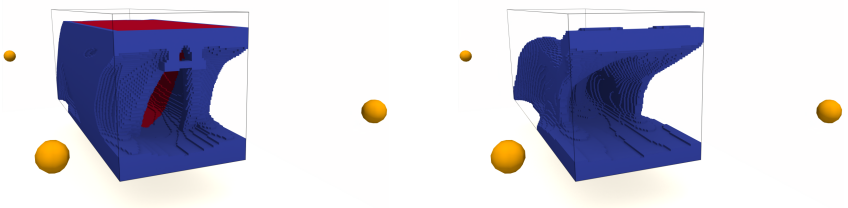
(c) Reference design without jetting filter.



(d) Result with jetting filter.$C = 1.387C_{\text{ref}}$

(e) Result with jetting filter.

(f) Result with jetting filter.



(g) Reference design, unaccessible boundary is red accessible boundary is blue.

(h) Result with jetting filter, unaccessible boundary is red accessible boundary is blue.

Figure 3.17: The 3D bridge compliance problem optimization results, without and with the jetting filter with the jet trajectory method. Projected with a 0.5 density threshold value.

4

GEOMETRIC FILTERS

A framework is proposed for geometric filters in density based topology optimization. Most geometric filters feature density detection in a specified region of interest. In our method this operation is decoupled from the finite element mesh by using interpolated densities. This allows for the filter configuration (e.g. tool orientation or printing direction) to be optimized simultaneously with the geometric layout. The framework is presented in a generic manner, and demonstrated on filters for: milling with simultaneous optimization of milling orientation, jetting with simultaneous optimization of jetting positions, and printing with simultaneous optimization of printing orientation. The performance of the framework is tested with numerical examples for compliance in 2D and 3D on a structured mesh, and in 2D on an unstructured mesh. The framework can extend the design freedom of existing filters, and can serve as a basis for the development of new geometric filters.

This chapter has been submitted for publication in Engineering Optimization, Giele *et al.* [43].

4.1. INTRODUCTION

Topology optimization (TO) is a computational design method that enables superior designs. One key aspect of its success is that only a few inputs are required to generate an optimized geometric layout of a component. In order to make TO more widely applicable for industrially relevant problems, various design requirements such as manufacturability have to be accounted for. Methods have been developed to ensure specific geometric conditions. Several methods ensure a manufacturable design, for example for additive manufacturing [14, 15], milling [33–35], moulding [36, 37], and general manufacturability [32]. Moreover, methods have been developed to ensure a component's surface access during usage, for e.g. cleanability [8, 29].

Geometric constraints are usually imposed through a filter, that converts an input density field into for instance, a manufacturable field. Filtering ensures full compliance with the desired geometric requirement in every design iteration. Note that, the availability of such a strict filter also allows for reformulations using either hard or soft constraints, by quantifying the difference between the input and filtered fields (see for example van de Ven *et al.* [44]). Hence, a filter provides both rigor and versatility, and in this chapter, we therefore focus on various filter implementations within TO. We do so in the context of density-based TO.

Geometric filters often involve settings that are chosen *a priori* by the designer, e.g. the milling tool orientation or printing direction. These predefined settings are referred to as *filter configuration settings*. Conventionally in the majority of filtering schemes, these are defined and remain fixed. However, for design problems that allow some flexibility, fixing the filter configuration limits the design freedom severely and unnecessarily. Allowing for the filter configuration to change as a part of the optimization process is therefore desired.

One example of this idea is to co-optimize the milling direction in a milling process, when only a single direction is allowed. Another example is seen in constraints for additive manufacturing, where the printing orientation could be co-optimized with the design. Several case-specific methods exist for simultaneous optimization of the geometric filter direction/orientation during the topology optimization. For example, for additive manufacturing, the influence of the printing orientation in the context of overhang filters has been addressed by e.g. Langelaar [19]. Simultaneous structural optimization and printing orientation has been addressed e.g. in Olsen and Kim [45], Wang and Qian [46], and Wang [47]. Simultaneous topology and milling orientation optimization for multiaxis machining has been presented by Gasick and Qian [48]. While all these examples have shown degrees of effectiveness, they are specific solutions for one geometric constraint case. A generic approach thus far is missing.

A second shortcoming of many existing geometric filters is that the filter operation is mesh dependent. The gathering of density information for existing filters is often performed with input from a single element or information defined at element centroids on a discretized grid. This can cause mesh dependency, which is for example demonstrated in Delissen *et al.* [49]. Also, it is not always clear how geometric filters defined on a structured grid can be extended

to unstructured grids, widely used in industrial applications. Mesh independent overhang filtering is for example proposed in Gaynor and Guest [14], where an unstructured mesh is mapped on points, and in van de Ven *et al.* [44], where a front propagation method is used to detect unprintable regions. Again, these are case-specific solutions that do not carry over to new use cases in a straightforward manner.

To address these challenges, in this chapter, we present a general framework that allows for co-optimization of the filter configuration for a wide variety of geometric filters and arbitrary meshes. Although different geometric filters serve different purposes, there are several important commonalities. These commonalities allow different geometric filters to be approached similarly with a general framework. Consequently, the proposed methodology in principle can be relevant for all geometric filters used in TO. This eliminates the need for case-specific solutions and facilitates the implementation of new filters.

In the presented framework, the filter configuration settings become additional design variables termed *filter configuration design variables*. This is done by considering neighboring elements during the filtering process. In the filter definition, regions of interest are approximated with input and output points, which are decoupled from the computational domain discretization. Through local filtering operations, element quantities are linked to these decoupled points. As a result, a smooth variation of regions of interest is possible. Consequently, the filter configuration design variables related to the region of interest (e.g. milling tool orientation, jet source location, print orientation) can be changed and optimized, with consistent sensitivities for these new design variables. This allows for simultaneous gradient-based optimization of the design and the filter configuration design variables, resulting in an increase in design freedom.

In Section 4.2, the general framework is explained in detail. Throughout the chapter, the general framework is demonstrated on three example filters: milling, jetting, and printing, for which the specific implementation is given in Section 4.3. Numerical examples are shown in 2D and 3D in Section 4.4. This is followed by a discussion and conclusions in Section 4.5 and 4.6, respectively. An example for how the sensitivities can be calculated is given in Section 4.7.

4.2. FRAMEWORK

Recall that the proposed method is intended for density based TO. The filter turns a input density field $\mathbf{x}$ into a filtered output density field $\hat{\mathbf{x}}$, for which certain geometric requirements are satisfied. For simplicity, we present the framework considering a 2D structured finite element mesh, however extensions to an unstructured grid and to 3D follow naturally. Section 4.2.1 describes the commonalities between seemingly different geometric filters. Next, the general concept and nomenclature are explained in Section 4.2.2, and the general equations from $\mathbf{x}$ to $\hat{\mathbf{x}}$ are given in Section 4.2.3.

4.2.1. COMMONALITIES BETWEEN GEOMETRIC FILTERS

Although different geometric filters serve different purposes, there are several commonalities, which are shown in Figure 4.1.

First, since the topology of the structure is unknown in advance, the desired geometric requirement should be analysed in the entire domain. Therefore, the entire design domain is covered with a finite number of *checkpoints*, where a certain geometric requirement should be met. Whether the geometric requirement is met at a checkpoint is determined through gathering density information within a region of interest, the *detection region*, associated with the checkpoint of interest. This region can be at a single element, a line, or an area connected to the checkpoint. The detection region may also have a specific orientation, size and/or position, here referred to as, the *detection region configuration*, which depends on the filter configuration setting. For example the milling tool orientation for machinability, the jet source location for jettability, or the printing orientation for additive manufacturability.

Secondly, by feeding the gathered density information to a *filter function* at every checkpoint, output densities are calculated. The output density should be such that the geometric requirement is met at the checkpoint. The filter function should also be differentiable to be suitable for gradient based optimization, and for smooth convergence it is desirable to add as little nonlinearity as possible to the optimization problem.

Thirdly, the output is created, by projecting calculated output values at output points onto an output field. Usually, output points are at the checkpoints, but other cases will also be shown, where output points are not coinciding with the checkpoint.

4.2.2. GENERAL CONCEPT AND NOMENCLATURE

The first step is the application of the well-known convolution filter $\mathscr{D}$ with radius R which turns the density design variables $\mathbf{x}$ into $\tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} = \mathscr{D}(\mathbf{x})$ (Bourdin [22] and Bruns and Tortorelli [23]). This step is not explained further, since it is widely used in density-based TO.

Next, in the proposed method the density detection is done with selected *input points* (IPs), which form a discretized representation of the detection region, depicted in Figure 4.2 (left). Next, at each IP, an *input density*, is computed from the $\tilde{\mathbf{x}}$ field. In existing filters, single element density values are often used to represent the density state in the detection region, but to decouple the density detection in the detection region from the finite element mesh, we propose to use interpolated densities at IPs. Subsequently, by performing the filter specific filter function on the input densities, an *output density* is computed. At the *output point* (OP), this output density value is projected onto a field such that a design is created which fulfills the filter requirements (depicted in Figure 4.2, right).

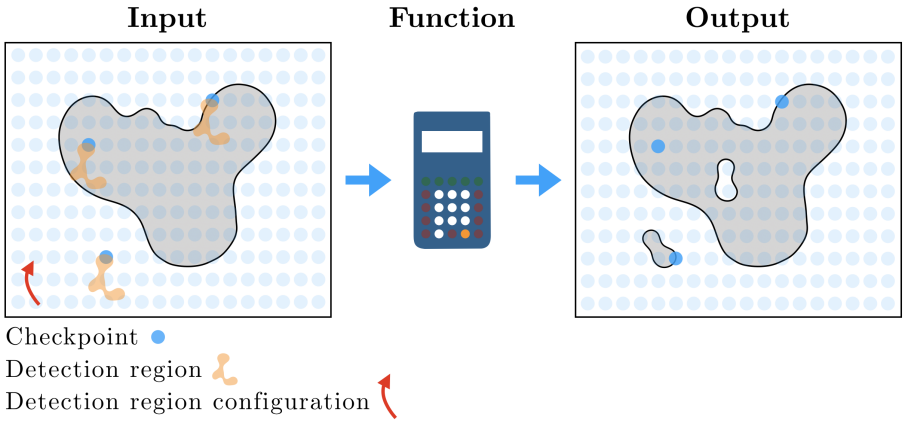


Figure 4.1: Schematic illustration of geometric filter concept, with 3 checkpoints highlighted. Density detection is performed on the input density field, based on which an output field is created. The *detection region configuration* is based on the *filter configuration setting* (shown in red), e.g. the mill orientation, the jet location, or the printing orientation. On the right, a filtered output design is shown.

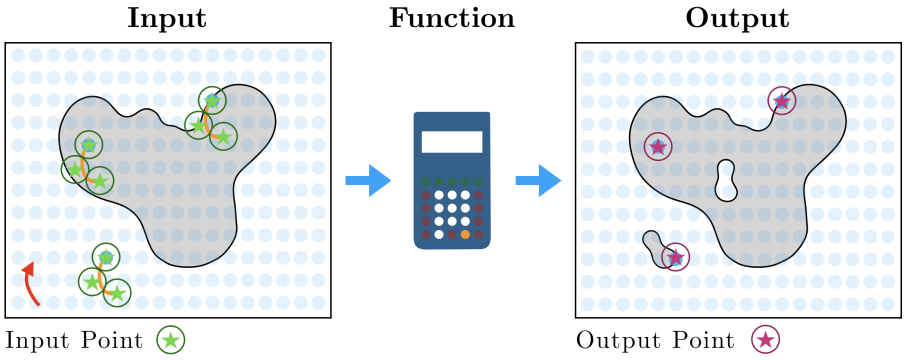


Figure 4.2: Schematic illustration of a generic geometric filter, with 3 checkpoints highlighted. The density detection (left) is performed at input points with (interpolated) input densities. The output field (right) is created at output points with the associated output densities.

An overview of the used steps and symbols is given in Figure 4.3. We refer to the filter configuration design variables as α. For the milling filter the new design variable α_m describes the milling orientation, for the jetting filter the new design variable α_j describes the jet location, and for the printing filter the new design variable α_p describes the printing orientation.

The coordinates of an IP are denoted with $\mathbf{c}^{(\mathrm{IP})}$, where the location is computed

with a filter specific function $\mathbf{c}^{(\mathrm{IP})} = \mathscr{H}(\alpha)$. The input density is denoted with $\chi^{(\mathrm{IP})}$. Subsequently, by performing the filter specific filter function $\mathscr{F}$ on the input densities, an *output density*, $\chi^{(\mathrm{OP})}$, is computed, i.e. $\chi^{(\mathrm{OP})} = \mathscr{F}(\chi^{(\mathrm{IP})})$. At the OP, this output density value is projected onto a field to obtain $\bar{\mathbf{x}}$. Finally, some post filter functions in general form $\hat{\mathbf{x}} = \mathscr{G}(\bar{\mathbf{x}})$ can be applied to obtain the final output field $\hat{\mathbf{x}}$.

Finally, throughout the chapter, $\square^{(\mathrm{IP})}$ and $\square^{(\mathrm{OP})}$ are used as labels for symbols related to the filters, and not as indices. Similarly, the labels $\square^{(\mathrm{m})}$, $\square^{(\mathrm{j})}$, and $\square^{(\mathrm{p})}$, are used to refer to milling, jetting, and printing, respectively. For the α design variable we leave out this label. Indexes are written in italic and subscript, as conventional. The comma in the subscript does not denote differentiation, but is used for multiple indexes.

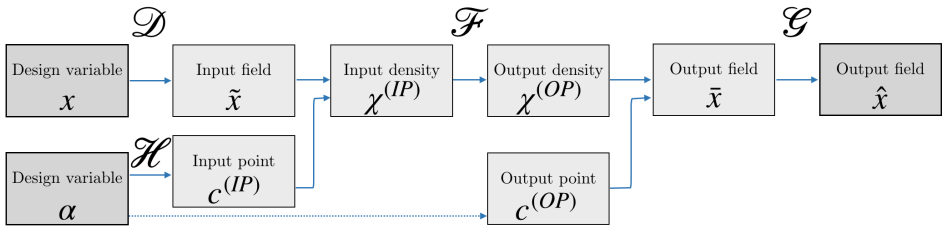


Figure 4.3: Overview of the steps of the geometric filtering framework. The dependency of the output point on design variable α is optional, depending on the specific filter. The operations $\mathscr{F}$, $\mathscr{G}$, and $\mathscr{H}$ are specific for each geometric filter.

4.2.3. GENERAL INPUT-OUTPUT RELATIONS

The IPs are used for the density detection in the detection region. The detection region is defined according to the specific geometric filter requirements. For example for milling, this is the region between the checkpoint and the location where the tool enters the domain. For jetting, this is the region between the checkpoint and the jet source location. For printing, this is the region immediately below the checkpoint according to the printing orientation, considering an overhang angle. The computation for the detection region, thus the IPs, is explained in the filter specific sections.

The input density is the weighted average of the densities within a circular interpolation domain with a radius $R^{(\mathrm{IP})}$. The input density is calculated with a convolution filter centered at the IP using the smoothened density values of the underlying mesh, as visualized in Figure 4.4. The weights of the element densities depend on the distance between the element center and the IP. This allows for smooth changes in the detection region configuration and hence calculation of the sensitivities.

$$w_e^{(\mathrm{IP})}(\mathbf{c}^{(\mathrm{IP})}) = \max(0, R^{(\mathrm{IP})} - \|\mathbf{c}_e - \mathbf{c}^{(\mathrm{IP})}\|), \tag{4.1}$$

where $w_e^{(\mathrm{IP})}$ is the linear weight function between the IP and element e. $\mathbf{c}^{(\mathrm{IP})}$ and $\mathbf{c}_e$ are the coordinates of the IP and the centroid of element e, respectively. Next, the weights $w_e^{(\mathrm{IP})}$ are normalized such that their values range between 0 and 1, and all weights related to e sum to 1:

$$\breve{w}_e^{(\mathrm{IP})}(\mathbf{c}^{(\mathrm{IP})}) = \frac{w_e^{(\mathrm{IP})}}{\sum_{j=1}^{N_{\mathrm{el}}} w_j^{(\mathrm{IP})}}, \tag{4.2}$$

where N_{el} is the total number of elements in the domain. It remains to calculate $\chi^{(\mathrm{IP})}$ from the smoothed density field $\tilde{\mathbf{x}}$ using the normalized weights $\breve{w}_e^{(\mathrm{IP})}$:

$$\chi^{(\mathrm{IP})}\left(\tilde{\mathbf{x}}, \mathbf{c}^{(\mathrm{IP})}\right) = \sum_{e=1}^{N_{\mathrm{el}}} \tilde{x}_e \breve{w}_e^{(\mathrm{IP})}. \tag{4.3}$$

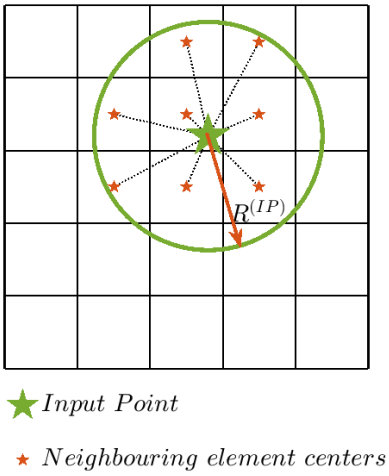


Figure 4.4: An input point (IP) is the location where the input density is calculated. It is computed as the weighted average of smoothed densities $\tilde{x}_e$ from the elements with centers in the interpolation domain.

Finally, it is important to emphasize that $\chi^{(\mathrm{IP})}$ not only depends on the density design variables, but also to the IP location $\mathbf{c}^{(\mathrm{IP})}$, and thus on the new filter configuration design variable α, such that the sensitivities with respect to the detection region configuration can be calculated. Figure 4.5 illustrates how the sensitivities of the input density with respect to $\tilde{\mathbf{x}}$ and α can be calculated. The steps from left to right correspond Eqs. (4.1-4.3). Following the steps in reverse order, taking the derivatives and using the chain rule leads to the sensitivities for the $\tilde{x}$ and α design variables.

Once the input densities are calculated it remains to perform the filter function, based on the specific filter characteristics, yielding the output value $\chi^{(\mathrm{OP})}$. The

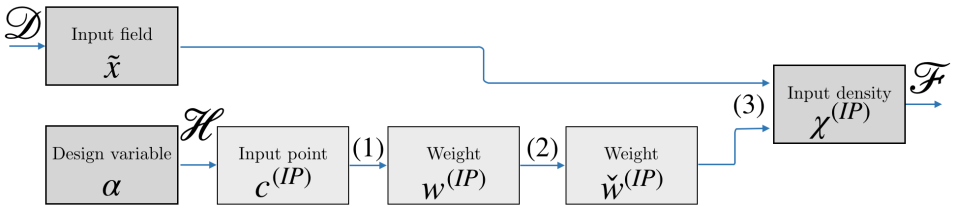


Figure 4.5: The flow diagram illustrating calculation of input density $\chi^{(\mathrm{IP})}$ from a density field $\tilde{\mathbf{x}}$ and the filter configuration design variable α. The number represents the equation used for each step. By following the steps in reverse order, calculating derivatives, and through the chain rule, the sensitivities are calculated.

4

operation should ensure that the filter requirements are met on the checkpoint. This very much depends on the specific filter and is therefore further discussed in the following filter specific sections (Section 4.3.1-4.3.3). The filter operation should also be smooth so that the sensitivities can be calculated consistently.

Lastly, the output field $\bar{\mathbf{x}}$ is created, which is often a density field. First, at each OP an output value $\chi^{(\mathrm{OP})}$ is projected onto one or more elements in its close proximity onto the field $\check{\mathbf{x}}$. Often, the OP coincides with the checkpoint, however examples with different locations will be shown in this chapter. The projection is also carried out with a convolution filter, for which first the weights on elements e are calculated:

$$w_e^{(\mathrm{OP})}(\mathbf{c}^{(\mathrm{OP})}) = \max\left(0, R^{(\mathrm{OP})} - \|\mathbf{c}^{(\mathrm{OP})} - \mathbf{c}_e\|\right), \tag{4.4}$$

where $R^{(\mathrm{OP})}$ is the output convolution filter radius, and $\mathbf{c}^{(\mathrm{OP})}$ and $\mathbf{c}_e$ are the coordinates of OP and the centroid of element e, respectively. Again the weights are normalized:

$$\check{w}_e^{(\mathrm{OP})}(\mathbf{c}^{(\mathrm{OP})}) = \frac{w_e^{(\mathrm{OP})}}{\sum_{j=1}^{N_{\mathrm{el}}} w_j^{(\mathrm{OP})}}. \tag{4.5}$$

The output density value is projected onto field $\check{\mathbf{x}}$ as follows:

$$\check{x}_e = \chi^{(\mathrm{OP})} \check{w}_e^{(\mathrm{OP})}. \tag{4.6}$$

To account for elements getting multiple contributions from multiple OPs, the final output value for element e is obtained by dividing by the sum of the contributions:

$$\bar{x}_e(\tilde{\mathbf{x}}) = \frac{\sum_{i=1}^{n_e} \check{x}_{e,i}}{\sum_{i=1}^{n_e} \check{w}_{e,i}^{(\mathrm{OP})}}, \tag{4.7}$$

where n_e is the number of contributions on element e, and $\check{x}_{e,i}^{(\mathrm{OP})}$ and $\check{w}_{e,i}^{(\mathrm{OP})}$ are the value and weight of the individual contributions of different input points i. Finally, it should be mentioned that in many cases, $R^{(\mathrm{OP})}$ is small so that the output value is projected to a single element only. The dependency of the output field $\bar{\mathbf{x}}$ on the output density (and potentially α) is illustrated in Figure 4.6.

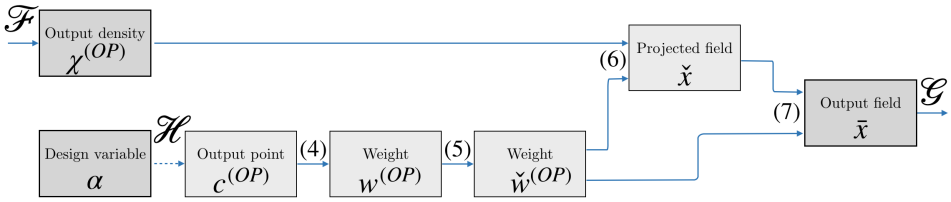


Figure 4.6: The flow from the output density (and potentially the filter configuration design variable α), to the output field. The number represents the equation used for each step. By following the steps in reverse order, and through the chain rule, the sensitivities follow naturally.

4.3. EXAMPLES OF GEOMETRIC FILTERS

In this section, the framework is applied to milling, jetting, and printing filters, respectively. The basic concepts of these filters and the ways filter configuration design variables can enhance design freedom are illustrated in Figure 4.7. In milling, the component's surface should be reachable by the milling tool, with orientations that can be optimized as well. For jetting, the component's surface should be reachable by the jetted fluid, originating from jet source locations that are to be optimized. Thirdly, for printing, material can only be built upon previously deposited material (or the baseplate) considering a critical overhang angle and according to a printing orientation, which can also be optimized simultaneously. The three filters and their implementation in the proposed framework are introduced below.

4.3.1. MILLING WITH TOOL ORIENTATION OPTIMIZATION

In this section, the first example is given of how the general framework can be applied to a milling filter with an adjustable tool orientation, inspired by Langelaar [33], but now with simultaneous optimization of milling tool orientation. The milling filter ensures that a component can be manufactured with 2.5D milling, such that tools should have access to the entire component surface. Thus undercuts or internal holes are not admissible. Consequently, tool path access is considered while neglecting the tool size for simplicity. In density based TO this means that regions further downstream the tool path should not have a lower density than regions encountered upstream. The tool access orientations are traditionally defined *a priori* and remain constant throughout the optimization. However, in our new filtering framework, the milling orientation is the filter configuration design variable designated as α_m. In the explanation in the first part of this section, only one milling orientation is considered.

According to the general framework the checkpoints should be located at all element centers within the domain. The detection region for each checkpoint becomes the region around a straight line upstream in the milling orientation. This is shown in Figure 4.8.

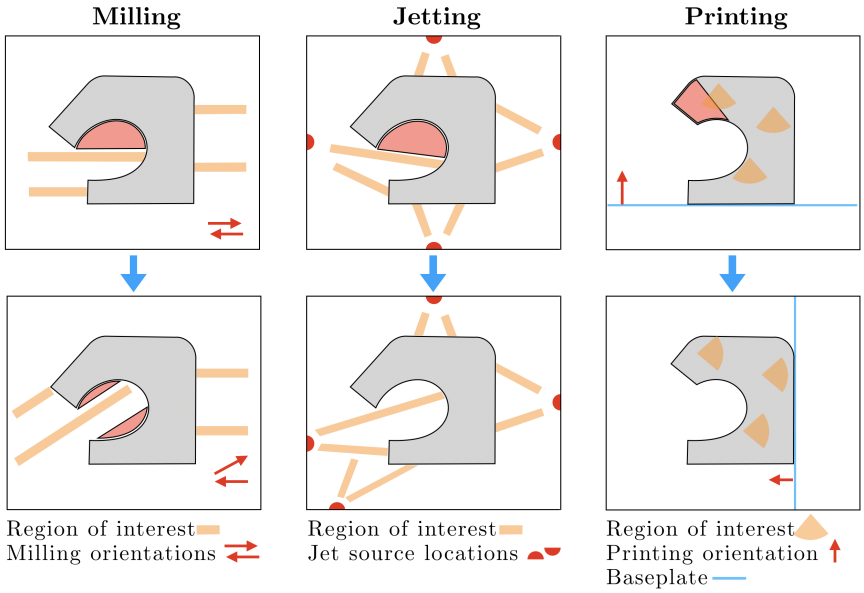


Figure 4.7: The influence of the *filter configuration design variables* on the design freedom for three example filters. The original design is light grey, forbidden regions marked in red. The three filters are discussed in Section 4.3. Top: designs with random filter configuration settings. Bottom: design with more favorable filter configuration settings.

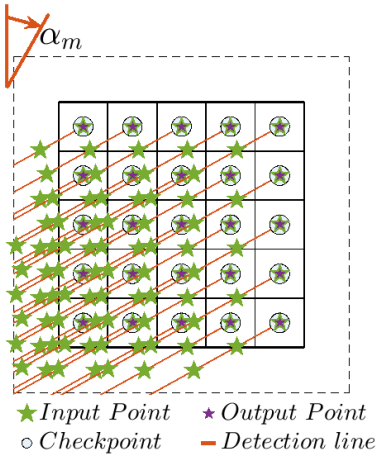


Figure 4.8: Schematic illustration of milling access filter. The checkpoints are located in all elements.

However, for computational gains, it is possible to reduce the number of checkpoints,

since many detection regions overlap. More specifically, the checkpoints furthest downstream can share density information with checkpoints in their detection region. Therefore, in our implementation, checkpoints are only located at the element centers of the downstream boundaries. This ensures that the entire domain is covered, although other choices for the checkpoint locations are possible. As illustrated in Figure 4.9, detection region lines are drawn towards these checkpoints, representing the detection region. Along each line, IPs are separated with length $l^{(\mathrm{IP;m})}$, which is chosen as the element size in x-direction $l^{(\mathrm{x})}$. The coordinates of the IPs can be calculated as follows:

$$\begin{aligned} \mathbf{c}_{k,i}^{(\mathrm{IP})} &= \mathscr{H}^{(\mathrm{m})}(\alpha) = \mathbf{c}_k - i\mathbf{r} \\ \text{with} \quad \mathbf{r} &= l^{(\mathrm{IP;m})} \begin{pmatrix} \cos(\alpha_m) \\ \sin(\alpha_m) \end{pmatrix} \\ \text{for} \quad i &= n_1 \dots n_2, \in \mathbb{Z}, \end{aligned} \tag{4.8}$$

where, $\mathbf{c}_k$ are the center coordinates of the boundary element k. n_1 and n_2 are respectively the lowest and highest integer values of i resulting in an IP within the domain plus some spacing to account for $R^{(\mathrm{IP})}$, such that $\mathbf{c}_{k,i}^{(\mathrm{IP})} \in \Omega^{(\mathrm{IP})}$.

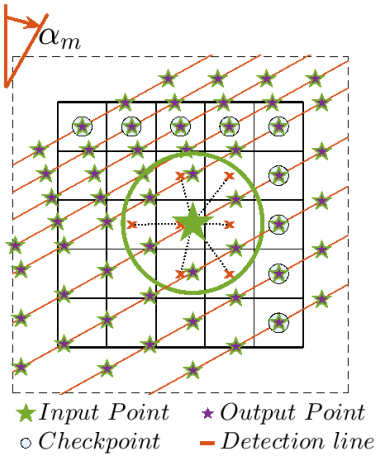


Figure 4.9: Schematic illustration of milling access filter. For clarity, only one IP is highlighted. The checkpoints are at the downstream boundary elements. Milling orientation α_m determines the detection line orientation, and $\Omega^{(\mathrm{IP})}$ is marked by the dashed square.

A linear operation that ensures elements further downstream do not have a lower density than elements upstream, is the cumulative sum:

$$\begin{aligned} \chi_{k,i}^{(\mathrm{OP})} &= \mathscr{F}^{(\mathrm{m})}\left(\chi^{(\mathrm{IP})}\right) = \\ &\begin{cases} \chi_{k,i}^{(\mathrm{IP})} & \text{for} \quad i = n_1, \\ \chi_{k,i-1}^{(\mathrm{OP})} + \chi_{k,i}^{(\mathrm{IP})} & \text{for} \quad n_1 < i \leq n_2, \end{cases} \end{aligned} \tag{4.9}$$

in which i is an integer index that runs along the IPs of a detection region line, and $\chi_i^{(\mathrm{IP})}$ and $\chi_i^{(\mathrm{OP})}$ are the input density and output density respectively.

The OPs for the milling filter concide with IPs, and the values obtained from Eq. (4.9) are projected from the OPs to the element centroids within $R^{(\mathrm{OP})}$.
The full milling filter procedure is summarized by the flowchart presented in Figure 4.10, visualizing how the output field $\bar{\mathbf{x}}$ is related to the input field $\tilde{\mathbf{x}}$ and the filter configuration design variable α_m. Observe that in Figure 4.10 a change in α smoothly influences the IP location. The IP location in turn influences $\chi^{(\mathrm{IP})}$, which influences the summation value $\chi^{(\mathrm{OP})}$ and ultimately the output field. Thus sensitivities of the output field with respect to α can be calculated.

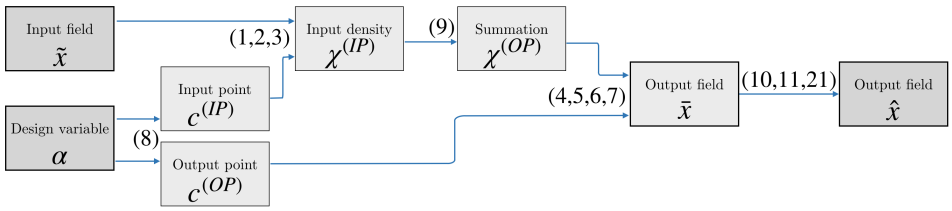


Figure 4.10: Schematic representation of how the output field $\hat{\mathbf{x}}$ is calculated, for the milling filter for one milling orientation. The numbers represent the equations used for each step. By following the steps in reverse order, and through the chain rule, the sensitivities follow naturally.

When multiple milling orientations are used in the manufacturing of the design, milling access from any considered milling orientation is sufficient for manufacturability. The combination of all milling access fields, for each orientation denoted here with ${}^m\bar{\mathbf{x}}$, yields $\bar{\mathbf{x}}^{(\min)}$. This can be achieved in a differentiable manner by applying the P-norm smooth minimum operator over the access fields of the different milling orientations:

$$\bar{x}_e^{(\min)} = \mathscr{G}_1^{(\mathrm{m})}(\bar{x}) = \left(\sum_{m=1}^{N^{(\mathrm{m})}} ({}^m\bar{x}_e)^{P_1} \right)^{\frac{1}{P_1}}. \tag{4.10}$$

Here, $P_1 < 0$ is the aggregation parameter, and $N^{(\mathrm{m})}$ is the number of milling orientations considered.

Note that the access field can have values higher than 1, in fact in the order of the maximum number of elements in all directions. This has to be converted back to a density field with values ranging between 0 and 1. A P-norm smooth minimum function is used, involving the total access field and the maximum density value of 1 as:

$$\bar{x}_e^{(\mathrm{out})} = \mathscr{G}_2^{(\mathrm{m})}\left(\bar{x}^{(\min)}\right) = \left(1 + \left(\bar{x}_e^{(\min)}\right)^{P_2}\right)^{\frac{1}{P_2}}. \tag{4.11}$$

Again, $P_2 < 0$ is the aggregation parameter.

4.3.2. JETTING WITH JET POSITION OPTIMIZATION

This section presents a second example of application of the proposed framework, for the formulation of a jetting filter. A jetting filter ensures that the entire component's surface can be cleaned with fluid jets, as considered in Giele *et al.* [29], but now with simultaneous optimization of the jet source locations. From the jet source locations the fluid is sprayed following straight lines in any direction, and should have access to the entire component's surface. In this chapter, for simplicity it is assumed that the jet source is a point, and a direct access from the source to the surface irrespective of the incoming jet angle will suffice (i.e. no secondary spray is considered). Note that, this is equivalent to a visibility filter (e.g. Chen *et al.* [32]). In density based TO, this implies that along each jetting line, regions further downstream are not permitted to have a lower density than those encountered upstream. The jet source locations were traditionally defined *a priori* (Giele *et al.* [29]). In this work, filter configuration design variables $\alpha_j^{(\mathrm{x})}$ and $\alpha_j^{(\mathrm{y})}$ are introduced which represent the x- and y-coordinates of the jet source positions. Since the jetting filter has various similarities to the milling filter, for most of the filtering steps the reader is referred to Section 4.3.1, and only differences are described here. Similar to the milling section, in this section only one jet source is considered.

In jetting, the detection regions are between the jet source location and each checkpoint, as visualized in Figure 4.11. Just like in milling, for computational reasons, only jet lines towards the (downstream) boundary elements centers are used. The coordinates of each IP can be calculated as follows:

$$\begin{aligned} \mathbf{c}_{k,i}^{(\mathrm{IP})} = \mathscr{H}^{(\mathrm{j})}(\alpha) &= \boldsymbol{\alpha}_j + i\mathbf{r} \\ \text{with} \quad \mathbf{r} &= l^{(\mathrm{IP;j})} \begin{pmatrix} \cos(\beta_k) \\ \sin(\beta_k) \end{pmatrix} \\ \text{for} \quad i &= n_1 \dots n_2, \in \mathbb{Z}. \end{aligned} \tag{4.12}$$

Again, $l^{(\mathrm{IP;j})}$ is set to be the element size in x-direction. β is the angle between the jet source location and the boundary element center it is aiming at:

$$\beta_k = \cos^{-1}\left(\frac{c_k^{(\mathrm{x})} - \alpha_j^{(\mathrm{x})}}{c_k^{(\mathrm{y})} - \alpha_j^{(\mathrm{y})}} \right). \tag{4.13}$$

Again, $\mathbf{c}_k$ are the center coordinates of the boundary element k. n_1 and n_2 are respectively the lowest and highest non-negative integer values of i resulting in an IP within the domain plus some spacing to account for $R^{(\mathrm{IP})}$, such that $\mathbf{c}_{k,i}^{(\mathrm{IP})} \in \Omega^{(\mathrm{IP})}$. The rest of the jetting filter steps are identical to those of the milling filter.

4.3.3. PRINTING WITH PRINT DIRECTION OPTIMIZATION

In this section, a third example is given, to illustrate how the proposed method can be applied as a printing filter, inspired by Langelaar [15], but now with simultaneous optimization of the printing direction. A printing filter ensures that

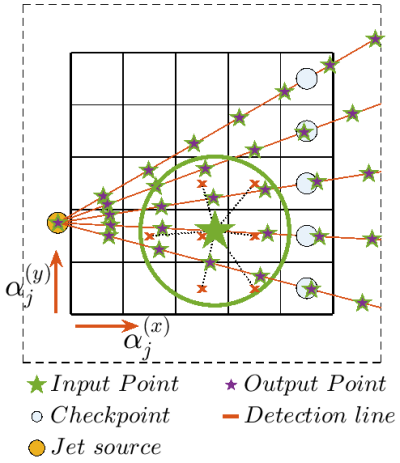


Figure 4.11: Jetting filter checkpoints and detection regions. For clarity, lines are only drawn on one domain boundary, and only one IP is highlighted. The checkpoints are at the downstream boundary elements. Jetting location α_j determines the detection line orientation, and $\Omega^{(\mathrm{IP})}$ is marked by the dashed square.

a component can be manufactured with additive manufacturing. This requires the design to be self-supporting or in other words, each new layer of material deposited is supported by previously deposited material, within the material and process dependent permissible overhang angles. In density based TO this can be ensured by a rule that a region further in the printing direction cannot have a density higher than that below. The 3D printing orientation and baseplate configuration in TO are traditionally defined *a priori*. In the first part of this section we will relax the fixed printing orientation assumption. An additional relaxation of the baseplate configuration (orientation and height) is only considered in the second part of this section. The new filter configuration design variable is angle α_p which represents the angle between the printing orientation and the y-axis. For clarity and simplicity, we focus on the 2D setting, where a single orientation variable suffices.

The checkpoints are located at the centers of all elements in the design domain. The detection region for each checkpoint k with coordinates $\mathbf{c}_k$ is the region opposite of the print orientation forming an arc of twice the overhang angle θ with a radius $l^{(\mathrm{IP;p})}$. In this region, three IPs are located (nine in a 3D implementation), as shown in Figure 4.12, with a distance of $l^{(\mathrm{IP;p})} = 1.5l^{(\mathrm{x})}$ from the checkpoint. The IPs are

defined as:

$$\begin{aligned} \mathbf{c}_{k,i}^{(\mathrm{IP})} &= \mathscr{H}^{(\mathrm{p})}(\alpha) = \mathbf{c}_k + \mathbf{r}_i \\ \text{with} \quad \mathbf{r}_i &= l^{(\mathrm{IP;p})} \begin{pmatrix} \cos(\alpha_p + \gamma_i) \\ \sin(\alpha_p + \gamma_i) \end{pmatrix} \\ \text{for} \quad \boldsymbol{\gamma} &= [-\theta, 0, \theta]. \end{aligned} \tag{4.14}$$

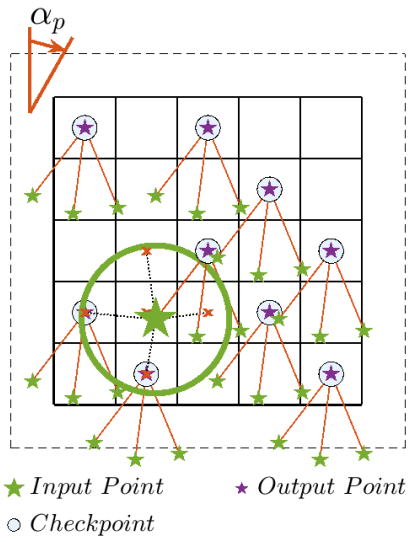


Figure 4.12: Schematic illustration of the printing filter, checkpoints and detection regions. For clarity, only several checkpoints are drawn, and only one IP is highlighted. The distance between the checkpoint and IP is $1.5l^{(\mathrm{x})}$ and is marked by the orange lines.

For clarity the filter operation is written in two directly related parts, $\mathscr{F}_1^{(\mathrm{p})}$ and $\mathscr{F}_2^{(\mathrm{p})}$. The first operation function that ensures printability is a smooth maximum operation between the three input densities:

$$\chi_k^{(\max)} = \mathscr{F}_1^{(\mathrm{p})}\left(\chi^{(\mathrm{IP})}\right) = \left(\sum_{i=1}^{N^{(\mathrm{p})}} (\chi_{k,i}^{(\mathrm{IP})})^{P_3}\right)^{\frac{1}{Q}}, \tag{4.15}$$

where following Langelaar [15], $P_3 > 0$ is the aggregation parameter, and $Q = P_3 + \frac{\log(N^{(\mathrm{p})})}{\log(0.5)}$, where $N^{(\mathrm{p})}$ is the number of IPs used. The second operation is a smooth minimum between the $\chi_k^{(\max)}$ value and the density value of the checkpoint element. This value, $\chi^{(\mathrm{k})}$, can also be interpreted as another input density with small radius, such that $\chi^{(\mathrm{k})} = \tilde{x}_e$ where element e is the element at checkpoint k, such that

$\mathbf{c}^{(\mathrm{IP})} = \mathbf{c}_k$. The minimum value is calculated as follows:

$$\chi_k^{(\mathrm{min})} = \mathscr{F}_2^{(\mathrm{p})}\left(\chi^{(\mathrm{k})}, \chi^{(\mathrm{max})}\right) = \frac{1}{2}\cdot\left(\left(\chi^{(\mathrm{k})} + \chi_k^{(\mathrm{max})}\right) - \sqrt{\left(\chi^{(\mathrm{k})} - \chi_k^{(\mathrm{max})}\right)^2 + \epsilon} + \sqrt{\epsilon}\right), \tag{4.16}$$

where ϵ is a parameter that controls the smoothness of the minimum function, and where for $\epsilon \to 0$ an exact non-smooth minimum operator is obtained. When no baseplate is considered, the output density value is the minimum value, such that $\chi_k^{(\mathrm{OP})} = \chi_k^{(\mathrm{min})}$. The OP coincides with the checkpoint, and the output field is $\bar{\mathbf{x}}^{(\mathrm{out})}$ obtained with a small $R^{(\mathrm{OP})} < l^{(\mathrm{x})}$ such that $\chi^{(\mathrm{OP})}$ is projected to only one element.

4

In contrast to the previous filter examples, for the printing filter the operation function should be executed in a certain order. In previously proposed mesh-bound printing filters (e.g. Langelaar [15]), this results in a layer by layer operation. In the present filter with variable orientation however, the order starts from the lowest unprocessed checkpoint in the printing orientation. The input density value $\chi^{(\mathrm{IP})}$ (from Eq. 4.3) is thus calculated using the output values $\bar{\mathbf{x}}^{(\mathrm{out})}$. The authors note that these input values are always previously calculated and in a lower layer. Effectively this sequential compound operation results in a filter function $\mathscr{F}$ that could complexly be written with only $\tilde{x}$ as input. The full procedure is summarized in Figure 4.13. Note in Figure 4.13 how a change in α, smoothly influences the IP locations. This influences $\chi^{(\mathrm{IP})}$, which influences the maximum, $\chi^{(\mathrm{max})}$, and minimum, $\chi^{(\mathrm{min})}$, values. Ultimately the output field is influenced. Thus sensitivities of the output field with respect to α can be calculated.

Also, note that, for this example filter, for simplicity the effect of changes in processing order due to changes in filter orientation variable is not accounted for in the sensitivity analysis. Neglecting this would probably not have a significant influence.

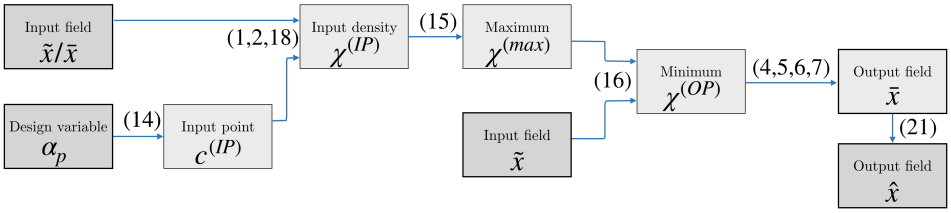


Figure 4.13: Schematic representation of the flow from the design variables to the output field, for the printing filter procedure without the baseplate consideration. The numbers represent the equations used for each step. By following the steps backwards, and through the chain rule, the sensitivities follow naturally.

The second part of this section focuses on optimizing the baseplate height. Printing

begins on the baseplate, meaning that only regions above the baseplate can be printed, and whereas regions in the design domain below the baseplate cannot be printed. While this can be achieved in various ways, for the purpose of illustrating the versatility of our framework, we will use a similar approach here with output points. This differs from the previous examples in the sense that the density input gathering step is not required, and the framework is used to project a solid region into the domain. The baseplate height is defined by filter configuration design variable α_b. Recall that the orientation of the baseplate is linked to the print orientation α_p.

First, a field is created with the same discritization as the density field, $\mathbf{x}^{(\mathrm{BP})}$, with values of 0 above the baseplate and values 1 below the baseplate, including a smooth transition region around the baseplate with radius $R^{(\mathrm{OP;BP})}$. The transition region is required for gradient based TO. In line with the framework, this field is created with OPs, whose location is based on α_p and α_b. One line of OPs, $R^{(\mathrm{OP;BP})}/2$ above the baseplate, project a value of $\chi^{(\mathrm{OP})} = 0$, while another line of OPs, $R^{(\mathrm{OP;BP})}/2$ below the baseplate, project a value of $\chi^{(\mathrm{OP})} = 1$, as shown in Figure 4.14. In this chapter a radius $R^{(\mathrm{OP;BP})} = 3l^{(\mathrm{x})}$ is used, and the distance between neighbouring OPs on each line is $0.5l^{(\mathrm{x})}$.

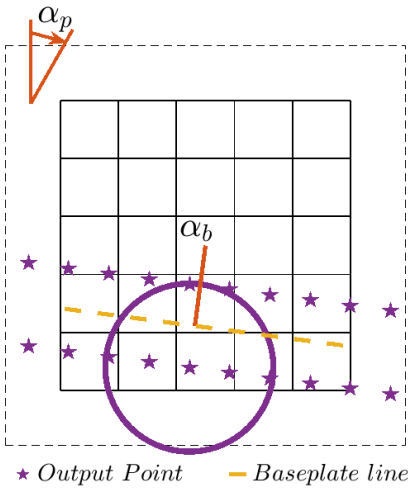


Figure 4.14: Printing baseplate field creation with OPs. For clarity of the figure, only one projection domain and $R^{(\mathrm{OP;BP})}$ is highlighted. The top row of OPs of projects a value of $\chi^{(\mathrm{OP})} = 0$, the bottom row of OPs projects a value of $\chi^{(\mathrm{OP})} = 1$.

Elements in the baseplate field below the baseplate without OP contributions are given a value of $x_e^{(\mathrm{BP})} = 1$, which still leads to consistent sensitivities since only the

transition region has an influence on the sensitivities of α_b and α_p.

The baseplate field $\mathbf{x}^{(\mathrm{BP})}$ can be used to create a printable output field, only above the baseplate, such that for checkpoint k the output density is calculated:

$$\chi_k^{(\mathrm{OP})} = \mathscr{F}_3^{(\mathrm{p})}\left(\chi_k^{(\min)}, x_k^{(\mathrm{BP})}\right) = \chi_k^{(\min)}(1 - x_k^{(\mathrm{BP})}). \tag{4.17}$$

Also, it should be simulated that it is possible to print on the baseplate. This is done by adding $\tilde{x}_e$ and $x_e^{(\mathrm{BP})}$ in the detection region in Eq. (4.3), such that this becomes:

$$\chi^{(\mathrm{IP})}\left(\tilde{\mathbf{x}}, \mathbf{c}^{(\mathrm{IP})}, \mathbf{x}^{(\mathrm{BP})}\right) = \sum(\tilde{x}_e + x_e^{(\mathrm{BP})})\,\check{w}_e^{(\mathrm{IP})}. \tag{4.18}$$

The procedure with baseplate consideration is shown in Figure 4.15.

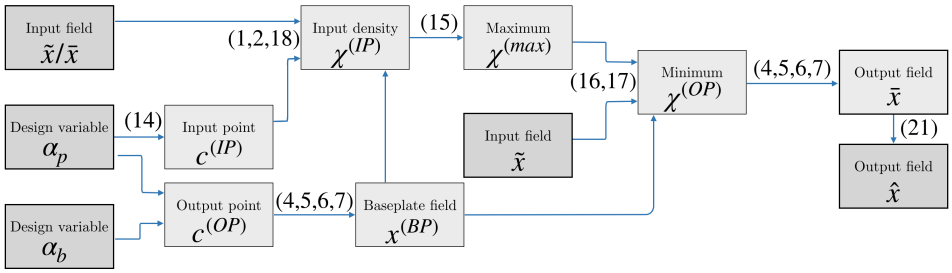


Figure 4.15: Schematic representation of the flow from the design variables to the output field, for the printing filter procedure including the baseplate consideration. The numbers represent the equations used for each step. By following the steps backwards, and through the chain rule, the sensitivities follow naturally.

4.4. NUMERICAL EXAMPLES

In this section, the numerical examples are presented and results are shown. Section 4.4.1 presents the problem formulation, including two extra steps to improve convergence and stability of the optimization, and the parameters used. 2D mechanical optimization problem on structured and unstructured meshes are presented in Section 4.4.2 and 4.4.3, respectively. Two 3D examples are presented in Section 4.4.4.

4.4.1. PROBLEM FORMULATION

For the numerical examples a mechanical optimization problem is considered with minimum compliance objective and a volume constraint. The density design variables are the input design field $\mathbf{x}$, which turn into the smoothed design $\tilde{\mathbf{x}}$, and the final geometric filtered design field is denoted by $\hat{\mathbf{x}}$. Note that the framework also works for other objectives and constraints.

The optimization problem is thus as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}}: \quad & C(\hat{\mathbf{x}}) = \mathbf{u}^T \mathbf{K}(\hat{\mathbf{x}})\mathbf{u} \\ \text{subject to}: \quad & \mathbf{K}(\hat{\mathbf{x}})\mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \frac{V(\hat{\mathbf{x}})}{V^*} - 1 \leq 0 \\ & 0 \leq x_e \leq 1 \quad \text{for} \quad e = 1, \dots N_{\text{el}}. \end{aligned} \tag{4.19}$$

In here $\mathbf{K}$, $\mathbf{u}$ and $\mathbf{f}$ denote the finite element system stiffness matrix, displacement vector and mechanical load vector, respectively. C is the compliance, the current design volume is V, the maximum allowed volume is V^*, and the number of elements in the domain is N_{el}.

In milling and jetting filters the smoothed input design is turned into the filtered design by adding material downstream. Printing filters turn some input solid regions into void upon filtering, to remove unprintable features from the design. For the former, as explained in Giele *et al.* [29], numerical stability can be improved when volume evaluations are performed on the filtered design $\hat{x}$, while the unfiltered smoothed design $\tilde{x}$ is used for the finite element analysis.

Thus, for the milling and jetting filter, the optimization problem is modified as:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}}: \quad & C(\tilde{\mathbf{x}}) = \mathbf{u}^T \mathbf{K}(\tilde{\mathbf{x}})\mathbf{u} \\ \text{subject to}: \quad & \mathbf{K}(\tilde{\mathbf{x}})\mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \frac{V(\hat{\mathbf{x}})}{V^*} - 1 \leq 0 \\ & 0 \leq x_e \leq 1 \quad \text{for} \quad e = 1 \dots N_{\text{el}}. \end{aligned} \tag{4.20}$$

The proposed geometrical filters described in Section 4.2 sometimes put significant restrictions on the design, causing undesirable convergence behaviour. In our approach, the convergence is improved by gradually activating the the geometric filters, similar to van de Ven *et al.* [16]. This is done by mixing the smoothed input design $\tilde{x}$ and the geometric filtered design $\bar{x}^{(\text{out})}$ with a general post filter function $\mathcal{G}^{(\text{g})}$:

$$\hat{x}_e = \mathcal{G}(\tilde{x}, \bar{x}^{(\text{out})}) = (1-\eta)\tilde{x}_e + \eta \bar{x}_e^{(\text{out})}, \tag{4.21}$$

where $\eta \in [0,1]$ is the scaling parameter. In this chapter, η is linearly increased from 0 to 1 in the first 25 optimization iterations.

For the Young's modulus mapping in each element e in Eqs. (4.19) and (4.20) the modified SIMP interpolation scheme proposed by Sigmund [5] is used, i.e.:

$$E(\hat{x}_e) = E_{\text{min}} + \hat{x}_e^p (E_{\text{max}} - E_{\text{min}}), \tag{4.22}$$

with penalization exponent $p = 3.0$, minimum and maximum Young's moduli $E_{\text{min}} = 10^{-9}$ and $E_{\text{max}} = 1$. For the finite element analysis, 4-node quadrilateral elements with bilinear shape functions are used for the 2D structured mesh, 3-node

triangular elements with linear shape functions are used for the 2D unstructured mesh, and 8-node hexahedral elements with trilinear shape functions are used in 3D.

The 2D problem is implemented as an extension to the 88 line MATLAB code by Andreassen *et al.* [25], supplemented with the MMA optimizer by Svanberg [26] with standard MMA parameter values. The 3D problem is implemented as an extension to the PETSc code by Aage *et al.* [27]. The optimization is terminated after 250 iterations, by which a desired level of convergence was always reached. The density filter radius is 1.5 element length $l^{(x)}$. An overview of the parameter values is given in Table 4.1. Using more IPs would lead to a better discretized approximation of the detection region, but would also increase computational costs. Bigger radii would lead to smoother optimization, but also include more information from outside of the detection region. Consequently, these values are fixed, and their influence is not studied in this work.

For the new filter configuration design variables it is important to mention the used scaling and MMA values. In our implementation $\alpha_m = 0$ implies a vertical milling in the direction of positive y-axis, and $\alpha_m = \pi/2$ implies a horizontal milling in the direction of positive x-axis. In our jetting filter implementation the position variables $\alpha_j^{(x)}$ and $\alpha_j^{(y)}$ are normalized with the domain dimensions to range between 0 and 1. In our implementation $\alpha_p = 0$ implies vertical (positive y-axis), and $\alpha_p = \pi/2$ implies vertical (positive x-axis). In addition, for the base plate position α_b is defined with respect to the domain center, with $\alpha_b = 0$ implies the bottom of the domain, and $\alpha_b = 1$ the top. The filter configuration variables can have high sensitivities compared to the density variables, hence move limits are applied. The used move limit for the milling angle α_m is 0.2 [rad] (or approximately 11°), for the jetting location α_j it is 0.2, for the printing orientation α_p it is 0.2 [rad] (or approximately 11°), and for the printing baseplate height α_b it is 0.03.

4.4.2. 2D STRUCTURED MESH

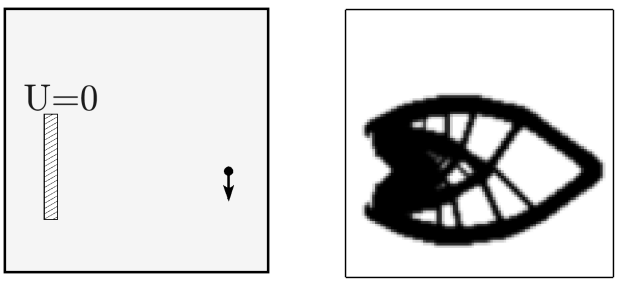
The first numerical example considers a 2D cantilever beam problem. This is a problem with well-known optimization behaviour. The boundary conditions and loading can be seen in Figure 4.16a. A volume constraint of $V^* = 0.25$ is used, and the design domain is discretized by 100×100 elements. For comparison, the result of the standard optimization without the geometric filters is shown in Figure 4.16b.

Next, the tests are done including the geometric filters. Two milling directions, three jet positions, or one printing orientation are considered. The initial filter configuration design variables values are given in Table 4.2, and given in the related figures. First, the example filters are applied without optimization of the filter configuration design variables. The resulting designs are shown in Figure 4.17. As can be seen, the designs obey the geometric constraints. Since the designs are highly restricted by the filters with fixed filter configuration design variables, the objective values are increased considerably, respectively by factors 1.420, 1.467, and 1.225 compared to the reference design in Figure 4.16.

Next, the example filters are applied including optimization of the filter

Table 4.1: Summary of used parameter values for numerical examples

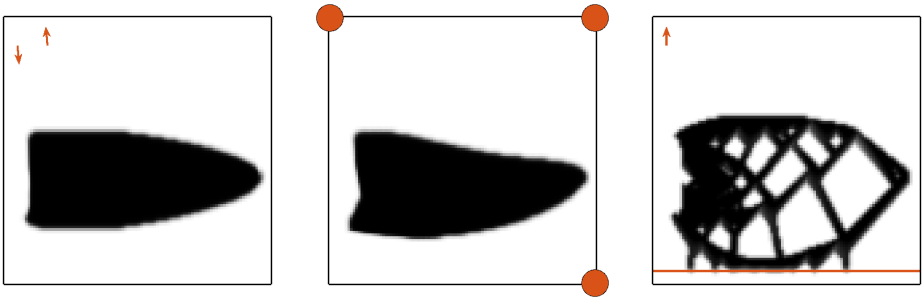
Parameter	All		Milling		Jetting		Printing	
	2D	3D	2D	3D	2D	3D	2D	3D
Filter radius R	1.5 $l^{(x)}$							
SIMP exponent p	3.0							
E_{min}	10^{-9}							
E_{max}	1							
Poisson's ratio ν	0.3							
Nr of iterations	250							
IP distance $l^{(IP)}$			1.0 $l^{(x)}$	1.0 $l^{(x)}$	1.0 $l^{(x)}$	1.0 $l^{(x)}$	1.5 $l^{(x)}$	1.9 $l^{(x)}$
IP radius $R^{(IP)}$			1.5 $l^{(x)}$	1.75 $l^{(x)}$	1.5 $l^{(x)}$	1.75 $l^{(x)}$	1.1 $l^{(x)}$	1.25 $l^{(x)}$
OP radius $R^{(OP)}$			1.5 $l^{(x)}$	1.75 $l^{(x)}$	1.5 $l^{(x)}$	1.75 $l^{(x)}$	0.5 $l^{(x)}$	0.5 $l^{(x)}$
OP radius $R^{(OP;BP)}$							4.5 $l^{(x)}$	4.5 $l^{(x)}$
P_1			-2		-2			
P_2			-4		-4			
P_3							20	
ϵ							10^{-4}	
Overhang θ							45[°]	
MMA movelimits α			0.2[rad]		0.2	0.1	for α_p: 0.2[rad] for α_b: 0.03	



(a) Problem definition.

(b) Reference design. $C := C_{ref}$

Figure 4.16: The 2D cantilever beam compliance problem. In (a) the load and boundary conditions are shown. The point load is applied 5% from the right edge, and 40% from the bottom. The fully clamped region is located 10% from the left edge, 40% from the bottom, and has a width of 2% and a height of 30%. In (b) the optimization result without geometry filters is shown.



(a) Milling filter. $C = 1.420C_{\text{ref}}$

(b) Jetting filter. $C = 1.467C_{\text{ref}}$

(c) Printing filter. $C = 1.225C_{\text{ref}}$

Figure 4.17: Results of the 2D cantilever beam compliance problem with fixed filter configuration design variables, obtained with (a) milling filter (b) jetting filter (c) printing filter, with C_{ref} taken from Figure 4.16b.

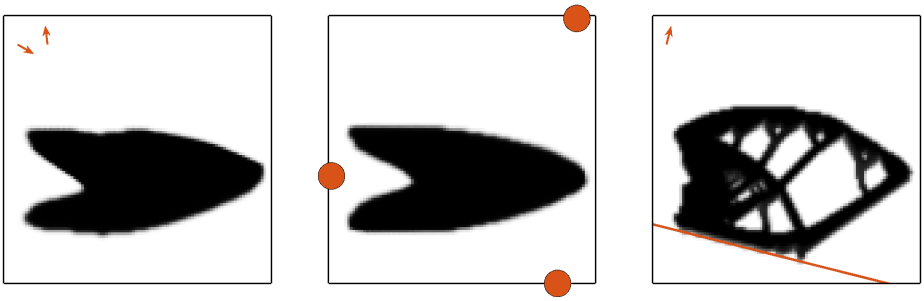
configuration design variables. The initial values for the filter configuration design variables are set to be their fixed values from the previous numerical example. The optimized designs can be seen in Figure 4.18, and the resulting filter configuration design variable values are reported in Table 4.2. As can be seen, the filter configuration design variables values have changed favorably. The relative objective values have improved with respect to the fixed filter variable example, specifically 1.381, 1.318, and 1.074 compared to the reference design. In the milling case, one milling orientation now has adjusted access to the left side of the component. It probably could lead to a better performance if the milling orientation would have rotated further, but apparently a local optimum was reached. In the jetting case, one jet moved to the left of the structure, allowing for a hole here and thus allocating the material more efficiently. In the printing case, allowing changes in printing orientation and baseplate height allowed a significantly improved objective value. However the nature of this filter, e.g. with elements close to the baseplate having a vast influence on the design, makes the optimization prone to local optima.

4.4.3. 2D UNSTRUCTURED MESH

Recall that an additional advantage of the proposed method is that it decouples the geometric filters from the mesh. Therefore, the method can readily be applied to optimization with an unstructured mesh.

For the geometric filters we have presented, the most important difference between an unstructured and a structured mesh is that the element size is not constant. In our application this is taken into account by assigning a weight to every element $w_e^{(\text{area})}$, which is calculated with element size normalized with respect to the average element size. This weight is subsequently included in Eq. (4.1) as follows (and similarly in Eq. (4.4)):

$$w_e^{(\text{IP})} = \max(0, R^{(\text{IP})} - \|\mathbf{c}_e - \mathbf{c}^{(\text{IP})}\|)\, w_e^{(\text{area})}. \tag{4.23}$$



(a) Milling. $C = 1.381 C_{\text{ref}}$

(b) Jetting. $C = 1.318 C_{\text{ref}}$

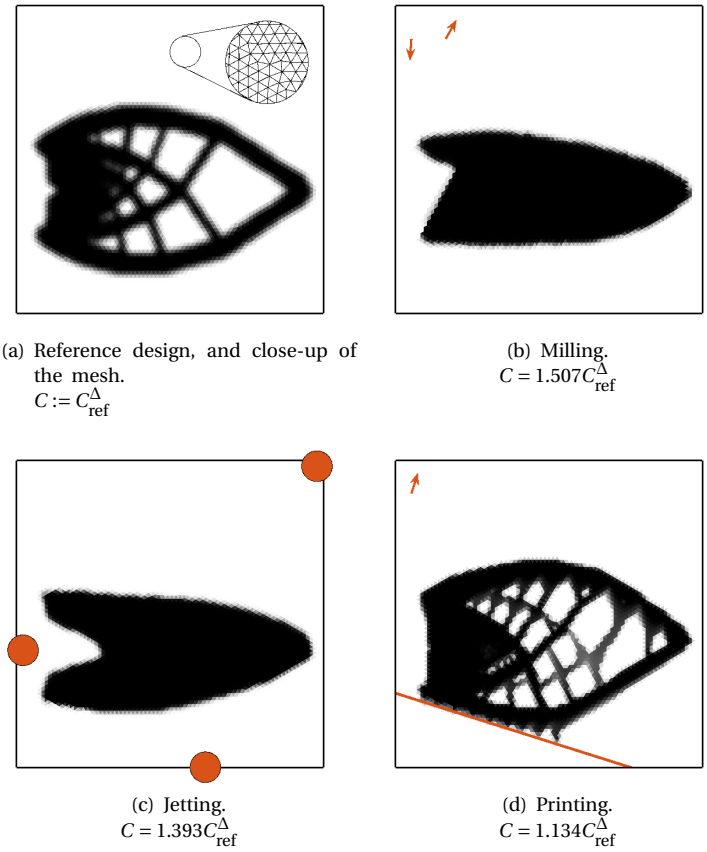
(c) Printing. $C = 1.074 C_{\text{ref}}$

Figure 4.18: Results of the 2D cantilever beam compliance problem with optimized filter configuration design variables, obtained with (a) milling filter (b) jetting filter (c) printing filter, with C_{ref} taken from Figure 4.16b.

Table 4.2: Summary of filter configuration design variables values for 2D tests. For clarity the orientation design variables for milling and printing ($\boldsymbol{\alpha}_m$ and α_p) are described in vector notation, even though these are optimized as angles. The jetting design variables ($\boldsymbol{\alpha}_j$) are described in coordinates.

Filter	Variable	Fixed	Optimized	Unstructured
Milling	α_1	[0.09 ; -1.00]	[0.87 ; -0.50]	[-0.07 ; -1.00]
	α_2	[-0.09 ; 1.00]	[-0.10 ; 0.99]	[0.49 ; 0.87]
Jetting	α_1	[0.00 ; 1.00]	[0.01 ; 0.40]	[0.02 ; 0.38]
	α_2	[1.00 ; 1.00]	[0.93 ; 0.99]	[0.98 ; 0.98]
	α_3	[1.00 ; 0.00]	[0.86 ; 0.00]	[0.62 ; 0.00]
Printing	α_p	[0.00 ; 1.00]	[0.24 ; 0.97]	[0.31 ; 0.95]
	α_b	[0.05]	[0.11]	[0.10]

A mesh with triangular elements was created, with approximately 13000 elements. The results for our method on an unstructured mesh are shown in Figure 4.19, and the improved filter configuration design variables are reported in Table 4.2. As can be seen, the results are similar to those reported in Figure 4.18, in geometric layout, optimized filter configuration design variable values, and objective values. This demonstrates versatility of the proposed method being suitable for an unstructured mesh. The grey elements in the printing case might be caused by the choice for the smooth maximum and minimum operations, but these regions might require further research.



(a) Reference design, and close-up of the mesh. $C := C^{\Delta}_{\text{ref}}$

(b) Milling. $C = 1.507C^{\Delta}_{\text{ref}}$

(c) Jetting. $C = 1.393C^{\Delta}_{\text{ref}}$

(d) Printing. $C = 1.134C^{\Delta}_{\text{ref}}$

Figure 4.19: The results of the 2D cantilever beam compliance problem with optimized filter configuration design variables on an unstructured mesh, obtained with (a) no geometric filter (b) milling filter (c) jetting filter (d) printing filter.

4.4.4. 3D NUMERICAL EXAMPLES

Next, the method is applied to two 3D problems discritized with a structured mesh. The first problem is a 3D cantilever beam problem with predictable optimization behaviour. The boundary conditions and applied loading can be seen in Figure 4.20a. A volume constraint of $V^* = 0.10$ is used, and a discretization of $144 \times 192 \times 96$ elements is considered. For comparison, the result of the standard optimization without geometric filters is shown in Figure 4.20b.

Again, the example filters are first applied with fixed filter configuration design variables. Two milling directions, four jet sources, and one printing orientation are considered. The initial values are shown on the left in Table 4.3, and in Figure 4.20.

The resulting designs can be seen in Figure 4.20c-e. As expected, the design

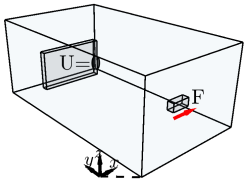
Table 4.3: Summary of filter configuration design variables values for tests. Again, for clarity the orientation design variables for milling and printing ($\boldsymbol{\alpha}_m$ and α_p) are described in vector notation, even though these are optimized as angles. The jetting design variables ($\boldsymbol{\alpha}_j$) are described in coordinates.

Filter	Variable	Beam problem		Bridge problem	
		Fixed	Optimized	Fixed	Optimized
Milling	α_1	[0; 0.87;0.5]	[0.33; 0.20; 0.92]	[1;0;0]	[0.84 ;0.00;-0.55]
	α_2	[0;-0.87;0.5]	[-1.66;-0.20;-0.97]	[-1;0;0]	[-0.91;0.40;-0.08]
Jetting	α_1	[0.075;0.10;0.05]	[1.88;1.01;0.00]	[1.40;-0.30;0.53]	[1.50;0.42;0.42]
	α_2	[0.075;0.90;0.05]	[2.25;1.71;1.50]	[1.40; 1.30;0.53]	[0.97;2.42;1.56]
	α_3	[0.925;0.10;0.05]	[0.00;2.49;1.50]	[-0.40; 3.30;0.53]	[0.24;2.10;1.56]
	α_4	[0.925;0.90;0.05]	[0.28;0.00;0.00]	[-0.40;-0.30;0.53]	[-0.50;0.43;0.42]
Printing	α_p	[0;0;1]	[-0.01;-0.14;0.99]	[0;0;1]	[0;0;1]
	α_b	[0.05]	[0.13]	[0.02]	[0.03]

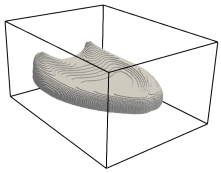
freedom is severely restricted by the filters when using fixed filter configuration design variables. The relative objective values increased, by factors 1.686, 1.385, and 1.258 respectively. It would take numerous attempts for a designer to manually find the best filter configurations for these cases.

Next, the example filters are applied with optimization of the filter configuration design variables. The initial values for the filter configuration design variables are the respective fixed values depicted in Figure 4.20c-e. The results can be seen in Figure 4.20f-h. We observe that the filter configuration design variables have changed significantly, and the resulting layouts are more similar to the unrestricted design shown in Figure 4.20b. For the milling and jetting filters, the relative objective values have improved significantly by allowing the filter configuration design variables to change, to 1.146 and 1.171 respectively. For the printing filter, just like for the 2D examples, the baseplate has a significant impact on the entire design, and the new relative objective value is 1.272, slightly higher than the fixed filter configuration counterpart. For orientation optimization in combination with overhang constraints/filters, it is known that this problem can be very nonconvex and multimodal in the orientation variables (see e.g. Langelaar [19] and Olsen and Kim [45]). Small changes in baseplate orientation and height, just like small changes in material close to the baseplate, can have big impact on the remaining design.

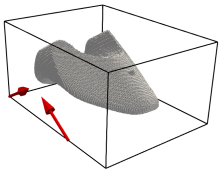
The second 3D problem is the bridge problem, for which the loading and boundary conditions are shown in Figure 4.21a. To ensure that the geometric filters are effective near the top of the domain, a local low volume constraint is applied to a thin region, as previously done in Giele *et al.* [8]. A volume constraint of $V^* = 0.25$ of the unconstrained design domain is used, and a discretization of $64 \times 192 \times 72$ elements. For comparison, the result of the standard optimization without geometric filters is shown in Figure 4.21b.



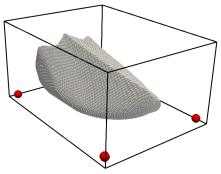
(a) Load and boundary conditions. The problem has dimensions L_x=1.5, L_y=2, and L_z=1. The load is applied in x-direction at: $x \in [0.73, 0.77]$, $y \in [0.19, 0.21]$, and $z \in [0.39, 0.41]$. The fully clamped part is located at: $x \in [0.375, 1.125]$, $y \in [1, 80, 1.90]$, and $z \in [0.2, 0.6]$.



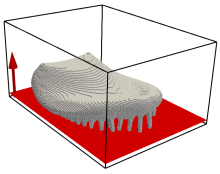
(b) Reference design without geometry filter. $C := C_{\text{ref}}$



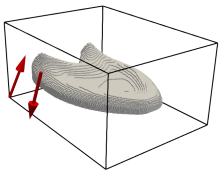
(c) Result with milling filter, with fixed α. $C = 1.686 C_{\text{ref}}$



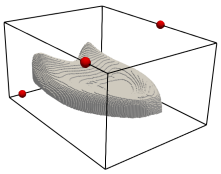
(d) Result with jetting filter, with fixed α. $C = 1.385 C_{\text{ref}}$



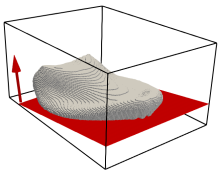
(e) Result with printing filter, with fixed α. $C = 1.258 C_{\text{ref}}$



(f) Result with milling filter, with optimized α. $C = 1.146 C_{\text{ref}}$



(g) Result with jetting filter, with optimized α. $C = 1.171 C_{\text{ref}}$



(h) Result with printing filter, with optimized α. $C = 1.272 C_{\text{ref}}$

Figure 4.20: The 3D cantilever beam compliance problem definition and optimization results, with the geometric filters with both fixed and optimized filter configuration design variables. Projected with a 0.5 density threshold value.

Two milling directions, four jet sources, and one printing orientation are considered, and the jet sources are allowed outside the design domain. Again, first the example filters are applied with fixed filter configuration design variables. The

initial values for the filter configuration design variables are shown in Table 4.3, and in Figure 4.21. The optimized designs are shown in Figure 4.21c-e. The initial filter variables were chosen such that even when they are fixed they are not too restrictive. However the objective values are still increased, by factors 1.190, 1.522, and 1.316 respectively. In Figure 4.21f-h the results can be seen for the three example filters with optimization of the filter configuration design variables. The initial values for the filter configuration design variables are the respective fixed values depicted in Figure 4.21c-e. The relative objective values have changed to 1.311, 1.307, and 1.363 respectively.

At these 3D tests the importance of the initial filter settings is noticeable. For both the milling and printing filter, the compliance is increased for the result with the optimized α. Since the initial filter settings were chosen more favorable, less improvement in the performance is obtained. The large influence of initialization, and the non-linearity of the problem, probably resulted in a local minimum with a worse performance.

4.5. DISCUSSION

The proposed method was found to be effective and adaptable for different geometric requirements. Still several limitations are present with potential improvements.

First, the optimization of the filter configuration design variables may converge to an unfavorable local optimum and is influenced by the initial filter setting values and several other parameters. Examples of this are the results of Figure 4.18, where local optima were reached. In our experience, the most influential parameters are the initial filter configuration values, the MMA movelimits, and the scaling parameter η. Big changes in filter configuration design variables throughout the optimization are unlikely as it is related to the current topology. Especially the nature of the printing filter, where material right above the baseplate can alter the output design significantly, makes this problem prone to local minima.

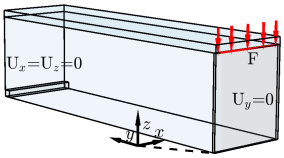
Secondly, for well chosen initial filter configurations, the improvement can be limited, see Figure 4.21. In this case, the filter adds unnecessary complexity to the optimization, which in some cases even resulted in (slightly) worse objective values.

Thirdly, the general concept of projecting densities to a mesh in a mesh-independent manner may also be of use in other operations, such as moving components through a mesh in feature mapping TO methods, which offers opportunities for future research.

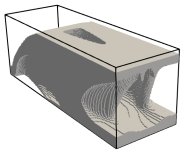
Lastly, performance comparisons should be made to existing methods that allow for optimization of the filter configuration design variables, specifically [45–48].

4.6. CONCLUSION

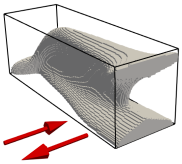
A general framework is presented applicable for any geometric filtering operations. In this novel scheme, the filter operation is decoupled from the mesh using interpolated densities. This allows for a continuous description of the filter configuration, and subsequently for introducing new filter configuration design variables that can be



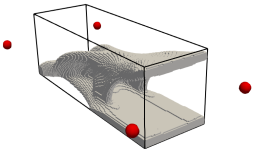
(a) Boundary conditions for the 3D bridge problem. The problem has dimensions L_x=1, L_y=3, and L_z=1.0625, where the top 0.0625 is involved in the local volume constraint. There is a symmetry boundary condition at the $x = 0$ plane, with a line load on top. The simply supported part is located at: $x \in [0,1]$, $y \in [2.98,3.0]$, and $z \in [0,0.02]$.



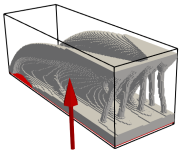
(b) Reference design without geometry filter. $C := C_{\text{ref}}$



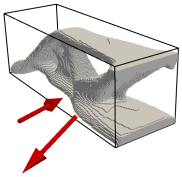
(c) Result with milling filter, with fixed α. $C = 1.190C_{\text{ref}}$



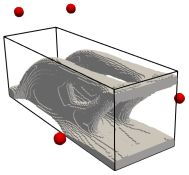
(d) Result with jetting filter, with fixed α. $C = 1.522C_{\text{ref}}$



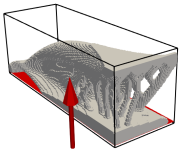
(e) Result with printing filter, with fixed α. $C = 1.316C_{\text{ref}}$



(f) Result with milling filter, with optimized α. $C = 1.311C_{\text{ref}}$



(g) Result with jetting filter, with optimized α. $C = 1.307C_{\text{ref}}$



(h) Result with printing filter, with optimized α. $C = 1.363C_{\text{ref}}$

Figure 4.21: The 3D bridge compliance problem definition and optimization results, with the geometric filters with fixed filter configuration design variables. Projected with a 0.5 density threshold value.

optimized simultaneously with the topology with consistent sensitivities leading to increase in design freedom and performance.

As shown by the three numerical examples, the framework can be applied to

various different geometric filters. The numerical examples showed that the initial filter configuration chosen has a strong influence on the design. Allowing this configuration to be optimized simultaneously with the topology, as enabled by the proposed method, can thus result in significantly improved designs with superior objective values. For examples with a well chosen initial filter configuration, the improvement in objective value is limited, and in some cases the method and additional design freedom may actually add complexity to the optimization resulting in convergence to (slightly) inferior local optima.

The biggest challenge and thus recommendation for future work is therefore regarding prevalence of local optima. Most noticeable in the printing example, but also in other examples, small changes in filter configuration may weaken the design significantly.

Finally, the chapter also introduces a general framework on how to approach geometric filters. Any geometry requirement can be enforced through wise consideration of the detection region. Any initial design decision, can be optimized with expended design freedom. Through this philosophy, engineers are encouraged and enabled to develop novel adaptive filters for their specific use cases.

4.7. SENSITIVITY ANALYSIS

This section gives an example of how the sensitivities can be calculated for the optimization of the filter configuration design variables. All equations can be traced to Figure 4.10 and 4.15. By simply following the steps in reverse, and using the chain rule, the sensitivities with respect to $\tilde{x}$ and α can be calculated.

In the remained, only one part is shown, for the part described in Figure 4.5. If the sensitivities of compliance C (or any other response) with respect to an input density, $\chi_i^{(\mathrm{IP})}$, are known as $\mathrm{d}C/\mathrm{d}\chi_i^{(\mathrm{IP})}$, the sensitivities for the density design variables can be calculated with the chain rule and Eq. 4.3:

$$\begin{aligned} \frac{\mathrm{d}C}{\mathrm{d}\tilde{x}_e} &= \sum_{i=1}^{N_{\mathrm{IP}}} \frac{\mathrm{d}C}{\mathrm{d}\chi_i^{(\mathrm{IP})}} \frac{\mathrm{d}\chi_i^{(\mathrm{IP})}}{\mathrm{d}\tilde{x}_e} \\ \text{with} \quad \frac{\mathrm{d}\chi_i^{(\mathrm{IP})}}{\mathrm{d}\tilde{x}_e} &= \check{w}_{e,i}^{(\mathrm{IP})}. \end{aligned} \tag{4.24}$$

Here, N_{IP} refers to the total number of IPs. Similarly, to obtain the sensitivities with respect to the position of a single IP omitting the subscript i for clarity, first the sensitivities for $\check{w}_e^{(\mathrm{IP})}$ can be calculated with the chain rule and Eq. 4.3:

$$\begin{aligned} \frac{\mathrm{d}C}{\mathrm{d}\check{w}_e^{(\mathrm{IP})}} &= \frac{\mathrm{d}C}{\mathrm{d}\chi^{(\mathrm{IP})}} \frac{\mathrm{d}\chi^{(\mathrm{IP})}}{\mathrm{d}\check{w}_e^{(\mathrm{IP})}} \\ \text{with} \quad \frac{\mathrm{d}\chi^{(\mathrm{IP})}}{\mathrm{d}\check{w}_e^{(\mathrm{IP})}} &= \tilde{x}_e. \end{aligned} \tag{4.25}$$

Next, the sensitivities for $w_e^{(\mathrm{IP})}$ can be calculated with the chain rule and Eq. 4.2:

$$\begin{aligned} \frac{\mathrm{d}C}{\mathrm{d}w_e^{(\mathrm{IP})}} &= \frac{\mathrm{d}C}{\mathrm{d}\check{w}_e^{(\mathrm{IP})}} \frac{\mathrm{d}\check{w}_e^{(\mathrm{IP})}}{\mathrm{d}w_e^{(\mathrm{IP})}} \\ \text{with} \quad \frac{\mathrm{d}\check{w}^{(\mathrm{IP})}}{\mathrm{d}w_e^{(\mathrm{IP})}} &= \frac{1}{\sum_{j=1}^{N_{\mathrm{el}}} w_j^{(\mathrm{IP})}} + \frac{-w_e^{(\mathrm{IP})}}{\left(\sum_{j=1}^{N_{\mathrm{el}}} w_j^{(\mathrm{IP})}\right)^2}. \end{aligned} \tag{4.26}$$

Subsequently, the sensitivities for $\mathbf{c}^{(\mathrm{IP})}$ can be calculated with the chain rule and Eq. 4.1:

$$\begin{aligned} \frac{\mathrm{d}C}{\mathrm{d}\mathbf{c}^{(\mathrm{IP})}} &= \frac{\mathrm{d}C}{\mathrm{d}w_e^{(\mathrm{IP})}} \frac{\mathrm{d}w_e^{(\mathrm{IP})}}{\mathrm{d}\mathbf{c}^{(\mathrm{IP})}} \\ \text{with} \quad \frac{\mathrm{d}w^{(\mathrm{IP})}}{\mathrm{d}\mathbf{c}^{(\mathrm{IP})}} &= \begin{cases} 0\,, & \text{if} \quad R^{(\mathrm{IP})} - \|\mathbf{c}_e - \mathbf{c}^{(\mathrm{IP})}\| < 0, \\ \frac{\mathbf{c}_e - \mathbf{c}^{(\mathrm{IP})}}{\|\mathbf{c}_e - \mathbf{c}^{(\mathrm{IP})}\|}\,, & \text{else.} \end{cases} \end{aligned} \tag{4.27}$$

For computational reasons, in cases of $\mathbf{c}_e = \mathbf{c}^{(\mathrm{IP})}$ it is possible to add a small offset preventing division by 0. Lastly, in a similar manner, the sensitivities for Eqs. 4.4-4.18 can be computed. Specifically the derivatives from operation $\mathscr{H}$ (shown in Eqs. 4.8, 4.12, and 4.14) will result in the sensitivities for α.

5

FEATURE MAPPING

A novel feature mapping topology optimization method is presented, allowing for the creation of features with highly flexible shapes. The method easily integrates with conventional density-based formulations. Feature shapes are implicitly described by NURBS control points. The feature shape dictates the locations of two sets of projection points to represent the solid void boundaries. At these projection points, density values are projected onto a finite element mesh. The method optimizes feature shapes in a gradient based manner, while allowing more specific control of the feature shapes than classical level set methods. Several feature fields can be combined to create a final output design. It is found that the eminent flexibility of the NURBS based feature definition is a benefit but also requires additional regularization to guarantee stability of the optimization.

This chapter is based on Giele *et al.* [50].

5.1. INTRODUCTION

Topology optimization (TO) is a computational design method to determine the distribution of material, such that the geometric layouts of components with superior performance can be determined. Several distinct TO approaches exist, where the most prevalent ones are density-based methods (see e.g. Sigmund and Maute [51]) and level-set (LS) methods (see e.g. Van Dijk *et al.* [52]). Density methods focus on the distribution of material (represented by a density field) in a discretized domain, benefiting from a of lack of restrictions on design changes throughout the design domain, while sensitivity information is available everywhere in the design domain. In LS methods the design is generated by moving the boundary between solid and void regions, benefiting from having a clear boundary description.

Another more recent category of TO approaches is known as feature mapping methods (FMMs), see Wein *et al.* [53] for an overview. In FMMs, solid (or void) features with specified geometric shapes are usually mapped onto a finite element mesh. While the shape of the features is often fixed, typically with limited options such as (hyper) ellipses or rectangles in 2D and their counterparts in 3D (Wein *et al.* [53]), their position, orientation, and size are optimized. The moving feature boundaries bear similarities to LS methods with a more strict shape restriction of each feature, such that simpler design outcomes are ensured. Final component designs are often obtained through combining and overlapping these features.

Several challenges occur in creating components by combining features. First, there is limited to no control over the final topology. Since designs in current FMMs are typically created with many features, the topology can easily change. For specific applications topology control can be required or desired, see e.g. He *et al.* [54].

Secondly, for designs created with fewer features, the design space is limited. One way to allow for complex designs with fewer components is to allow some flexibility in the feature shape. In the method of Moving Morphable Components (MMC) framework, this has been addressed e.g. by Shannon *et al.* [55] with curved features with varying thickness defined using generalized Bézier curves, or by Zheng and Kim [56] with NURBS (non-uniform rational basis splines) shaped components (although only one fixed feature shape is considered). To the best of our knowledge, the feature shapes in the aforementioned studies are almost unchangeable, limiting the design freedom of the resulting layout. An exception is e.g. Zhang *et al.* [57] the shape of features is optimized with closed B-splines, however only limited shape flexibility is allowed in this parameterization. Other examples of limited spline based TO include Greifenstein *et al.* [58], Guo *et al.* [59], and Zhu *et al.* [60].

Thirdly, it is beneficial if the topology optimized design geometry can easily be converted to a common CAD (computer aided design) description, which is often required for further processing of the design. For this purpose, TO methods have been proposed that are focused on explicit output shapes. For instance, Schmidt *et al.* [61] consider a semi-analytical gradient-based optimization of exact CAD models using intermediate field representations, and Yi and Kim [62] identify the boundaries of TO results to extract basic parametric features to close the gap with parameterized CAD models.

To address the three aforementioned gaps simultaneously, this chapter studies and presents a novel FMM framework for creating features with flexible shapes using a NURBS parameterization, such that feature shapes are adjustable during the optimization for more design freedom and convenient for post-processing of designs.

The steps of the proposed method starting from design variables to a feature geometry are illustrated in Figure 5.1. First, we define the parameterization of the feature's boundary shape. We opt to use the locations and weights of the *control points* (CPs) as design variables. Next, the discretized feature boundary curve defines the first set of *projection points* (PPs), whereas a second set is of PPs is located at an offset outside the feature boundary. These two sets of PPs are used to project a solid respectively void density value onto the the underlying mesh, eventually creating an aggregate output density field. This use of projection points to relate the NURBS curve to the density field is the main novel concept we introduce, and allows for a convenient integration with conventional density-based TO procedures. The performance of the design projected onto the mesh is obtained through finite element analysis, and using a gradient based TO process, the design variables are optimized iteratively, so that the feature shapes are adapted towards the optimal geometry.

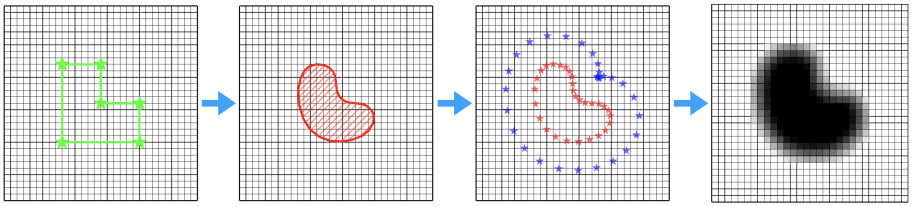


Figure 5.1: A schematic illustration of the method, for creating one feature in a 2D design domain. First, design variables (*control points* location and weights) represented by green stars are used to create a feature shape. Next, the feature shape is discretized, and two sets of *projection points* (represented by red and blue stars) are defined. Lastly, these two sets of projection points are projected onto the finite element mesh to create an output density field.

An important benefit of the proposed method is that intricate shapes can be attained by each feature, such that complex output designs can be created, even when using relatively few features. Better control over the topology is achievable, and an easier conversion to a useful output design is facilitated through the use of the NURBS description. The density field output allows the proposed novel method to build on the existing standard density-based TO, and even can be combined with standard density fields, where both the standard density field and the feature associated density field together describe the component. Since the design variables, i.e. the CP locations and weights, are mathematically related to element densities, gradient-based optimization can be used with conventional density-based adjoint sensitivity analysis, building on density-based TO implementations. Naturally, the

method can be applied both to solid and void features. An opacity design variable is added to allow for features to disappear (similar to Norato *et al.* [7]).

This chapter focuses on the 2D version of the new method, both for clarity and because it already presents a considerable number of novel aspects to investigate. An extension to 3D is out of the scope of the current study, but the extension of the method for 3D cases is discussed in Section 5.4. Finally, it should be mentioned that the proposed method also poses new challenges related to shape restriction, therefore extra shape regularization schemes are extensively outlined in this study.

5.2. METHOD

For a simple and clear description of the method, solid features are considered within a design domain discretized with a structured mesh. Section 5.2.1 presents the procedure for creating NURBS shapes. Section 5.2.2 describes how the NURBS shapes are projected onto the mesh as a density field. Lastly, several regularization schemes are proposed in Section 5.2.3 to improve the optimization stability.

5.2.1. NURBS WITH CONTINUOUS SHAPE FUNCTIONS

This section describes the steps to obtain feature shapes from NURBS *control points* (CPs), defined by design variables. NURBS are a generalization of B-splines, while B-splines are a generalized form of the Bézier curve. NURBS represent shapes with great versatility, an example of an open NURBS curve is shown in Figure 5.2, and an example of a discretized NURBS curve is shown in Figure 5.3. For more background information on NURBS, the reader is referred to e.g. Piegl and Tiller [63] and Austin Cottrell *et al.* [64].

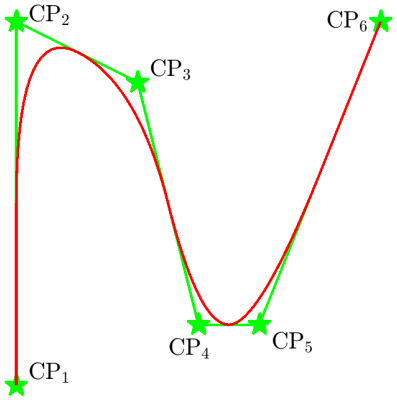


Figure 5.2: An example of an open NURBS curve (red) created from six CPs (green stars).

As outlined in the introduction, the proposed method defines feature shapes using CPs. Each CP has a specified local influence on the boundary curve. The

CP locations and weights, together with appropriate basis functions, can describe a complex shape. As design variables, we introduce the arrays $\mathbf{p}^{(x)}$, $\mathbf{p}^{(y)}$, and $\mathbf{p}^{(w)}$, which represent x- and y-coordinates of the CPs and their weights, respectively.

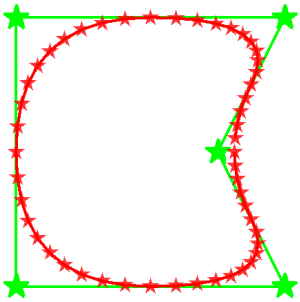


Figure 5.3: A closed boundary curve defining the outline of a feature (red stars), created from five CPs (green stars).

First, the knot vector is the sequence of parameter values that determine where and how the control points affect the NURBS curve. In this work, the knot vector is defined as $\mathscr{U} = [u_1, ..., u_m]$, which is a non-decreasing sequence of real numbers. Here, u_i are the knots, where $i = 1, \ldots, m$ is the knot index, and the knot vector has length $m = k + q + 1$, where q is the polynomial order of the basis functions, and k is the number of basis functions used to construct the NURBS curve. In our application, we will use an unclamped, uniform, knot vector, which means that all knot spans are of equal length, e.g. $\mathscr{U} = [0, 1, 2, ..., 8, 9, 10]$.

Next, the zeroth degree ($q = 0$) basis function $N_{i,0}$ is described as a piece-wise function of curve parameter u:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if} \quad u_i \leq u < u_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

The higher ($q > 0$) degree basis functions $N_{i,q}$ are calculated by recursion:

$$\begin{aligned} N_{i,q}(u) = &\frac{u - u_i}{u_{i+q} - u_i} N_{i,q-1}(u) + \\ &\frac{u_{i+q+1} - u}{u_{i+q+1} - u_{i+1}} N_{i+1,q-1}(u). \end{aligned} \tag{5.2}$$

In our application, second order ($q = 2$) basis functions are used.

With the basis functions $N_{i,q}$, the weight of each control point $p_i^{(w)}$ and its

x-coordinates $p_i^{(\mathrm{x})}$, the NURBS curve x-coordinates, $c^{(\mathrm{x})}(u)$, are calculated as follows:

$$c^{(\mathrm{x})}(u) = \frac{\sum_{i=1}^{k} N_{i,q}(u)\, p_i^{(\mathrm{w})}\, p_i^{(\mathrm{x})}}{\sum_{i=1}^{k} N_{i,q}(u)\, p_i^{(\mathrm{w})}}. \tag{5.3}$$

The coordinates in other dimensions (e.g. $c^{(\mathrm{y})}(u)$) follow similarly. Finally, we note that assuming a knot vector beforehand restricts feature shapes. Sharp corners for example require repeated knot values. In this study, we do not allow repeated knot values and consequently we restrict ourselves to features without intrinsic sharp corners.

Finally, since the feature shape is defined by a closed boundary curve that encloses a finite area, we opt for the curve to be second order continuous at the start, which also coincides with the end of the curve. This is implemented by incorporating three extra CPs, with locations and weights that are identical to the first three CPs. For example, for a 5 CP feature, $p_1^{(\mathrm{x})} = p_6^{(\mathrm{x})}$, $p_2^{(\mathrm{x})} = p_7^{(\mathrm{x})}$, and $p_3^{(\mathrm{x})} = p_8^{(\mathrm{x})}$, and the same holds for $\mathbf{p}^{(\mathrm{y})}$ and $\mathbf{p}^{(\mathrm{w})}$. An example of the basis functions is shown in Figure 5.4a, and the basis function for each unique CP is shown in Figure 5.4b. The reader is referred to Piegl and Tiller [63] for more examples.

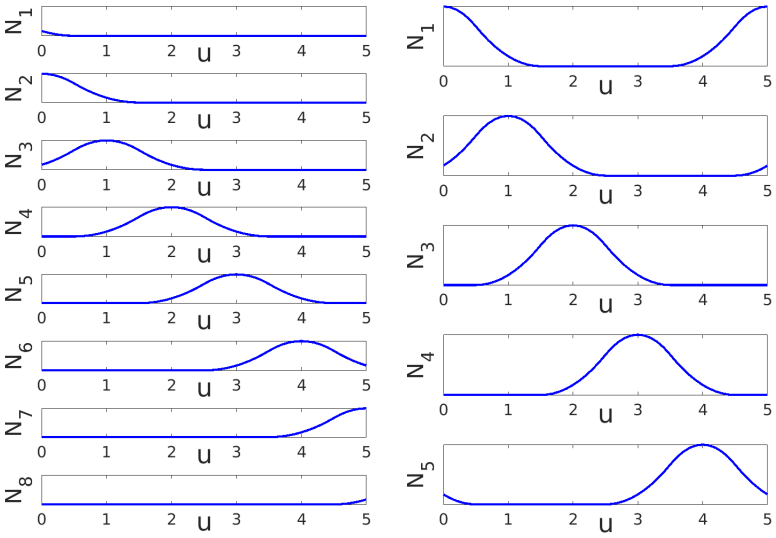
5.2.2. CREATING FEATURES WITH PROJECTION POINTS

It remains to describe the steps to obtain a density field from a NURBS curve. First, the continuous NURBS curve is discretized. In our application, this is done by discretizing the basis functions $N_{i,q}$, with $n^{(\mathrm{CP})}$ points per CP, resulting in a discretized NURBS curve. Next, two sets of *projection points* (PPs) are defined: *'solid PPs'* which are located on the feature's surface, and *'void PPs'* located outside of the feature. Consequently, in our application, the solid PP location, $\dot{\mathbf{c}}_j$, coincides with the discretized NURBS boundary curve described in Eq. (5.3), where integer j denotes the discretized PP index. The void PPs, each denoted with $\mathring{\mathbf{c}}_j$, are located at an offset curve with respect to $\dot{\mathbf{c}}_j$:

$$\mathring{\mathbf{c}}_j = \dot{\mathbf{c}}_j + r\mathbf{n}_j, \tag{5.4}$$

where r is the offset distance. The calculation of unit outward normal $\mathbf{n}_j$ is given in Section 5.2.3. An example of the calculated void PPs is shown in Figure 5.5.

Once all PP coordinates have been calculated using Eqs. (5.3) and (5.4), the feature density field can be created. Each PP projects a specified density value, χ, onto several fixed mesh elements in its close proximity. The density projection is performed similar to the standard convolution filter (Bourdin [22], Bruns and Tortorelli [23]), as shown in Figure 5.6. However, note that since the PPs are detached from the mesh, they are not restricted to coincide with element centers.



(a) 8 basis functions are associated to 5 CPs.

(b) 5 basis functions are associated to 5 CPs.

Figure 5.4: An example of second order basis functions with five CPs. In (a), CP_1 is associated to N_2 and N_7, CP_2 is associated to N_3 and N_8, and CP_3 is associated to N_1 and N_6. In (b), the basis functions for the 1st, 2nd, and 3rd basis functions are combined with the 6th, 7th, and 8th basis functions, respectively.

The spatial weights associated with the projection to each element are calculated using the proximity of the element center $\mathbf{c}_e$ to PP j as follows:

$$w_{e,j}(\mathbf{c}_j) = \max\left(0, R - \|\mathbf{c}_j - \mathbf{c}_e\|\right). \tag{5.5}$$

Here, R is the interpolation radius, and $\mathbf{c}_j$ and $\mathbf{c}_e$ are the coordinates of the PP and the centroid of element e, respectively. $\mathbf{c}_j$ represents both solid ($\dot{\mathbf{c}}_j$) and void ($\mathring{\mathbf{c}}_j$) PPs. The calculation of $w_{e,j}$ ensures a gradual transition from solid to void at the feature boundary, which allows for sensitivity analysis. Note that these weights are related to the density projection, and are not linked to the CP weights defining the NURBS curve, which are controlled by the optimizer. Also note that the subscript e,j refers to element e and PP j, and the comma in the subscripts does not denote differentiation in this chapter. These interpolation weights are subsequently normalized as follows:

$$\breve{w}_{e,j}(\mathbf{c}_j) = \frac{w_{e,j}(\mathbf{c}_j)}{\sum_{e=1}^{N_{\text{el}}} w_{e,j}(\mathbf{c}_j)}, \tag{5.6}$$

where N_{el} is the number of elements in the domain. Next, the output value χ is

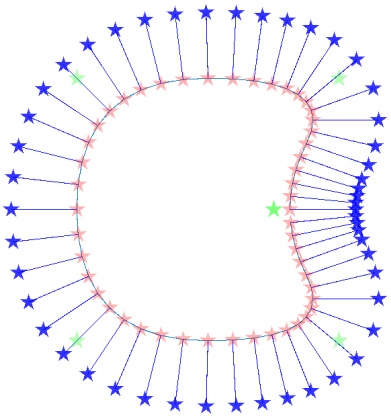


Figure 5.5: An example of the two sets of PPs, with $n^{(\mathrm{CP})} = 10$. The solid PPs (in red) are located on the curve depicted in Figure 5.3. The void PPs (in blue) lie on an offset curve with respect to the solid PP curve.

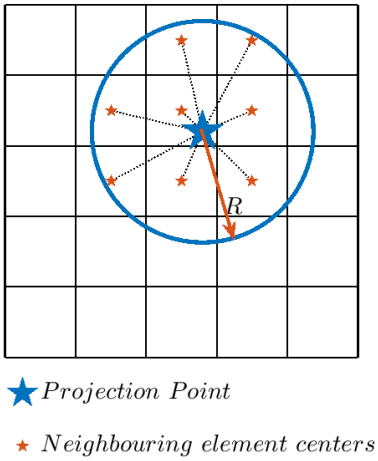


Figure 5.6: Projection points are used to project a density value onto its surrounding elements.

projected to element e resulting in a projected value:

$$\check{x}_{e,j} = \chi \check{w}_{e,j}. \tag{5.7}$$

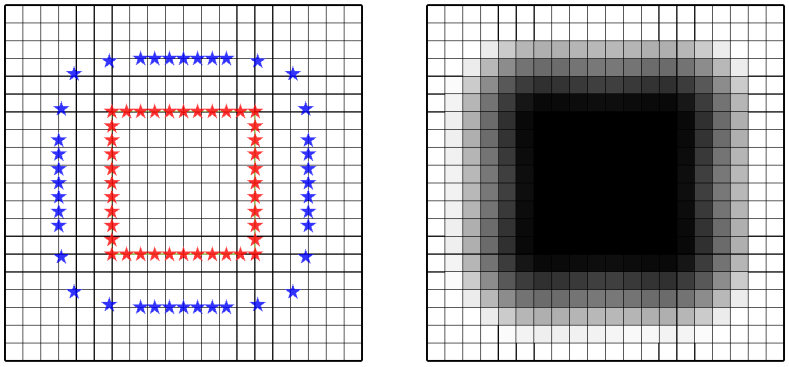
The solid PPs each project a value of $\chi = 1$, the void PPs each project a value of $\chi = 0$. To account for elements getting contributions from multiple PPs, the final output value for element e is obtained by dividing the sum of the projected values

by the sum of the normalized weights of the contributions:

$$\bar{x}_e = \frac{\sum_{j=1}^{n_j} \check{x}_{e,j}}{\sum_{j=1}^{n_j} \check{w}_{e,j}}, \tag{5.8}$$

where n_j is the number of PPs that has projected a (solid or void) density to element e, and $\check{x}_{e,j}$ and $\check{w}_{e,j}$ are the value and weight of the individual contributions. Note that this operation with only one nonzero weight contribution results in $\bar{x}_e = \chi$, thus taking exactly the density value from the only contributing PP.

An example of solid (red) and void (blue) PPs, and the resulting density field projected onto the FE mesh is shown in Figure 5.7. The elements inside the feature, where the element centroid is not close enough to any PPs are assigned a density value of $\bar{x}_e = 1$. The intermediate density zone across the solid-void interface enables obtaining a differentiable boundary motion and is hence required to calculate consistent sensitivities for the feature's boundary shape and location.



(a) PPs (b) Projected density field

Figure 5.7: Solid (red) and void (blue) projection points are used to project density onto the structured mesh, representing a feature. The intermediate density region between the solid and void PPs ensures consistent sensitivities for the feature boundary location.

The equations used for the steps from the PP locations to the output density field $\bar{\mathbf{x}}$ are shown schematically in Figure 5.8. The sensitivities $\partial\bar{\mathbf{x}}/\partial\mathbf{p}^{(x)}$, $\partial\bar{\mathbf{x}}/\partial\mathbf{p}^{(y)}$, and $\partial\bar{\mathbf{x}}/\partial\mathbf{p}^{(w)}$ follow naturally with the chain rule for each operation in Figure 5.8, such that the sensitivities of the objective/constraints with respect to $\mathbf{p}^{(x)}$, $\mathbf{p}^{(y)}$, and $\mathbf{p}^{(w)}$ can be computed. The integration of specific constraints is thus convenient if sensitivities with respect to a density field can be calculated. Note that only elements with intermediate density values have an influence on the boundary, and will thus contribute to the sensitivities.

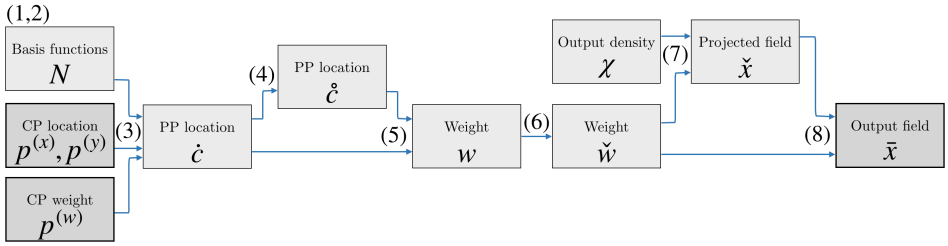


Figure 5.8: The flow diagram illustration of the steps from the PP locations and weights to the feature density field. The numbers represents the equations used for each step. By following the steps backwards, and through the chain rule, the sensitivities follow naturally.

After the density fields for each feature f have been created, denoted here as ${}^f\bar{\mathbf{x}}$, the steps towards the final component design can be taken. In order to exclude unnecessary features, every ${}^f\bar{\mathbf{x}}$ is multiplied with an opacity design variable ${}^f\alpha$, similar to Norato *et al.* [7]. Next, multiple features can be combined with a smooth maximum operation over the feature density fields, which will be applied in Section 5.3.2. This is done through a P-norm:

$$\hat{x}_e = \left(\sum_{f=1}^{N_f} ({}^f\alpha \cdot {}^f\bar{x}_e)^{P_1} \right)^{\frac{1}{P_1}}. \tag{5.9}$$

Here, $P_1 > 0$ is the aggregation parameter, and N_f is the number of features considered. In this smooth maximum operation overlapping features can lead to output values higher than 1, these are normalised with a nonsmooth maximum operation.

It is also possible to combine a feature with a classical density design variable field $\mathbf{x}$, which will be used in Section 5.3.3:

$$\hat{x}_e = \left({}^f\bar{x}_e^{P_1} + x_e^{P_1} \right)^{\frac{1}{P_1}}. \tag{5.10}$$

The sensitivities from Eqs. (5.9) and (5.10) and follow naturally, and with the chain rule the sensitivities with respect to the design variables can be obtained.

5.2.3. REGULATION OF FEATURE SHAPES

The method described in Section 5.2.1 and 5.2.2 succeeds in creating features, whose shape can be optimized. However, it was found that undesired feature shapes can emerge during the optimization process. In this section, we will describe these, and some techniques which we use to regulate feature shapes.

Four problems will be addressed: self intersecting boundaries, uneven PP distribution, proximity of solid and void PPs, and initial design influence. Note that,

the proposed restrictions not always address the fundamental cause of a problem, and instead restrict it indirectly because of simplicity or associated computational cost. Also, extra constraints add extra complexity to the optimization problem limiting design freedom. Possibly future improvements on the method may make these regulations redundant.

The first problem that was tackled, is self intersecting feature curves. Self intersecting feature shapes do not necessarily have the intermediate density region on the outside of the feature, while this is crucial for consistent sensitivities. In our application, self intersecting shapes are prevented by adding a constraint imposed on the minimum distance between a CP and all line segments connecting consecutive CPs. An example is shown in Figure 5.9. The signed distance d from point $\mathbf{a}_2$ to the line segment between $\mathbf{a}_1$ and $\mathbf{a}_3$ is calculated similarly to e.g. Smith and Norato [65] and Norato *et al.* [7], and is given by:

$$
\begin{aligned}
d(\mathbf{a}_1,\mathbf{a}_2,\mathbf{a}_3) := &\begin{cases} \|\mathbf{b}_1\| & \text{if} \quad \mathbf{h}_1\cdot\mathbf{h}_2 \le 0, \\ \|\mathbf{b}_2\| & \text{if} \quad 0 < \mathbf{h}_1\cdot\mathbf{h}_2 < \mathbf{h}_1\cdot\mathbf{h}_1, \\ \|\mathbf{b}_3\| & \text{if} \quad \mathbf{h}_1\cdot\mathbf{h}_2 \ge \mathbf{h}_1\cdot\mathbf{h}_1, \end{cases} \\
&\text{with} \quad \mathbf{h}_1 := \mathbf{a}_2 - \mathbf{a}_1 \quad \text{and} \quad \mathbf{h}_2 := \mathbf{a}_3 - \mathbf{a}_1.
\end{aligned}
\tag{5.11}
$$

Next, each distance d_i between a CP and all line segments connecting the remaining CPs is constrained to stay above a threshold value, $d_{\min}$. First, the distance d_i is normalized and transformed to $\tilde{d}_i = 1 - d_i/d_{\min}$, such that $\tilde{d}_i > 0$ indicates constraint violation, and $\tilde{d}_i < 0$ indicates a feasible design. Next, a smooth rectifier function is used, which ensures that $\tilde{d}_i > 0$ values are projected to $\hat{d}_i > \epsilon$, while $\tilde{d}_i < 0$ approach to zero, i.e. $\hat{d}_i \to 0$, including a smooth transition required for the sensitivity analysis. This function is shown in Figure 5.10, and it allows for aggregation of all distance constraints through summation. Finally, the inequality constraint for the optimization problem g_d is computed as:

$$
\begin{aligned}
\tilde{d}_i &= 1 - \frac{d_i}{d_{\min}} \\
\hat{d}_i &= \frac{\ln\left(\mathrm{e}^{\left(\beta_2 \frac{\beta_1 + \tilde{d}_i}{\beta_1}\right)} + 1\right)}{\beta_2} \\
g_d &= \sum_i^{n_d} \hat{d}_i - \epsilon \\
g_d &\le 0
\end{aligned}
\tag{5.12}
$$

Here, we use $\beta_1 = 0.2$ and $\beta_2 = 3$, n_d is the number of line segments considered, and $\epsilon = \ln\left(\mathrm{e}^{\beta_2} + 1\right)/\beta_2 \approx 1.016$ is used.

In our experience, this method was found to be simple, fast, and sufficiently effective. However, one downside is that the distance constraint can prevent small features. Also, this method does not prevent self intersection of the actual curve, since only the zeroth order NURBS curve is constrained, for simplicity. Finally, self intersection prevention of a zeroth order curve during optimization is also not ensured when the optimization movelimits allow big CP displacements that might lead to CPs jumping over a line segment without violating the constraint.

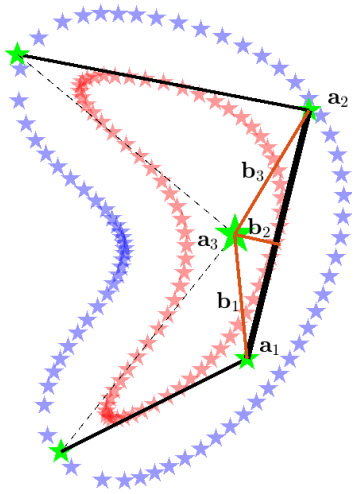


Figure 5.9: Self intersecting solid PPs are undesirable. This problem is mitigated by constraining a minimum distance between a CP and line segments between other CPs. For the marked CP ($\mathbf{a}_3$), the distance is measured towards all black line segments, with the line segment between $\mathbf{a}_1$ and $\mathbf{a}_2$ highlighted.

The second problem that has been tackled, is an uneven PP distribution in the void PP offset curve, such as shown in Figure 5.11. This occurs especially at sharp corners due to large CP weights. Increasing the number of PPs would mitigate this problem, but with additional computational costs.

Sharp corners can be prevented by limiting the range of $\mathbf{p}^{(w)}$ values. Usually NURBS weights are allowed within the range $[0,1]$. In our implementation a smaller range of $[0.25,0.75]$ is imposed. Other ranges also work for most problems, however in our experience this selected range ensured good performance in all test cases. Secondly, the distribution of the void PPs is further smoothened through modifying the calculation of the offset curve with $\mathbf{n}_j$. Instead of using the local normal of the NURBS curve, an approximate normal determined by finite difference using a relatively large step size leads to a more uniform spacing of void PPs. Therefore, in this study, we use a finite difference operation involving PPs $j+2$ and $j-2$:

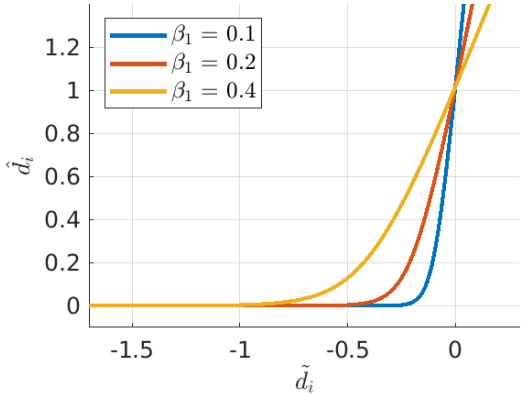


Figure 5.10: Input is (scaled) distance, output is a constraint value. By adding a maximum constraint value, a minimum distance is ensured strictly.

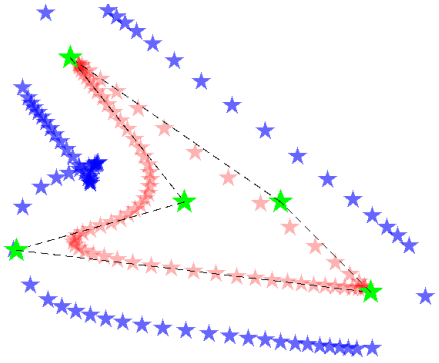


Figure 5.11: An uneven distribution of void PPs at inner or outer corners is undesirable, for the projection to the density field. This problem is mitigated by restricting the weights of the CPs, and by smoothening of the offset curve calculation.

$$\begin{aligned} &\mathbf{n}_j = \frac{\mathbf{h}_1}{\|\mathbf{h}_1\|}, \\ &\text{with} \quad \mathbf{h}_1 := \mathbf{h}_2 \times \mathbf{h}_3, \\ &\text{where} \quad \mathbf{h}_2 := \dot{\mathbf{c}}_{j+2} - \dot{\mathbf{c}}_{j-2} \quad \text{and} \quad \mathbf{h}_3 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{5.13}$$

This is valid for an anti-clockwise PP ordering, and since the curve is continuous the first PP of the curve uses the next to last PP of the curve for the $j-2$, and vice versa.

The third problem is void PPs getting too close to or inside of the solid PP curve, caused by sharp inner corners. An example of such CP locations is shown in Figure 5.12. Therefore, another constraint is added, limiting the angle between two adjacent line segments between CPs. These angles are calculated, and constrained with a similar projection and summation as used before:

$$\begin{aligned} \tilde{\theta}_i &= 1 - \frac{\theta_i}{\theta_{\min}} \\ \hat{\theta}_i &= \frac{\ln\left(e^{\left(\beta_2 \frac{\beta_1 + \tilde{\theta}_i}{\beta_1}\right)} + 1\right)}{\beta_2} \\ g_\theta &= \sum_i^{n_\theta} \hat{\theta}_i - \epsilon \\ g_\theta &\leq 0 \end{aligned} \tag{5.14}$$

Again, $\beta_1 = 0.2$ and $\beta_2 = 3$ are used, n_θ is the number of constrained inner corners, $\epsilon = \ln\left(e^{\beta_2} + 1\right)/\beta_2$, and the angle θ is scaled $\tilde{\theta}_i = 1 - \theta_i/\theta_{\min}$, such that the constraint of becomes active for angles smaller than $\theta_{\min}$. This gives an indirect control over the sharpness of the corners.

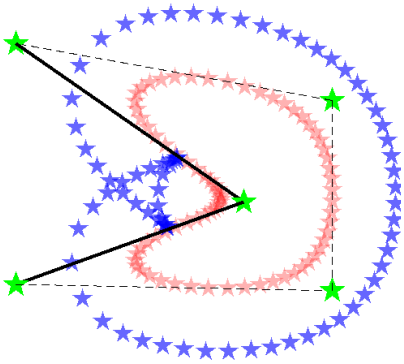


Figure 5.12: Sharp inner corners can cause an undesirable distribution of void PPs. This problem is mitigated by constraining the angle θ between connected CPs, with Eq. (5.14).

The fourth problem that was tackled concerns the influence of the initial design. It is common in existing LS and FFM methods for the initial design state to have a significant influence on the outcome. Consequently, convergence to different local optima is observed. In our method with fewer features especially, this is relevant. In order to decrease the influence of the initial design, the features are therefore gradually activated with a continuation scheme, similar to the approach presented

in van de Ven *et al.* [16]. In the first iterations of the optimization process, we mix the final feature density field $\hat{\mathbf{x}}$ with a classical density field $\mathbf{x}$:

$$\check{x}_e = (1-\gamma)x_e + \gamma\hat{x}_e, \tag{5.15}$$

where $0 \leq \gamma \leq 1$ is the mixing parameter. In this chapter γ is continuously increased from 0 to 1 in the first 20 optimization iterations.

5.3. NUMERICAL EXAMPLES

In this section, the numerical examples are presented. Section 5.3.1 presents the optimization formulation and the parameters used. Next, a mechanical optimization problem with solid features is described and analyzed in Section 5.3.2. Subsequently, Section 5.3.3 presents a thermal optimization problem with void features.

5.3.1. OPTIMIZATION PROBLEM FORMULATION

The first numerical example considers a mechanical optimization of solid features for minimum compliance as objective in the presence of a volume constraint:

$$\begin{aligned}
\underset{\mathbf{x},\mathbf{p}}{\text{minimize}}: \quad & C(\check{\mathbf{x}}) = \mathbf{u}^T\mathbf{K}_m(\check{\mathbf{x}})\mathbf{u} \\
\text{subject to}: \quad & \mathbf{K}_m(\check{\mathbf{x}})\mathbf{u} - \mathbf{f} = \mathbf{0} \\
& \frac{V(\check{\mathbf{x}})}{V^*} - 1 \leq 0 \\
& g_d \leq 0 \\
& g_\theta \leq 0
\end{aligned} \tag{5.16}$$

Here, $\mathbf{K}_m$, $\mathbf{u}$ and $\mathbf{f}$ denote the finite element system stiffness matrix, displacement vector and mechanical load vector. $\mathbf{p}$ is used to denote the design variables $\mathbf{p}^{(x)}$, $\mathbf{p}^{(y)}$, and $\mathbf{p}^{(w)}$. Note that $\check{\mathbf{x}} = \check{\mathbf{x}}(\mathbf{x},\mathbf{p})$. The objective is compliance C, the current design volume is V, and the maximum allowed volume is $V^* = 0.2$ of the design domain.

The second numerical example considers a thermal optimization problem with a single void feature, with minimum thermal compliance as objective and a volume constraint. In this problem, the structure is formed by a density design variable field, combined with a void feature that has a minimum volume. This is done with a minimum operation for Eq. (5.10), with $P_1 = -6$. One application for this problem would for example be the placement and shape determination of a tank with a specified minimum volume in a design area. The second optimization problem is

given as follows:

$$\begin{aligned} \underset{\mathbf{x},\mathbf{p}}{\text{minimize}}: \quad & C(\check{\mathbf{x}}) = \mathbf{T}^T \mathbf{K}_t(\check{\mathbf{x}})\mathbf{T} \\ \text{subject to}: \quad & \mathbf{K}_t(\check{\mathbf{x}})\mathbf{T} - \mathbf{q} = \mathbf{0} \\ & \frac{V(\check{\mathbf{x}})}{V^*} - 1 \leq 0 \\ & g_d \leq 0 \\ & g_\theta \leq 0 \\ & 1 - \frac{V_f}{V_f^*} \leq 0 \end{aligned} \tag{5.17}$$

Now, $\mathbf{K}_t$, $\mathbf{T}$ and $\mathbf{q}$ denote the finite element system conductivity matrix, temperature vector and thermal load vector. The minimum volume constraint of the void feature is added with its current volume denoted by V_f and its minimum volume by V_f^*.

5

For each element e the Young's modulus required for the calculation of $\mathbf{K}_m$ in Eq. (5.16) and the conductivity required for the calculation of $\mathbf{K}_t$ in Eq. (5.17), the modified SIMP interpolation scheme proposed by Sigmund [5] is used, i.e.:

$$E(\check{x}_e) = E_{\min} + \check{x}_e^p (E_{\max} - E_{\min}), \tag{5.18}$$

with $p = 3.0$, and Young's moduli $E_{\min} = 10^{-9}$ and $E_{\max} = 1$ for the mechanical problem. The same interpolation scheme is used for the thermal problem, with conductivities $E_{\min} = 10^{-3}$ and $E_{\max} = 1$. For the finite element analysis, a structured mesh comprising 4-node quadrilateral elements with bilinear shape functions is employed. These aspects of the problem are kept relatively simple and standard, as our focus is on evaluating the strengths and weaknesses of the novel formulation.

The problem is implemented as an extension to the 88 line MATLAB code by Andreassen *et al.* [25], supplemented with the MMA optimizer by Svanberg [26] with an additional set of CP design variables. The optimization is terminated after 250 iterations, by which a desired level of convergence was always reached.

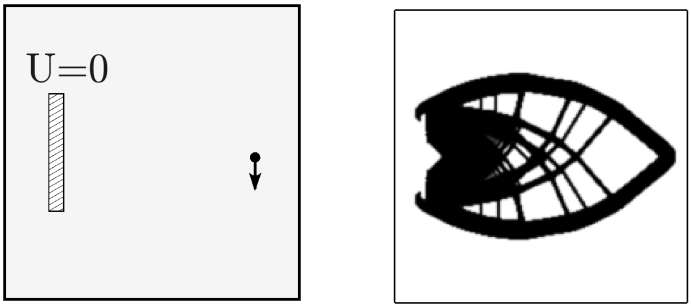
Solid features with 8 CPs were used for the mechanical problem, and one void feature with 8, 12, or 16 CPs was used in the thermal problem. The number of features and CPs affects the resulting designs. Therefore, in the industrial setting, these parameters can be determined after several numerical experiments to adjust the geometric complexity of the design with the manufacturing means. Second order basis functions were used for the NURBS. The knot vector is unclamped uniform. The NURBS are discretized with 30 solid PPs per CP. The projection radius R is 4.5 times element length l_x, which is also equal to the separation between solid and void PPs, r. The MMA limits for the CP coordinates $\mathbf{p}^{(x)}$ and $\mathbf{p}^{(y)}$ are $[0,1]$, for a unit square design domain, to keep the CPs inside. The MMA limits for the CP weights $\mathbf{p}^{(w)}$ are $[0.25, 0.75]$ as explained in Section 5.2.3. The minimum distance and angle constraints are set at $d_{\min} = 4.5 l_x$ and $\theta_{\min} = 45°$, respectively. An overview of the used parameters is given in Table 5.1.

Table 5.1: Summary of used parameter values

Parameter	Value
Projection radius R	$4.5l_x$
Offset curve distance r	$4.5l_x$
P_1	6 / -6
SIMP exponent p	3.0
E_{min}	10^{-3} / 10^{-9}
E_{max}	1
Poisson's ratio ν	0.3
Number of iterations	250
Number of solid PPs per CP $n^{(CP)}$	30
Minimum distance d_{min}	$4.5l_x$
Minimum angle θ_{min}	45°

5.3.2. MECHANICAL PROBLEM WITH SOLID FEATURES

The first numerical example is a cantilever beam case. Since this problem has predictable behaviour, the performance of the proposed method can be easily observed. The loading, boundary conditions and a conventional density-based TO result are shown in Figure 5.13. A discretization of 200×200 elements is used.



(a) Problem loading.

(b) Reference design. $C := C_{ref}$

Figure 5.13: The cantilever beam compliance problem. In (a) the load and boundary conditions are shown. The unit load is applied 5% from the right edge, and 50% from the bottom. The fully clamped region is located 10% from the left edge, 50% from the bottom, and has a width of 2% and a height of 30%. In (b) an optimization result obtained using the standard density method is shown.

For this load case given in Figure 5.13a, the novel method is tested with a different

number of features. Initially, as many as 16 features are introduced, similar to what is typically considered in existing FMMs (Wein *et al.* [53]). Next, the number of features is reduced to 4 and 2, to demonstrate the full potential of the highly flexible feature shapes. The initial feature designs are shown in Figure 5.14a, 5.14f, and 5.14k. Furthermore optimization with 2 features with another initialization is also performed (Figure 5.15a). Finally, the 4 feature optimization is performed without the distance and angle constraints, and without tight movelimits for the CP weights to eliminate sharp corners. However, the mixing of the designs in Eq. (5.15) is still included to diminish the influence of the initialization.

Several intermediate designs and the end results of the optimizations are shown in Figure 5.14. For the first three tests with 16, 4, and 2 solid features, a relatively simple cantilever beam is created. The case with 16 features is very similar to classic FFM, where many features are successfully combined, however the shapes of the individual features are fairly simple. The cases with 4 and 2 features create similar designs with fewer features, utilizing the flexibility in the feature shape of the proposed method. This simplicity comes with only a minor increase in relative compliance, with values of 1.030, 1.060, and 1.084, for the 16, 4, and 2 features respectively.

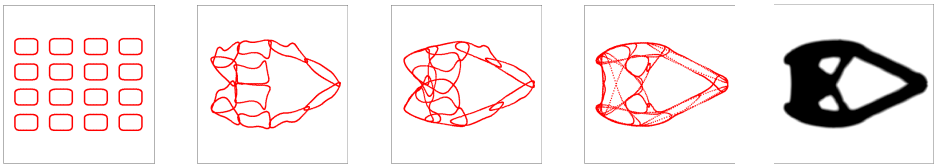
The test with an alternative initial design (Figure 5.15a-e) shows a similar part shape but the hole in the middle of the structure is absent, and the relative compliance increases considerably to 1.282. The last test shows that, for this load case, a design of almost identical performance can be successfully obtained even without the regularization schemes introduced in Section 5.2.3. However, the proposed regularizations were generally found to further ensure a stable optimization process and mitigate the risk of undesired designs.

Next, the convergence performance of the method is considered. Two convergence plots are displayed in Figure 5.16 and 5.17. Irrespective of the number of features considered, swift convergence for the objective is observed, in which the features quickly connect, and smoothly evolve to an optimized shape. Some peaks can be observed until the 20th iteration, during which the features' density fields are still combined with a classical density field, leading to a decrease in the objective. However, as the fraction of the classical density field is continuously decreased, the objective can increase. In these three tests, the features are connected around iteration 20, but in our experience features could often still connect at a later stage if not yet connected at iteration 20.

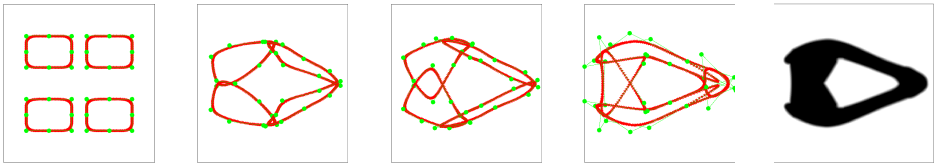
The constraints also show good convergence behaviour. The results shown in Figure 5.17 for the case with 4 features are representative for other cases. The distance constraint is active or close to be active, which performs as intended. The inner corner angle constraint is not active at any iteration, and would thus not be needed in this case.

5.3.3. HEAT CONDUCTION PROBLEM WITH VOID FEATURE

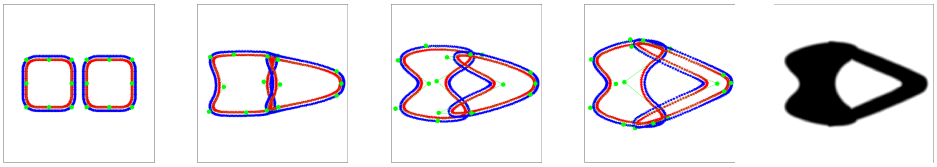
The second numerical example is the heat sink problem, for which the loading condition and a typical conventional density-based TO result are shown in



(a) 16 features, iteration 1 (b) 16 features, iteration 20 (c) 16 features, iteration 40 (d) 16 features, iteration 250 (e) 16 features, iteration 250, $C := 1.030C_{\text{ref}}$



(f) 4 features, iteration 1 (g) 4 features, iteration 20 (h) 4 features, iteration 40 (i) 4 features, iteration 250 (j) 4 features, iteration 250, $C := 1.060C_{\text{ref}}$

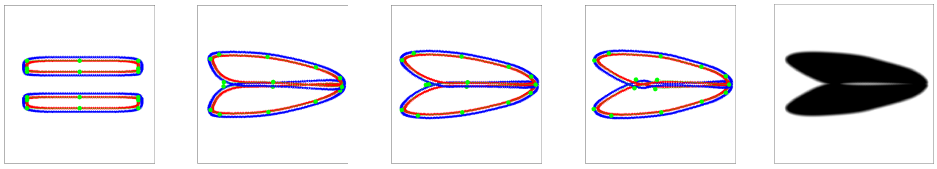


(k) 2 features, iteration 1 (l) 2 features, iteration 20 (m) 2 features, iteration 40 (n) 2 features, iteration 250 (o) 2 features, iteration 250, $C := 1.084C_{\text{ref}}$

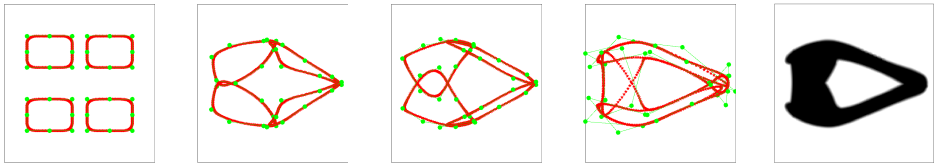
Figure 5.14: Results of the cantilever beam problem, with 16, 4, and 2 features given in the first, second, and third row, respectively. For clarity, only features with ${}^{f}\alpha > 0.5$ are shown, void PPs have been omitted from cases with more than two features, and CPs have been omitted from the case with more than 4 features.

Figure 5.18. Again, a discretization of 200×200 is used.

For this case we aim to add a void feature with a specified minimum area, which could for example represent a situation of how and where to locate a (fluid) tank with a flexible shape inside a domain. This is similar to a flexible void area considered in e.g. Clausen *et al.* [66], however with a fixed topology. This design challenge is here combined with the thermal compliance problem, which tends to form a branching network of conductive material throughout the domain (see Figure 5.18b). In this way, the void fluid tank represent a conflict for the thermal compliance objective. The void feature field has a value of 1 outside of the feature, and 0 inside of the feature shape. This field is added with a smooth minimum operation to a classical density field. Its opacity parameter is fixed at $\alpha = 1$, since the only feature should



(a) 2 features, iteration 1 (b) 2 features, iteration 20 (c) 2 features, iteration 40 (d) 2 features, iteration 250 (e) 2 features, iteration 250, $C := 1.282C_{\text{ref}}$



(f) 4 features, no regulations, iteration 1 (g) 4 features, no regulations, iteration 20 (h) 4 features, no regulations, iteration 40 (i) 4 features, no regulations, iteration 250 (j) 4 features, no regulations, iteration 250, $C := 1.061C_{\text{ref}}$

Figure 5.15: Results of the cantilever beam problem. The first row is with 2 features with a different initialization, while the second row is with 4 features and no regulations. For clarity, only features with ${}^{f}\alpha > 0.5$ are shown, void PPs have been omitted from cases with more than two features, and CPs have been omitted from the case with more than 4 features.

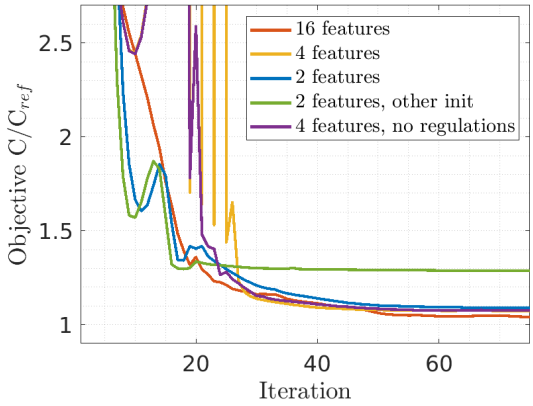


Figure 5.16: Convergence behaviors for the objective, for all mechanical problems.

not disappear.

First, to study the influence of the number of CPs, the feature is created with 8, 12, or 16 CPs, and the (asymmetric) initial designs can be seen in Figure 5.19a, 5.19f,

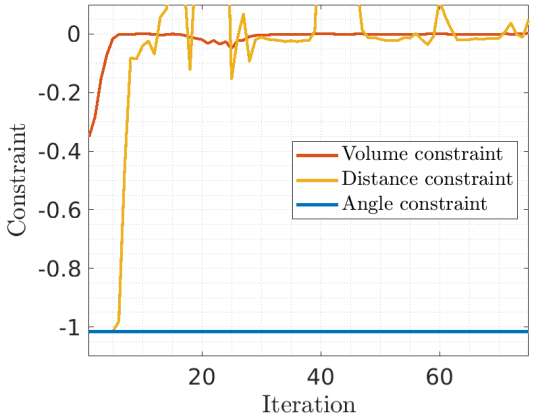
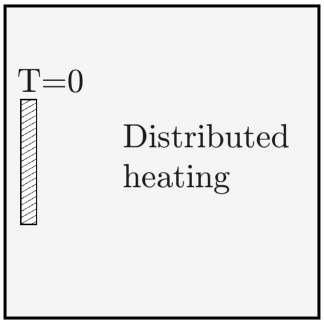
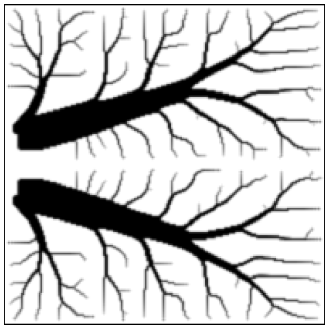


Figure 5.17: Convergence behaviour for the three constraints, for the test with 4 features (second row in Figure 5.14).



(a) Problem loading.



(b) Reference design. $C := C_{\text{ref}}$

Figure 5.18: The heat conduction problem. In (a) the load and boundary conditions are shown. The heat sink region is located 2.5% from the left edge, 50% from the bottom, and has a width of 5% and a height of 20%. In (b) the optimization result without any features is shown as reference.

and 5.19k. Secondly, a 12 CP feature is considered again, but the angle constraint is made more strict to enforce wider inner angles of 70° specifically.

The results of the optimizations are shown in Figure 5.19. The novel method allows for the void region have a flexible shape, yet it is ensured to keep its topology. The void features with 8, 12, or 16 CPs lead to an relative compliance of 1.429, 1.230, and 1.247 respectively. Increasing the number of CPs increases the shape freedom, however as seen in e.g. Figure 5.19n, the void PPs almost penetrate the solid PP

curve. This implies the constraints are successful in keeping realistic feature shapes. The feature with 16 CPs can still create sharp inner corners by putting two CPs close next to each other. The minimum distance constraint however still ensures that the void PPs do not touch the solid PP curve.

The fourth case, with a more strict angle constraint, for which the results can be seen in Figure 5.19p-t, shows that the constraints succeed in controlling the feature shape and a more rounded void region is obtained albeit with a relative compliance of 1.415.

A constraint convergence plot for the thermal problem problem is given in Figure 5.20. As can be seen, all constraints except the distance constraint are active or close to be active. For this problem all constraints are needed in keeping a desired feature shape.

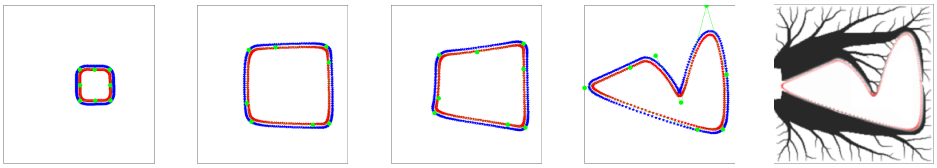
5.4. DISCUSSION

As shown in Section 5.3, the proposed method succeeds in optimizing feature shapes with high geometric flexibility. This new method offers a combination of flexibility and control, which is different from more restricted explicit methods yet more flexible than existing implicit feature-based methods. Next to these advantages, the method however still has some shortcomings. The nature of the method has two major downsides: i) the high feature flexibility necessitates applying restrictions on flexible shaped features in general ii) our approach for obtaining sensitivities, using a void PP offset curve.

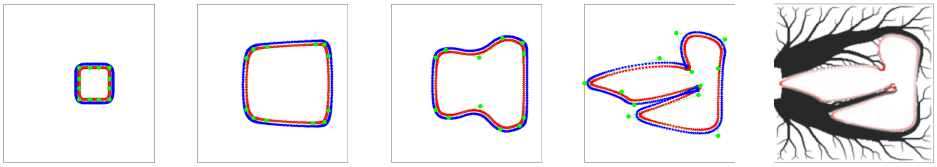
First, the feature shape freedom of the novel FFM method can be misused to create unrealistic and undesired flexible structures, e.g. self intersection. The restrictions proposed in Section 5.2.3 aim to regulate the shape freedom, in order to only create realistic features, even though the regulations also limit the design freedom and may prohibit certain admissible shapes. The problem of how much shape restriction is required, is inherent to shaped features, and becomes particularly evident with the relatively high shape flexibility introduced by our method. While the methods we have developed proved effective, it is quite possible that less restrictive and less conservative regularization formulations are possible. Furthermore, the concept of flexible features regulated with various restrictions/constraints also offers new opportunities: for specific ranges of shape deformations specific geometric rules could be imposed, allowing e.g. combining parametric design with TO. Both aspects are identified as directions for future research.

Secondly, the void PP offset curve, an essential part for the grey boundary which is crucial for obtaining sensitivities, introduces challenges. For example, in the presented formulation the void PP curve can suffer from uneven PP distributions with big gaps or clustering, or void PPs can penetrate the solid PP curve. One advantage advocating for the offset curve is consistent sensitivities in void regions surrounded by solid regions, such that this location is equally close to multiple boundaries/edges.

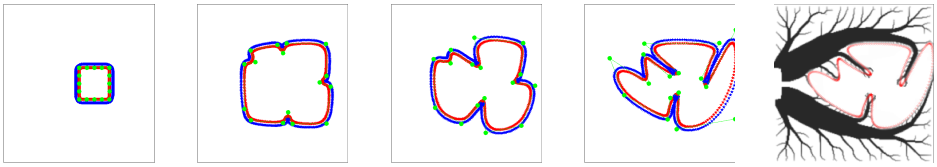
Next, the feasibility of extending the method to 3D should be addressed. For all



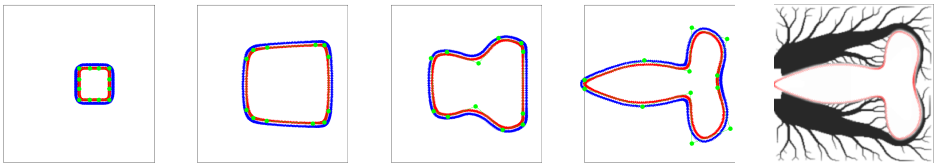
(a) 8 CPs, iteration 1 (b) 8 CPs, iteration 20 (c) 8 CPs, iteration 40 (d) 8 CPs, iteration 250 (e) 8 CPs, iteration 250, $C := 1.429C_{\text{ref}}$



(f) 12 CPs, iteration 1 (g) 12 CPs, iteration 20 (h) 12 CPs, iteration 40 (i) 12 CPs, iteration 250 (j) 12 CPs, iteration 250, $C := 1.230C_{\text{ref}}$



(k) 16 CPs, iteration 1 (l) 16 CPs, iteration 20 (m) 16 CPs, iteration 40 (n) 16 CPs, iteration 250 (o) 16 CPs, iteration 250, $C := 1.247C_{\text{ref}}$



(p) 12 CPs, 70° constraint, iteration 1 (q) 12 CPs, 70° constraint, iteration 20 (r) 12 CPs, 70° constraint, iteration 40 (s) 12 CPs, 70° constraint, iteration 250 (t) 12 CPs, 70° constraint, iteration 250, $C := 1.415C_{\text{ref}}$

Figure 5.19: Results of the thermal problem. The first row is one NURBS feature with 8 CPs, the second row with 12 CPs, the third row with 16 CPs, and the fourth row with 12 CPs and an angle constraint of 70°.

operations in Section 5.2.1 and 5.2.2 the extensions to a third dimension follow straightforward. However, the feature shape problems addressed in Section 5.2.3 could potentially be more challenging. Naturally, 3D features are created with more CPs, which increases the design freedom, with potential for unrealistic shapes. Furthermore, the presented regulation methods do not directly transfer to 3D. The

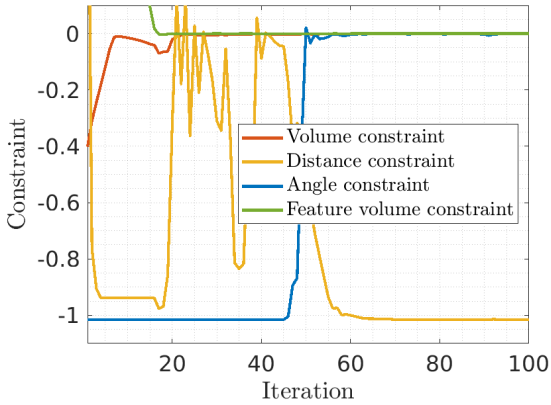


Figure 5.20: Convergence behaviour for the three constraints, for the thermal test with 12 CPs (second row in Figure 5.19).

minimum distance calculation from Eq. (5.11) should become a distance between a CP and a plane segment. Similarly, the angle used in Eq. (5.14) should potentially be between several neighbouring CPs, or a different way to smoothen the normals should be applied. While these are definitely challenges that require further investigation, the concepts introduced by the proposed method fundamentally carry over to 3D.

With the expansion to 3D, further research can be done on controlling feature sizes. This is a useful property of explicit FFMs, and integrating it with the proposed method could further benefit practical applications.

In our experience using an unoptimized implementation, the computational cost of all feature operations (creation and sensitivities) was less than 10% of the FE solve, in the studied problems. Feature operation costs scale up linearly with the number of features, and the number of PPs, so the fraction of the feature operations on the total cost will be even lower for problems involving finer meshes. It is important to note that our approach requires a limited number of control points to describe complex feature shapes which is favorable for computational tractability.

Ultimately, FFMs are methods in which feature boundaries move. Just like in LS methods, but with shape restrictions to ensure simpler designs. Consequently, design outcomes are limited to a subset of all possible geometries, which likely also implies reduced performance. However, the degree of restriction is controlled by the designer. Our method provides a new option to control the range from full design freedom and potentially complex outcomes, to limited design freedom and simple outcomes. For each specific combination of application demands, manufacturing and cost considerations it allows for a desired amount of restrictions and regulations.

5.5. CONCLUSION

A new method for feature optimization with flexible feature shapes is proposed, through defining feature shapes with NURBS. With the NURBS control points and their weights as design variables a feature density field is constructed. The method builds on conventional density-based TO procedures. All steps are differentiable, allowing for consistent sensitivities to be calculated.

Features can be created with complex shapes, allowing for layouts to be represented with a fixed topology, by fewer features, and with varying degrees of geometric control. The features can be either solid or void, including an opacity parameter, and used seamlessly in combination with established density methods. As shown by different numerical examples, the method succeeds in optimizing feature shapes with good convergence characteristics, and follows for solving new problems with a degree of shape control not available in other approaches.

In several aspects, the proposed method offers potential for further development. Firstly, extending its implementation to 3D is worth exploring. While the concept fundamentally remains the same as in the planar case, various implementation details may need additional consideration. Secondly, to enable specific control of feature shapes, the set of regulation schemes proposed and demonstrated in this chapter could be further refined and extended. Along this path the method offers a way to combine parametric design concepts and free-form TO, and provides designers with new means to control the computational design process.

6

CONCLUSIONS AND RECOMMENDATIONS

6.1. CONCLUSIONS

First, it can be concluded that incorporating various cleanability requirements into topology optimization (TO) has been made possible, which opens up a range of possibilities for designers. Specifically, in Chapter 2, in a layerwise process, a given blueprint design is transformed into a drainable design. With this, drainability can now be ensured within TO. Next, with regard to cleaning by fluid jets in Chapter 3 and Chapter 4 access fields are computed for each jet, which are then combined into a total access field from which a jettable design field is obtained. Integrating this into TO allows for generating designs that can be cleaned through jetting from predefined and even optimized positions. In addition, a wide range of geometric requirements can be ensured with the general framework of Chapter 4. The presented methods are easy to combine with existing density based TO. Together, the developed methods address main cleanability requirements.

Next, it can be concluded that seemingly different geometric requirements can often be handled in a similar manner. For new geometric requirements, inspiration can be taken from existing methods, and in special cases directly borrowed. This was seen in Chapter 2, where the draining filter bears strong conceptual similarities with a 3D printing filter, and in Chapter 3 where the jetting filter was inspired by both a milling and a 3D printing filter. Conversely, in Chapter 4, a philosophy was introduced that allows a novel common framework to be used for various different geometric requirements. To be efficient with obtained knowledge and prevent a *va-et-vient* of highly specific application papers, it is an important insight that geometric filtering methods can often serve in several seemingly unrelated applications.

Furthermore, it can be concluded that the sets of points decoupled from the design domain discretization, as introduced in this thesis, become a versatile tool for many TO applications. This has been demonstrated in Chapters 4 and 5. There are several benefits of using these decoupled points. First, they allow for a simple and versatile description of shapes. Secondly, the local interpolation/projection is beneficial because i) only local computations are required, reducing computational cost ii) detachment from the mesh allows for unstructured meshes, iii) consistent sensitivities for the point location can be obtained. Finally, the point location sensitivities are instrumental for extra design variables that increases design freedom, through extra flexibility in either the shape or orientation of the described shape. While two ways to exploit these benefits have been demonstrated in this thesis, it is expected that many more use cases can be conceived.

A final conclusion drawn here is about a concept that is coined here as the TO version of repressive tolerance (Marcuse [67]). It is clear that all settings of a TO problem can be optimized, and added as design variables to the original problem. However, the overload of options to include in the optimization problem, can take away the attention from the most important design considerations, and consequently negatively influence the design outcome. For example, most industrial components would undergo only minor changes if the draining filter is added to the optimization

process. And optimization of the printing orientation is prone to be stuck in a local optimum close to the initial design. Therefore, given the current optimizer capabilities, the engineer should always make a trade-off how much gain could be achieved from each additional optimization variable versus how much nonlinearity and complexity is added to the optimization problem. Experience from engineers can help to determine which additional optimization variables should be used.

6.2. RECOMMENDATIONS

While clear steps have been made in integrating cleanability requirements in TO, several topics for further research remain. First, it is recommended to combine draining and jetting filters with the method presented in Chapter 4. The framework there is more general and versatile, and this will also make it easier to integrate the different filters.

A second recommendation for future research concerns measures to improve convergence. Unstable convergence behaviour can occur during TO, which can result in a design with an underperforming local optimum. Important reasons for such behavior include, in the experience of the author, i) geometric requirements being implemented/enforced with filters, which can be excessively strict, and ii) small regions can have significant influence on other large portions of the design domain, resulting in high sensitivity values and drastic design updates.

In the presented work, the geometric requirements are always enforced through a filter operation. An advantage of this is that the filtered design is guaranteed to fulfill design requirements. However, for complex problems with multiple geometric requirements, the rather strict filter implementation increases the complexity of the feasible domain of an already non-convex optimization problem, and may quickly steer the design towards unfavorable local optima. Therefore, it is recommended to consider other less strict methods for enforcing the geometric requirement. For example constraints that can be temporarily violated, for instance with constraint tolerances, or a more sophisticated mixing of unfiltered and filtered design.

Regarding the second cause of unstable convergence, in geometric filters it is often encountered that small regions of design variables have impact on much larger regions. For example, in Chapter 2, the sink holes influence the entire region above, in Chapter 3 the edge of the structure is more influential than its interior, and in Chapter 4 small changes close to the baseplate significantly influence the design above. Several specific heuristic methods have been presented and applied to mitigate the disproportional influence of small regions. However, it would be preferred if a more elegant and general solution could be found for this.

To take this idea even further, a third recommendation is to perform further research for combinations of geometric requirements. In industrial settings, usually multiple geometric requirements are in place. Applying several filter operations sequentially does not necessarily ensure each individual requirement in the final design. Furthermore, some filters can have an unreasonable amount of influence on the optimization

process. This calls for a method to elegantly combine different geometric design requirements, without having to develop new versions for every possible combination.

Finally, further research is recommended regarding aspects of design for cleanability that could not be addressed by this thesis. The numerical methods presented in this work represent an important first step, but can be further improved and expanded. For instance, several simplifying assumptions were made in the developed methods. Future research can for example look into more accurately representing the cleaning effect of a fluid jet, including a minimum angle of attack, the trajectory of spray after the first surface contact, include distance requirements, and a computationally cheap way to model many possible jet positions. Another research topic that is left for future work includes e.g. access for cleaning tools. Potentially, the flexible void features from Chapter 5 could be used to model the shape of cleaning tools, or even to design the part and corresponding dedicated cleaning tools in an integrated TO process. Lastly, physical tests are required for the presented numerical methods and the designs generated by them, as this will undoubtedly lead to new valuable insights.

BIBLIOGRAPHY

[1] M. P. Bendsøe and O. Sigmund. *Topology Optimization*. Springer Berlin, Heidelberg, 2002. ISBN: 978-3-540-42992-0. DOI: 10.1007/978-3-662-05086-6.

[2] C. Wang, Z. Zhao, M. Zhou, O. Sigmund, and X. Shelly. "A comprehensive review of educational articles on structural and multidisciplinary optimization". In: *Structural and Multidisciplinary Optimization* 64 (2021), pp. 2827–2880. DOI: 10.1007/s00158-021-03050-7.

[3] M. Bayat, O. Zinovieva, F. Ferrari, C. Ayas, M. Langelaar, J. Spangenberg, R. Salajeghe, K. Poulios, S. Mohanty, O. Sigmund, and J. Hattel. "Holistic computational design within additive manufacturing through topology optimization combined with multiphysics multi-scale materials and process modelling". In: *Progress in Materials Science* 138 (2023), p. 101129. DOI: 10.1016/j.pmatsci.2023.101129.

[4] F. Wang, B. S. Lazarov, and O. Sigmund. "On projection methods, convergence and robust formulations in topology optimization". In: *Structural and Multidisciplinary Optimization* 43 (2011), pp. 767–784. DOI: 10.1007/s00158-010-0602-y.

[5] O. Sigmund. "Morphology-based black and white filters for topology optimization". In: *Structural and Multidisciplinary Optimization* 33 (2007), pp. 401–424. DOI: 10.1007/s00158-006-0087-x.

[6] R. Bartz, T. Franke, S. Fiebig, and T. Vietor. "Density-based shape optimization of 3D structures with mean curvature constraints". In: *Structural and Multidisciplinary Optimization* 65.5 (2022). DOI: 10.1007/s00158-021-03089-6.

[7] J. A. Norato, B. K. Bell, and D. A. Tortorelli. "A geometry projection method for continuum-based topology optimization with discrete elements". In: *Computer Methods in Applied Mechanics and Engineering* 293 (2015), pp. 306–327. DOI: 10.1016/j.cma.2015.05.005.

[8] R. Giele, F. van Keulen, and M. Langelaar. "Design for drainability in density-based topology optimization". In: *Structural and Multidisciplinary Optimization* 65.183 (2022). DOI: 10.1007/s00158-022-03272-3.

[9] J. Alexandersen and C. S. Andreasen. "A review of topology optimisation for fluid-based problems". In: *Fluids* 5.1 (2020), pp. 1–32. DOI: 10.3390/fluids5010029.

[10] A. T. Gaynor and T. E. Johnson. "Eliminating occluded voids in additive manufacturing design via a projection-based topology optimization scheme". In: *Additive Manufacturing* 33 (2020), p. 101149. DOI: 10.1016/j.addma.2020.101149.

[11] J. Liu, A. T. Gaynor, S. Chen, Z. Kang, K. Suresh, A. Takezawa, L. Li, J. Kato, J. Tang, C. C. Wang, L. Cheng, X. Liang, and A. C. To. "Current and future trends in topology optimization for additive manufacturing". In: *Structural and Multidisciplinary Optimization* 57 (2018), pp. 2457–2483. DOI: 10.1007/s00158-018-1994-3.

[12] X. Qian. "Undercut and overhang angle control in topology optimization: A density gradient based integral approach". In: *International Journal for Numerical Methods in Engineering* 111.3 (2017), pp. 247–272. DOI: 10.1002/nme.5461.

[13] G. Allaire, C. Dapogny, R. Estevez, A. Faure, and G. Michailidis. "Structural optimization under overhang constraints imposed by additive manufacturing technologies". In: *Journal of Computational Physics* 351 (2017), pp. 295–328. DOI: 10.1016/j.jcp.2017.09.041.

[14] A. T. Gaynor and J. K. Guest. "Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design". In: *Structural and Multidisciplinary Optimization* 54 (2016), pp. 1157–1172. DOI: 10.1007/s00158-016-1551-x.

[15] M. Langelaar. "Topology optimization of 3D self-supporting structures for additive manufacturing". In: *Additive Manufacturing* 12 (2016), pp. 60–70. DOI: 10.1016/j.addma.2016.06.010.

[16] E. van de Ven, R. Maas, C. Ayas, M. Langelaar, and F. van Keulen. "Continuous front propagation-based overhang control for topology optimization with additive manufacturing". In: *Structural and Multidisciplinary Optimization* 57 (2018), pp. 2075–2091. DOI: 10.1007/s00158-017-1880-4.

[17] M. Langelaar. "An additive manufacturing filter for topology optimization of print-ready designs". In: *Structural and Multidisciplinary Optimization* 55 (2017), pp. 871–883. DOI: 10.1007/s00158-016-1522-2.

[18] M. Hoffarth, N. Gerzen, and C. Pedersen. "ALM Overhang Constraint in Topology Optimization for Industrial Applications". In: *12th World Congress on Structural and Multidisciplinary Optimisation* (2017).

[19] M. Langelaar. "Combined optimization of part topology, support structure layout and build orientation for additive manufacturing". In: *Structural and Multidisciplinary Optimization* 57 (2018), pp. 1985–2004. DOI: 10.1007/s00158-017-1877-z.

[20] O. Sigmund. "Manufacturing tolerant topology optimization". In: *Acta Mechanica Sinica/Lixue Xuebao* 25 (2009), pp. 227–239. DOI: 10.1007/s10409-009-0240-z.

[21] B. S. Lazarov, F. Wang, and O. Sigmund. "Length scale and manufacturability in density-based topology optimization". In: *Archive of Applied Mechanics* 86 (2016), pp. 189–218. DOI: 10.1007/s00419-015-1106-4.

[22] B. Bourdin. "Filters in topology optimization". In: *International Journal for Numerical Methods in Engineering* 50.9 (2001), pp. 2143–2158. DOI: 10.1002/nme.116.

[23] T. E. Bruns and D. A. Tortorelli. "Topology optimization of non-linear elastic structures and compliant mechanisms". In: *Computer Methods in Applied Mechanics and Engineering* 190.26-27 (2001), pp. 3443–3459. DOI: 10.1016/S0045-7825(00)00278-4.

[24] D. Trillet, P. Duysinx, and E. Fernández. "Analytical relationships for imposing minimum length scale in the robust topology optimization formulation". In: *Structural and Multidisciplinary Optimization* 64 (2021), pp. 2429–2448. DOI: 10.1007/s00158-021-02998-w.

[25] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. "Efficient topology optimization in MATLAB using 88 lines of code". In: *Structural and Multidisciplinary Optimization* 43 (2011), pp. 1–16. DOI: 10.1007/s00158-010-0594-7.

[26] K. Svanberg. "The method of moving asymptotes—a new method for structural optimization". In: *International Journal for Numerical Methods in Engineering* 24.2 (1987), pp. 359–373. DOI: 10.1002/nme.1620240207.

[27] N. Aage, E. Andreassen, and B. S. Lazarov. "Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework". In: *Structural and Multidisciplinary Optimization* 51 (2015), pp. 565–572. DOI: 10.1007/s00158-014-1157-0.

[28] N. Aage and B. S. Lazarov. "Parallel framework for topology optimization using the method of moving asymptotes". In: *Structural and Multidisciplinary Optimization* 47 (2013), pp. 493–505. DOI: 10.1007/s00158-012-0869-2.

[29] R. Giele, C. Ayas, and M. Langelaar. "Fluid jet access in topology optimization for cleanable parts". In: *Computers and Structures* 301 (2024), p. 107420. DOI: 10.1016/j.compstruc.2024.107420.

[30] W. Li, S. Garg, and S. McMains. "2D accessibility analysis for waterjet cleaning". In: *Computer-Aided Design and Applications* 6.2 (2009), pp. 231–241. DOI: 10.3722/cadaps.2009.231-241.

[31] G. Campana, E. Uhlmann, M. Mele, L. Raffaelli, A. Bergmann, J. Kochan, and J. Polte. "Numerical investigation into cleanability of support structures produced by powder bed fusion technology". In: *Rapid Prototyping Journal* 28.3 (2022), pp. 445–452. DOI: 10.1108/RPJ-02-2021-0039.

[32] Y. Chen, J. Lu, and Y. Wei. "Topology optimization for manufacturability based on the visibility map". In: *Computer-Aided Design and Applications* 13.1 (2016), pp. 86–94. DOI: 10.1080/16864360.2015.1059199.

[33] M. Langelaar. "Topology optimization for multi-axis machining". In: *Computer Methods in Applied Mechanics and Engineering* 351 (2019), pp. 226–252. DOI: 10.1016/j.cma.2019.03.037.

[34] A. M. Mirzendehdel, M. Behandish, and S. Nelaturi. "Topology optimization with accessibility constraint for multi-axis machining". In: *CAD Computer Aided Design* 122 (2020), p. 102825. DOI: 10.1016/j.cad.2020.102825.

[35] L. C. Høghøj and E. A. Träff. “An advection–diffusion based filter for machinable designs in topology optimization”. In: *Computer Methods in Applied Mechanics and Engineering* 391 (2022), p. 114488. DOI: 10.1016/j.cma.2021.114488.

[36] A. R. Gersborg and C. S. Andreasen. “An explicit parameterization for casting constraints in gradient driven topology optimization”. In: *Structural and Multidisciplinary Optimization* 44 (2011), pp. 875–881. DOI: 10.1007/s00158-011-0632-0.

[37] G. H. Yoon and S. I. Ha. “A New Development of a Shadow Density Filter for Manufacturing Constraint and Its Applications to Multiphysics Topology Optimization”. In: *Journal of Mechanical Design, Transactions of the ASME* 143.6 (2021). DOI: 10.1115/1.4048818.

[38] J. E. Bresenham. “Algorithm for computer control of a digital plotter”. In: *IBM Systems Journal* 4.1 (1965), pp. 25–30. DOI: 10.1147/sj.41.0025.

[39] J. A. Sethian. “A fast marching level set method for monotonically advancing fronts”. In: *Proceedings of the National Academy of Sciences of the United States of America* 93.4 (1996), pp. 1591–1595. DOI: 10.1073/pnas.93.4.1591.

[40] M. Herrmann. “A domain decomposition parallelization of the Fast Marching Method”. In: *Center for Turbulence Research, NASA* (2003).

[41] J. Yang and F. Stern. “A highly scalable massively parallel fast marching method for the Eikonal equation”. In: *Journal of Computational Physics* 332 (2017), pp. 333–362. DOI: 10.1016/j.jcp.2016.12.012.

[42] Y. Luo, Q. Li, and S. Liu. “Topology optimization of shell–infill structures using an erosion-based interface identification method”. In: *Computer Methods in Applied Mechanics and Engineering* 355 (2019), pp. 94–112. DOI: 10.1016/j.cma.2019.05.017.

[43] R. Giele, C. Ayas, and M. Langelaar. “A framework for simultaneous optimization of topology and filter configurations through local density interpolation”. In: *Engineering Optimization* (under review).

[44] E. van de Ven, C. Ayas, M. Langelaar, R. Maas, and F. van Keulen. “A PDE-Based Approach to Constrain the Minimum Overhang Angle in Topology Optimization for Additive Manufacturing”. In: *Advances in Structural and Multidisciplinary Optimization. WCSMO 2017.* (2018), pp. 348–362. DOI: 10.1007/978-3-319-67988-4_89.

[45] J. Olsen and I. Y. Kim. “Design for additive manufacturing: 3D simultaneous topology and build orientation optimization”. In: *Structural and Multidisciplinary Optimization* 62 (2020), pp. 1989–2009. DOI: 10.1007/s00158-020-02590-8.

[46] C. Wang and X. Qian. “Simultaneous optimization of build orientation and topology for additive manufacturing”. In: *Additive Manufacturing* 34 (2020), p. 101246. DOI: 10.1016/j.addma.2020.101246.

[47] C. Wang. “Simultaneous optimization of build orientation and topology for self-supported enclosed voids in additive manufacturing”. In: *Computer Methods in Applied Mechanics and Engineering* 388 (2022), p. 114227. DOI: 10.1016/j.cma.2021.114227.

[48] J. Gasick and X. Qian. “Simultaneous topology and machine orientation optimization for multiaxis machining”. In: *International Journal for Numerical Methods in Engineering* 122.24 (2021), pp. 7504–7535. DOI: 10.1002/nme.6839.

[49] A. Delissen, E. Boots, D. Laro, H. Kleijnen, F. van Keulen, and M. Langelaar. “Realization and assessment of metal additive manufacturing and topology optimization for high-precision motion systems”. In: *Additive Manufacturing* 58 (2022), p. 103012. DOI: 10.1016/j.addma.2022.103012.

[50] R. Giele, C. Ayas, and M. Langelaar. “Flexible feature mapping topology optimization using NURBS-based component projection”. In: *Structural and Multidisciplinary Optimization* 68 (2025). DOI: 10.1007/s00158-025-04047-2.

[51] O. Sigmund and K. Maute. “Topology optimization approaches: A comparative review”. In: *Structural and Multidisciplinary Optimization* 48 (2013), pp. 1031–1055. DOI: 10.1007/s00158-013-0978-6.

[52] N. P. Van Dijk, K. Maute, M. Langelaar, and F. Van Keulen. “Level-set methods for structural topology optimization: A review”. In: *Structural and Multidisciplinary Optimization* 48 (2013), pp. 437–472. DOI: 10.1007/s00158-013-0912-y.

[53] F. Wein, P. D. Dunning, and J. A. Norato. “A review on feature-mapping methods for structural optimization”. In: *Structural and Multidisciplinary Optimization* 62 (2020), pp. 1597–1638. DOI: 10.1007/s00158-020-02649-6.

[54] Y. He, Z. L. Zhao, X. Lin, and Y. M. Xie. “A hole-filling based approach to controlling structural complexity in topology optimization”. In: *Computer Methods in Applied Mechanics and Engineering* 416 (2023), p. 116391. DOI: 10.1016/j.cma.2023.116391.

[55] T. Shannon, T. T. Robinson, A. Murphy, and C. G. Armstrong. “Generalized Bezier components and successive component refinement using moving morphable components”. In: *Structural and Multidisciplinary Optimization* 65.193 (2022). DOI: 10.1007/s00158-022-03289-8.

[56] R. Zheng and C. Kim. “An Enhanced Topology Optimization Approach Based on the Combined MMC and NURBS-Curve Boundaries”. In: *International Journal of Precision Engineering and Manufacturing* 21 (2020), pp. 1529–1538. DOI: 10.1007/s12541-020-00353-5.

[57] W. Zhang, L. Zhao, T. Gao, and S. Cai. “Topology optimization with closed B-splines and Boolean operations”. In: *Computer Methods in Applied Mechanics and Engineering* 315 (2017), pp. 652–670. DOI: 10.1016/j.cma.2016.11.015.

[58] J. Greifenstein, E. Letournel, M. Stingl, and F. Wein. “Efficient spline design via feature-mapping for continuous fiber-reinforced structures”. In: *Structural and Multidisciplinary Optimization* 66.5 (2023), pp. 1–14. DOI: 10.1007/s00158-023-03534-8.

[59] X. Guo, W. Zhang, J. Zhang, and J. Yuan. "Explicit structural topology optimization based on moving morphable components (MMC) with curved skeletons". In: *Computer Methods in Applied Mechanics and Engineering* 310 (2016), pp. 711–748. ISSN: 00457825. DOI: 10.1016/j.cma.2016.07.018.

[60] B. Zhu, R. Wang, N. Wang, H. Li, X. Zhang, and S. Nishiwaki. "Explicit structural topology optimization using moving wide Bezier components with constrained ends". In: *Structural and Multidisciplinary Optimization* 64.1 (2021), pp. 53–70. ISSN: 16151488. DOI: 10.1007/s00158-021-02853-y.

[61] M. P. Schmidt, P. Clausen, C. B. Pedersen, and P. Hebrard. "Semi-analytical gradient-based optimization of exact CAD models using intermediate field representations". In: *Structural and Multidisciplinary Optimization* 66.138 (2023). DOI: 10.1007/s00158-023-03595-9.

[62] G. Yi and N. H. Kim. "Identifying boundaries of topology optimization results using basic parametric features". In: *Structural and Multidisciplinary Optimization* 55 (2017), pp. 1641–1654. DOI: 10.1007/s00158-016-1597-9.

[63] L. Piegl and W. Tiller. *The NURBS Book*. Springer Berlin, Heidelberg, 1997. ISBN: 978-3-642-59223-2. DOI: 10.1007/978-3-642-59223-2.

[64] J. Austin Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley Publishing, 2009. ISBN: 978-0-470-74908-1. DOI: 10.1002/9780470749081.

[65] H. Smith and J. A. Norato. "A MATLAB code for topology optimization using the geometry projection method". In: *Structural and Multidisciplinary Optimization* (2020), pp. 1579–1594. DOI: 10.1007/s00158-020-02552-0.

[66] A. Clausen, N. Aage, and O. Sigmund. "Topology optimization with flexible void area". In: *Structural and Multidisciplinary Optimization* 50 (2014), pp. 927–943. DOI: 10.1007/s00158-014-1109-8.

[67] H. Marcuse. "Repressive tolerance". In: *Political Elites in a Democracy* (1965). DOI: 10.4324/9781315126654-8.

[68] M. Slebioda, R. Giele, and M. Langelaar. "Topology optimization for infinite fatigue life of cyclic symmetric structures subjected to non-proportional loading". In: *Computers and Structures* 286 (2023), p. 107113. DOI: 10.1016/j.compstruc.2023.107113.

[69] R. Giele, J. P. Groen, N. Aage, C. S. Andreasen, and O. Sigmund. "On approaches for avoiding low-stiffness regions in variable thickness sheet and homogenization-based topology optimization". In: *Structural and Multidisciplinary Optimization* 64 (2021), pp. 39–52. DOI: 10.1007/s00158-021-02933-z.

[70] N. Aage, R. Giele, and C. S. Andreasen. "Length scale control for high-resolution three-dimensional level set–based topology optimization". In: *Structural and Multidisciplinary Optimization* 64 (2021), pp. 1127–1139. DOI: 10.1007/s00158-021-02904-4.

Curriculum Vitæ

Reinier Jan Pieter GIELE

2020–2025	PhD in Topology Optimization Delft University of Technology
2017–2019	MSc in Aerospace Engineering Delft University of Technology
2017–2019	MSc in Wind Energy Technical University of Denmark
2013–2017	BSc in Aerospace Engineering Delft University of Technology
2015	Minor in Philosophy University of British Columbia
2007–2013	VWO Gymnasium Stanislascollege Westplantsoen

List of Publications

7. R. Giele, C. Ayas, and M. Langelaar. “Flexible feature mapping topology optimization using NURBS-based component projection”. In: *Structural and Multidisciplinary Optimization* 68 (2025). DOI: 10.1007/s00158-025-04047-2

6. R. Giele, C. Ayas, and M. Langelaar. “A framework for simultaneous optimization of topology and filter configurations through local density interpolation”. In: *Engineering Optimization* (under review)

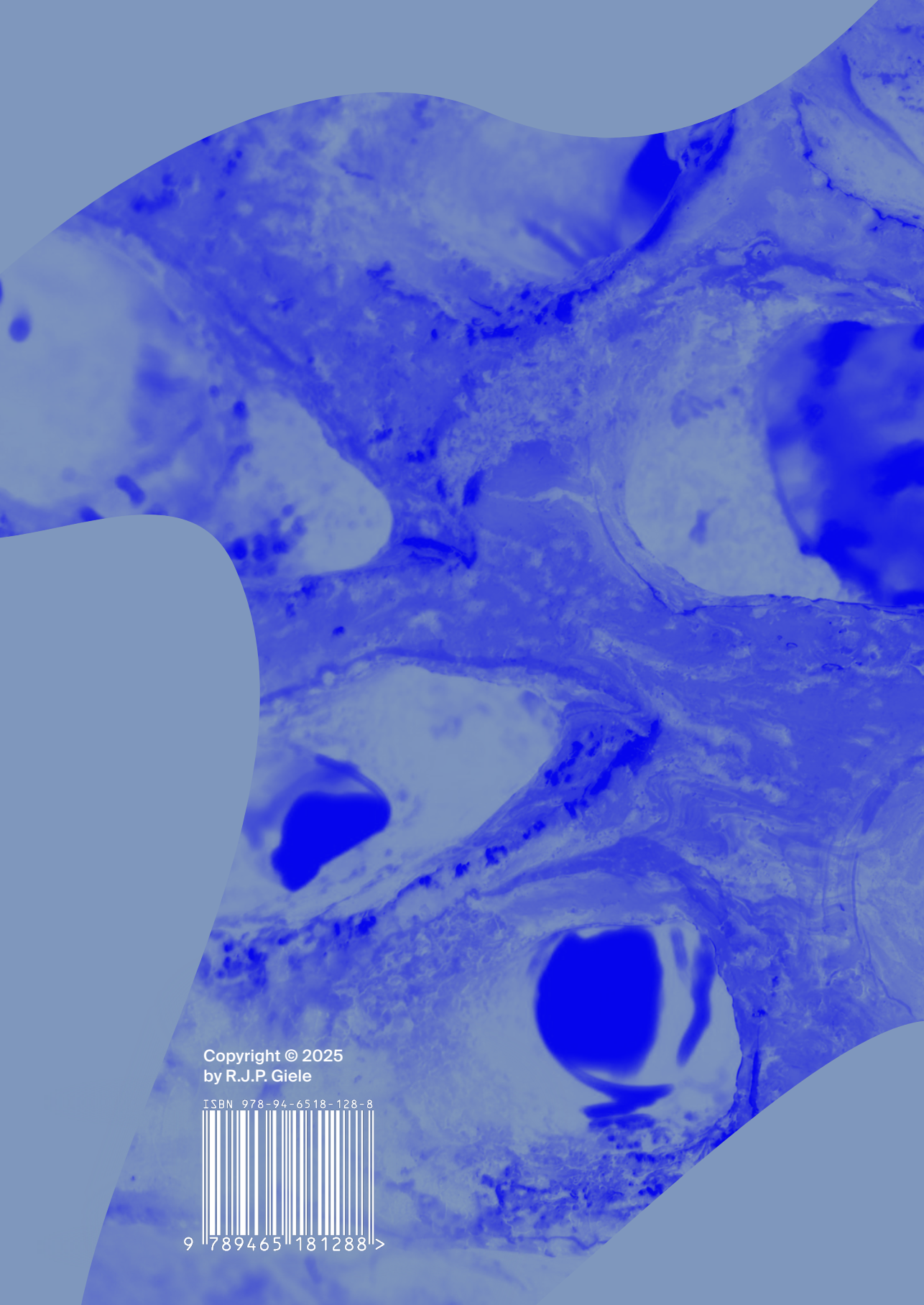
5. R. Giele, C. Ayas, and M. Langelaar. “Fluid jet access in topology optimization for cleanable parts”. In: *Computers and Structures* 301 (2024), p. 107420. DOI: 10.1016/j.compstruc.2024.107420

4. M. Slebioda, R. Giele, and M. Langelaar. “Topology optimization for infinite fatigue life of cyclic symmetric structures subjected to non-proportional loading”. In: *Computers and Structures* 286 (2023), p. 107113. DOI: 10.1016/j.compstruc.2023.107113

3. R. Giele, F. van Keulen, and M. Langelaar. “Design for drainability in density-based topology optimization”. In: *Structural and Multidisciplinary Optimization* 65.183 (2022). DOI: 10.1007/s00158-022-03272-3

2. R. Giele, J. P. Groen, N. Aage, C. S. Andreasen, and O. Sigmund. “On approaches for avoiding low-stiffness regions in variable thickness sheet and homogenization-based topology optimization”. In: *Structural and Multidisciplinary Optimization* 64 (2021), pp. 39–52. DOI: 10.1007/s00158-021-02933-z

1. N. Aage, R. Giele, and C. S. Andreasen. “Length scale control for high-resolution three-dimensional level set–based topology optimization”. In: *Structural and Multidisciplinary Optimization* 64 (2021), pp. 1127–1139. DOI: 10.1007/s00158-021-02904-4



ISBN 978-94-6518-128-8