

## Department of Precision and Microsystems Engineering

### Utilizing topology optimization to create cleanable structures

Arjen Peijen

Report no : 2022.082  
Coach : Reinier Giele  
Professor : Matthijs Langelaar  
Specialisation : CDM  
Type of report : Master thesis  
Date : 29 November 2022





# Abstract

Topology optimization is a valuable tool for the optimization of all kinds of structures. It can create highly efficient but complex designs. This complexity can make these structures challenging to clean. However, cleanability is often a requirement in, for example, the medical or food industries. Currently, no method exists to reduce the complexity and create these cleanable structures using topology optimization. The goal of this project is to create a method that can generate two-dimensional cleanable structures. In this context, cleanability has been defined using two requirements. Every section of the surface must be visible, and no sharp edges can appear on the exterior. These requirements are met by generating a structural and cleanable shell around the design. The shell needs to be optimized to minimize the compliance of the entire structure. The addition of this shell has been achieved by the use of a hybrid method. This method combines two structural optimization methods, a level-set method, and the modified SIMP approach. The level-set method acts as a shape optimization method. Its shape forms a boundary in which the modified SIMP method is used to generate a design. Material is placed at this boundary of the level-set shape to create a shell that encloses the structure. Experiments on several sets of boundary conditions show a successful creation of a shell in every case. The resulting designs are not guaranteed to be cleanable, but a satisfactory result has been achieved in every case by changing some parameters. This method sets the stage for further development toward the application of topology optimization to create cleanable designs.



# Acknowledgements

This thesis is the product of not just a year of work but a much longer process of education and personal growth. I want to extend my gratitude to those who supported me during this process. First I would like to thank my coach **Reinier** for our weekly meetings and what must have amounted to a full working week of coaching and discussions. I would also like to thank my professor **Matthijs** for his guidance and insights. A thank you to all my friends. From 'Complex', **Dennis, Eric, Lieuwe**, and **Moritz**. From 'Flux', **Jan, Maja**, and **Thomas**. From 'Hause Hermann', **Bas, Felix, Hannah, Kim**, and **Thijs**. From 'Hoezo? Ouzo', **Abel, Bert, Marco, Moritz**, and **Pieter**. From the group that is currently known as 'Arjen en de Ingenieurs', **Rens**, and **Thomas**. And the friends I have to share with Diba, **Edwin, Rohan, Shannon**, and **Shilan**. Thank you. You have been both a support and a distraction, but I could not have done it without you. I want to thank my family, **Annelies, Martin**, and **Bart** for I would not have achieved nearly as much if it were not for all the things you have done for me. I would like to thank my extra family **Darya, Gosro**, and **Ziba** for your support. Most of all I would like to thank my girlfriend **Diba**. I do not know where I would be without your love or support, you mean the world to me. I love you.

*Arjen Peijen  
Rotterdam, November 2022*



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Cleanability</b>	<b>3</b>
2.1	Establishing the requirements . . . . .	3
2.2	Improving Cleanability . . . . .	4
<b>3</b>	<b>Creating an optimal shelled structure</b>	<b>5</b>
3.1	Creating an optimal shell . . . . .	5
3.2	Combining shape and topology optimization . . . . .	7
<b>4</b>	<b>The method</b>	<b>9</b>
4.1	Creating the program . . . . .	9
4.2	Optimization methods . . . . .	9
4.2.1	SIMP method . . . . .	10
4.2.2	Level-set methods . . . . .	10
4.3	The implementation of the shell method . . . . .	11
4.4	Mapping and shell . . . . .	11
4.5	Analyzing the structure . . . . .	14
4.6	Calculating sensitivities . . . . .	15
4.7	Filtering . . . . .	17
4.8	The optimizer . . . . .	17
4.9	Controlling the shell . . . . .	18
<b>5</b>	<b>Experiments</b>	<b>21</b>
5.1	Boundary conditions . . . . .	21
5.1.1	Four points setup . . . . .	21
5.1.2	Iron setup . . . . .	22
5.2	Comparing the methods . . . . .	22
5.2.1	Four points setup . . . . .	24
5.2.2	Iron setup . . . . .	26
5.2.3	Convergence . . . . .	26
5.3	Controlling the shell . . . . .	27
5.3.1	Default values . . . . .	27
5.3.2	Shell control parameters . . . . .	29
5.4	Shell creation using different boundary conditions . . . . .	31
5.5	Comparison to adding a shell . . . . .	35
<b>6</b>	<b>Discussion</b>	<b>39</b>
6.1	Definition of cleanability . . . . .	39
6.2	The degree of control over the shell . . . . .	39
6.3	The use of local level-set functions . . . . .	39

---

<b>7 Conclusion</b>	<b>41</b>
<b>8 Future work</b>	<b>43</b>
8.1 Finding a way to quantify cleanability . . . . .	43
8.2 Extending to the 3rd dimension . . . . .	43
<b>A Density and level-set flowcharts</b>	<b>45</b>
<b>B The Heaviside function</b>	<b>47</b>
<b>C Finite difference results</b>	<b>49</b>
<b>D Parameters</b>	<b>53</b>
D.1 Maximum step sizes . . . . .	53
D.2 Beta . . . . .	53
D.3 Number of sample points . . . . .	56
<b>E Boundary condition tables</b>	<b>59</b>

# 1

## Introduction

Optimization is, in many ways, the very basis of engineering. It is optimization that turns acceptable designs into outstanding designs. These optimized designs can be generated using a computer in a process called topology optimization and be optimized for many different purposes. Examples are the reduction of weight (Cavazzuti et al., 2011, Merulla et al., 2019), improving aerodynamic performance (Muyl et al., 2004), reducing compliance (Sigmund, 2001), and many more. The designs that are created with topology optimization often perform better than conventional designs when it comes to the optimization objective. For that reason, topology optimization is now an established tool that is applied in different industries such as the automotive and aerospace industries (Bendsoe and Sigmund, 2003, Zhu et al., 2016, Papoutsis-Kiachagias and Giannakoglou, 2016).

However, designs created using topology optimization often require post-processing. One of the reasons for this is that some aspects of a design cannot be easily implemented in a topology optimization process. This can be aspects such as the cost of manufacturing, the ease of maintenance, or the subject of this report, the cleanability. Cleanability, the ability to clean or easily clean a structure, is a requirement in different industries and applications. Two examples are the medical and food industries. In these industries, keeping everything clean and avoiding contamination is an absolute necessity. On the other hand, topology optimization may result in complex designs that include features that make them challenging to clean. One example of this is features that trap fluids. These features are the subject of research done by Giele et al., 2022. Other features can, for example, be unreachable areas or tiny crevices. Take the design shown in Figure 1.1 as an example. It is imaginable that it takes a significant effort for this structure to be cleaned. That might disqualify it for use in the previously mentioned industries. To make topology optimization a feasible design method in these cases, it needs the capability to generate cleanable results but also still achieve an otherwise optimized result. In this report, an attempt is made to solve this problem for two-dimensional designs. These can be seen as cross-sections of three-dimensional structures. The method should be designed to be expandable to a three-dimensional space to be a stepping stone for future research. This report seeks to answer the following question;

How can a two-dimensional topology optimization process be extended to create optimized structures which meet the cleanability requirement?

To find the answer to this question, another question has to be resolved first. What makes a structure cleanable? Answering this question is the subject of Chapter 2, where cleanability is defined and its requirements established. Chapter 3 discusses the method for achieving



Figure 1.1: An example of a structure generated using topology optimization that seems difficult-to-clean, from Galjaard et al., 2015.

the desired result. The implementation of this method is described in Chapter 4. The result is a topology optimization program that can generate cleanable structures. Different experiments have been performed to validate the method. These experiments and their results are shown in Chapter 5. The work is then discussed in Chapter 6, and concluded in Chapter 7. Recommendations for future work are given in Chapter 8.

# 2

## Cleanability

Cleanability needs to be clearly defined to be able to create cleanable designs. In this chapter, it is discussed what factors make a structure cleanable. A design has to meet certain requirements to be considered cleanable. These requirements are defined in Section 2.1. After that, in Section 2.2, a solution is proposed to turn structures into cleanable structures.

### 2.1. Establishing the requirements

Cleaning an object can be done in numerous ways. The proposed requirements are for cleanability on a macro scale. Cleaning surfaces on a micro-scale is outside the scope of this project. The cleaning is done via an unspecified physical method. Unfortunately, little research is published on the role of the geometry of structures in their cleanability. A single example is the algorithms that can check if a water jet can reach a surface by Li et al., 2009. The authors assume that any surface that is unreachable is also uncleanable. This inherent characteristic of physical cleaning methods will become part of the requirements for cleanability for this project. What can be considered reachable differs for each cleaning method. Implementing a method to judge whether a surface is reachable would require a significant amount of time. For those reasons, judging what is and what is not reachable has been done manually for this project.

The small amount of literature on the subject requires further assumptions to be made. The main assumption is that exclusively the exterior of a design will have to be cleaned. The inside should not get dirty as long as it is completely closed off. The exterior itself then needs to be cleanable. The main problem that could arise is the accumulation of material in inward-facing corners. This material can be troublesome to remove, especially if the angle of these corners is acute. It would be too constraining to disallow such corners. For that reason, only corners with an acute angle will be considered uncleanable. It can be considered an improvement to reduce the number of inwards facing corners, but this will not be a requirement. In short: The two requirements are that every surface has to be reachable, and acute angles may not be present on the exterior. The structure can be considered cleanable if it meets these two requirements.

Three examples are shown in Figure 2.1 to further illustrate the requirements. Design A on the left can be considered cleanable, as it meets all the requirements. Design B is also cleanable, it has no difficult-to-reach places and the angle of the bottom corners is not quite acute. This structure could be easier to clean when these corners are no longer present. There is a difficult-to-reach area present in design C and it is therefore not cleanable.

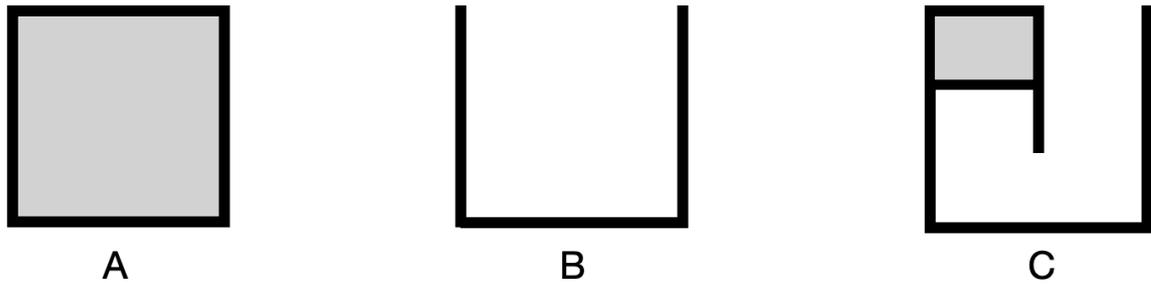


Figure 2.1: Three example designs to illustrate the cleanability requirements. Examples A and B are cleanable, as the corners of example B are not quite acute. Example C is uncleanable as it contains an unreachable area.

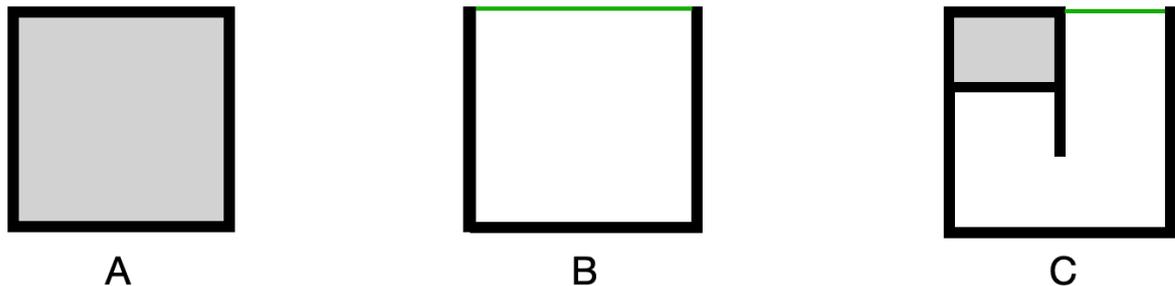


Figure 2.2: The designs of Figure 2.1 with an additional green shell. All designs now meet the requirements due to the addition of this shell.

## 2.2. Improving Cleanability

In the previous section, cleanability has been defined and the requirements have been established. Now, it is possible to create a method to make the designs in Figure 2.1 cleanable. It is established that only the shape of the exterior of the design impacts the cleanability. A solution is to enclose the structures generated using topology optimization within a shell. The solution is focused on the exterior to minimize the impact of the cleanability requirement on the overall design. This solution can be seen in Figure 2.2. These are the boxes from Figure 2.1 with additional material, shown in green, enclosing the interior. Nothing changes for structure A, but structures B and C now meet the requirements and are as easy to clean as structure A. Encapsulating a design within a shell can significantly reduce the complexity of the design when viewed from the outside. The shape of the shell is then the only attribute dictating the cleanability. Controlling this shape means controlling the cleanability.

To conclude, cleanability has been defined as only being dependent on the exterior of a structure. Every surface has to be reachable, and acute angles may not be present on this exterior. Creating a shell around such a structure will render it cleanable while having a minimal impact on the design.

# 3

## Creating an optimal shelled structure

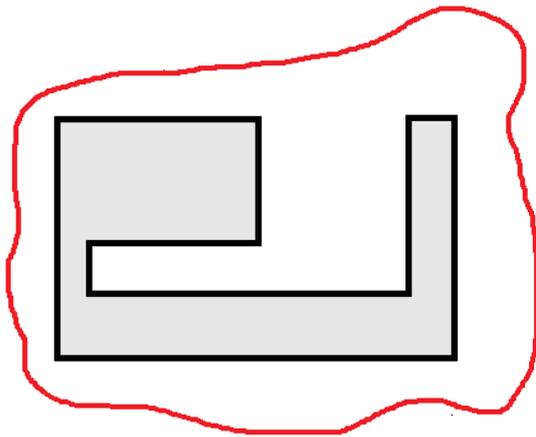
It is established in the previous chapter that structures will usually become cleanable by enclosing them. In this chapter, it is discussed how to create an optimal shelled structure. Section 3.1 discusses how to create an optimal shell using topology optimization. Section 3.2 discusses combining the shell with topology optimization.

### 3.1. Creating an optimal shell

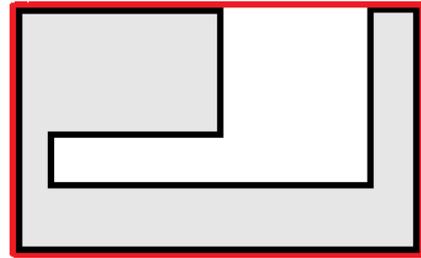
The goal of topology optimization is to create an optimal design. Thus, the performance of a cleanable design should be as close to the optimal uncleanable design as possible. This means that the shape of the cover should contribute as much as attainable to the optimization objective.

It is often possible to add secondary requirements to the topology optimization problem by making them a constraint or part of this objective. Unfortunately, this implementation is not feasible for cleanability. The main reason is that cleanability is challenging to quantify as it is a discrete objective. A design is either cleanable or uncleanable. This quantification problem means that methods for multi-objective topology optimizations will not work. There are other ways to add material to the design without it being an objective. It is possible to use a filter and place material around surfaces that are found to be uncleanable. The main problem with a filter is that many solutions exist to make a structure cleanable. Creating a filter that finds the optimal solution for any set of boundary conditions is challenging.

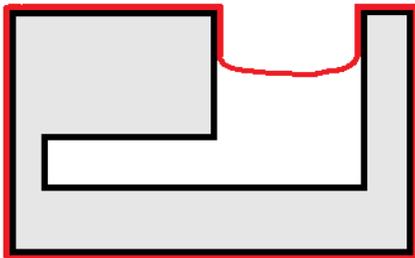
The alternative to explicitly implementing cleanability via an objective is to implicitly add it by forcing the shell into the design. The optimizer should be able to change the shape of this shell to improve its optimization objective but be unable to remove it. Otherwise, this shell would usually be gotten rid of during the process for not contributing enough to the objective. A topology optimization method allows for the creation of holes in the shell and lets the optimizer remove it. A shape optimization method, such as the level-set method, is more suited for this purpose. Such a method keeps the topology of the shell consistent, only allowing the shape to change. This shape can be optimized for the objective while feasibly meeting the cleanability requirements. Figure 3.1 shows a hypothetical shape optimization process of the shell. It is expected that the shell will wrap around the structure, as this structure represents the most optimal shape. The gaps will then be covered, as can be seen in Figure 3.1b. The shell could converge further inwards as can be seen in Figure 3.1c. This behavior is desirable as it improves the objective while still meeting the cleanability requirements. The curvature of the shell does have to be limited to prevent the behavior that can be seen in Figure 3.1d. In this scenario, the shell has merged with the original structure and the design is reverted to being uncleanable.



(a) A design with an initial shape of a shell.



(b) The shell is fitted around the exterior but bridges the gap. This design is cleanable.



(c) The shell converges into the gap. This design is also cleanable.



(d) The shell is fully converged and has become one with the rest of the structure. This design has most likely the best objective value, but it is no longer cleanable.

Figure 3.1: The shape optimization of the shell, shown in red. From top left to bottom right, this describes a hypothetical optimization process.

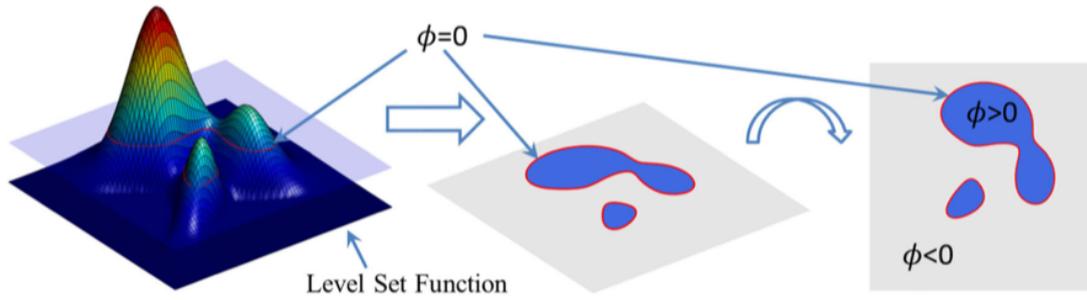


Figure 3.2: An illustration of the level-set function. This function is transformed into a two-dimensional design. The boundaries of this design are shown in red. In the hybrid method, this boundary acts as a boundary of the design domain. Figure taken from Wei et al., 2018.

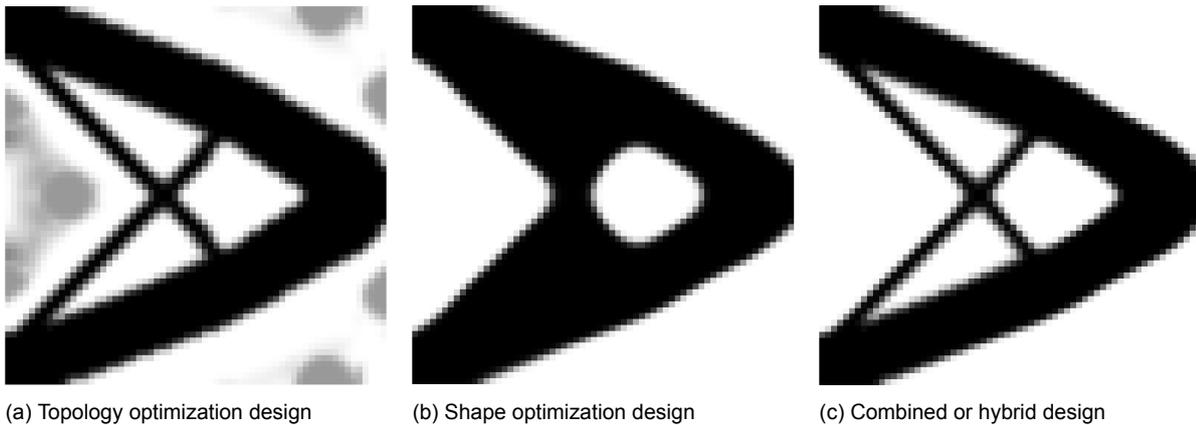


Figure 3.3: Three figures showing how the individual designs combine to create a new hybrid design. It can be seen that only elements that are present in both designs end up in the hybrid design.

### 3.2. Combining shape and topology optimization

Shape optimization is not necessarily the best method to optimize the rest of the structure. The results achieved using shape optimization methods depend strongly on the initial designs. A topology optimization method does not come with this downside. Therefore such a method is more suitable to use for the interior. It is possible to combine shape optimization and topology optimization. Combining these has been done before by Geiss and Maute, 2018 and Giele et al., 2021. In these methods, a level-set and a density method are combined. These methods will be further explained in Section 4.2. The level-set method is used for shape optimization. The level-set function defines a boundary, this boundary can be seen in Figure 3.2. Inside this boundary, a density method is used to create a structure. Material is only placed where both fields overlap, an example can be seen in Figure 3.3. Effectively, this level-set function acts as a boundary to the design domain. Making the elements around this boundary solid would result in an enclosed design. This solution combines the best of both worlds. An optimal shelled design should emerge by optimizing this boundary and the rest of the structure inside it. The exact implementation of this method is discussed in the next chapter. The combination of the level-set and density methods will be referred to as ‘the hybrid method’ The hybrid method with an additional shell will be referred to as ‘the shell method.’



# 4

## The method

In the previous chapter, the use of a hybrid method to create cleanable structures has been presented. In this chapter, the implementation of the shell method to generate these cleanable structures is discussed. This method is designed to have minimization of compliance as an optimization objective. This objective is the only one that has been considered. Different objectives might work for this method, but these have not been tested. The programming language and resources used to create the program are discussed in Section 4.1. The density and the level-set method are discussed in Section 4.2. These methods help form the shell method. The implementation of the shell method is explained in Section 4.3. Then, in Section 4.4, the mapping and shell are examined. In Section 4.5, the finite element analysis (FEA) is shortly discussed. Then in Section 4.6, the calculation of the sensitivities is explained. After that, in Section 4.7, the density filter is briefly discussed. The optimizer is discussed in Section 4.8. Lastly, multiple ways to control the behavior of this shell are discussed in Section 4.9.

### 4.1. Creating the program

The program which implements the method outlined in this chapter is created using the MATLAB programming language. This language can perform the many required matrix operations relatively quickly. Beyond that, it has been the programming language of choice for many topology optimization programs. Many of those programs are well documented. A lot of the code for this program has been influenced by or taken from such programs. The finite element method and the simplified isotropic material with penalization (SIMP) approach have been inspired by or taken from the ‘top88’ program by Andreassen et al., 2011. The implementation of the method of moving asymptotes and level-set mapping are influenced by or taken from the ‘topcut’ program by Andreasen et al., 2020. The program can use the modified SIMP, the level-set, the hybrid method, and the shell method. This ability makes it easier to test the different methods separately and to keep changes consistent across them.

### 4.2. Optimization methods

In the previous chapter, it was mentioned that the hybrid method uses density methods and level-set methods. These have been selected because they are both commonly used methods with many available resources and this combination has been successfully implemented by Geiss and Maute, 2018 and Giele et al., 2021. In this section, the density and level-set methods will be discussed.

### 4.2.1. SIMP method

The topology optimization method used in the hybrid function is the Simplified Isotropic Material with Penalization (SIMP) method. The SIMP method is a density method. This is a group of methods that describe the structure using intermediate densities. This means that the elements in the design are not just solid or void, but also anything in between. These intermediate densities turn the discrete design variables into continuous variables. Bendsøe and Kikuchi, 1988 introduced the first density method. This density method has then been improved by Bendsøe, 1989, introducing the SIMP method. The SIMP method and its derivatives remain some of the most popular methods to date. The SIMP method calculates the Young's modulus of each element using Equation 4.1.

$$E_e(\rho_e) = \rho_e^p E_0 \quad (4.1)$$

The Young's modulus of each element,  $E_e$ , is the default Young's modulus,  $E_0$ , multiplied by the density of the element,  $\rho_e$ . The density can be any value between 1, representing a solid element, and 0, representing a void element. The density is penalized by a factor of  $p$ . The penalty reduces the Young's modulus of intermediate densities, thereby encouraging the use of either solid or void elements. The result of this equation can not be exactly 0 as this would lead to matrix singularity during the analysis of the design. This results in a minimum density value. Sigmund, 2007 has proposed a slight modification of the original SIMP approach. This is called the modified SIMP approach. This approach introduces a minimum Young's modulus for each element,  $E_{min}$ . This can be seen in Equation 4.2.

$$E_e(\rho_e) = E_{min} + \rho_e^p (E_0 - E_{min}) \quad (4.2)$$

The minimum value,  $E_{min}$ , is several orders of magnitude smaller than  $E_0$ . For this project the value of  $E_0$  is 1, the value of  $E_{min}$  is  $10^{-9}$ . The values of  $E_e$  are needed to analyze the design, this is further explained in Section 4.5.

### 4.2.2. Level-set methods

Osher and Sethian, 1988 first introduced the use of the level-set function to describe shapes. Later, it has been suggested to use the level-set function for topology optimization by Haber and Bendsøe, 1998. Level-set methods use the level-set function  $\phi(x)$  to describe a design. The values of  $\phi(x)$  are linked to the material distribution by Equation 4.3.

$$\rho = \begin{cases} \phi(x) < c & : 0 & \text{(void)} \\ \phi(x) = c & : \Gamma & \text{(boundary)} \\ \phi(x) > c & : 1 & \text{(material)} \end{cases} \quad (4.3)$$

The material is represented by  $\rho$ . Conventionally, the values of  $\phi$  range from -1 to 1, and the value of  $c$  is usually 0. The interface is defined by  $\phi(x) = c$ . This is the boundary between the solid material of the areas where  $\phi$  is larger than  $c$  and the void where  $\phi$  is smaller than  $c$ . This boundary is neither material nor void and is represented using the letter  $\Gamma$ . The  $\phi$  function consists of multiple smaller basis functions. These basis functions can affect one or multiple nodal values. The use of nodal values differs from the density method where element values are used. There are many different possible basis functions. The level-set function used in this project uses bilinear local basis functions on a fixed mesh. A visual example can be seen in Figure 4.1. These functions are relatively easy to implement and straightforward to extend to a third dimension (Van Dijk et al., 2013). These authors also note the downside that a large number of iterations is needed because each function only affects a single node.

The usage of nodal values means that these values have to be transformed into element values. There are many different approaches to this problem. Only the approach that has

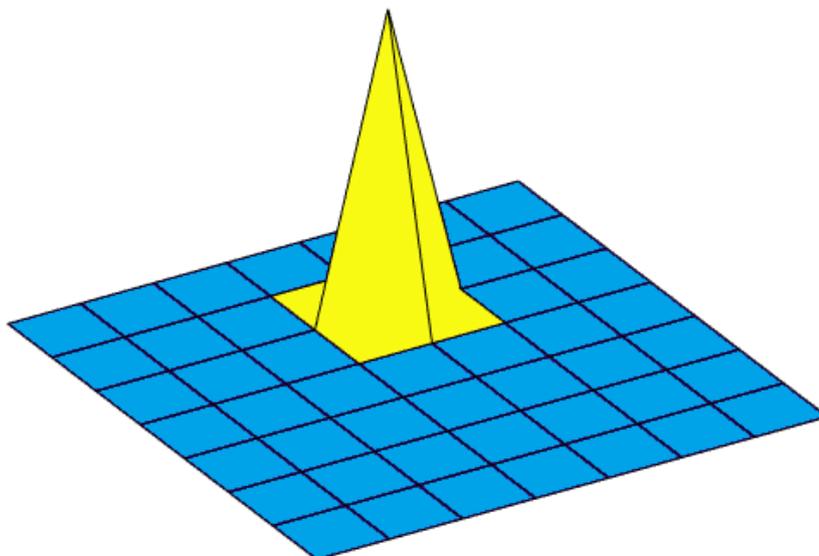


Figure 4.1: A nearly uniform field with a single bilinear local basis function with a higher value. The linear shape and the fact that it only affects a single node can be seen. Each nodal value is connected to a different basis function.

been used for this project will be discussed. This discussion can be found in Section 4.4. This implementation of the level-set method is not strictly a shape optimization. Changes to the topology are discouraged by reducing the sensitivities, but not impossible.

### 4.3. The implementation of the shell method

The shell method combines the modified SIMP method with the level-set method. Figure 4.2 shows a simplified flowchart of a part of the hybrid process. This figure can help visualize the implementation described in this section. The simplified flowcharts of the density and level-set method are found in Appendix A. The hybrid method combines the density field,  $\mathbf{x}_f$ , and the level-set field,  $\phi_f$ . The 'f' subscript in the figure indicates that the fields have been filtered using a density filter. The density filter is explained in Section 4.7. These fields form the basis and are the design variables of the method. To merge the fields, the nodal field  $\phi_f$  is transformed into an element field. This process is explained in depth in Section 4.4. One of the two fields to result from this operation is  $\mathbf{s}_f$ . This field is combined with the  $\mathbf{x}_f$  field. Using element-wise multiplication of the  $\mathbf{x}_f$  and  $\mathbf{s}_f$  fields, the hybrid field  $\mathbf{x}\mathbf{s}_f$  is created. This multiplication means that material is placed only in locations where both fields are not void. Figure 4.3 shows the result of this operation. To turn this hybrid method into a shell method, material has to be placed near the boundary of the level-set function. This placement is done via a separate field, the shell field,  $\mathbf{s}_{shell,f}$ . This shell field is the second field to emerge from the mapping process. This field is added to the  $\mathbf{x}\mathbf{s}_f$  field, resulting in the final  $\xi_{ph}$  field. This addition means that the shell is unaffected by the values of the density field, and therefore it should always be present. This is further explained in the next section.

### 4.4. Mapping and shell

The mapping of the level-set function to the density field plays a key role in unifying both methods. The nodal values of  $\phi_f$  are mapped to the element density values  $\mathbf{s}_f$  and  $\mathbf{s}_{shell,f}$ . The goal is to find the volume fraction, or density, of each element. The first step is to find the values of  $\phi_f$  inside the element using the shape functions. Then two different volume fractions

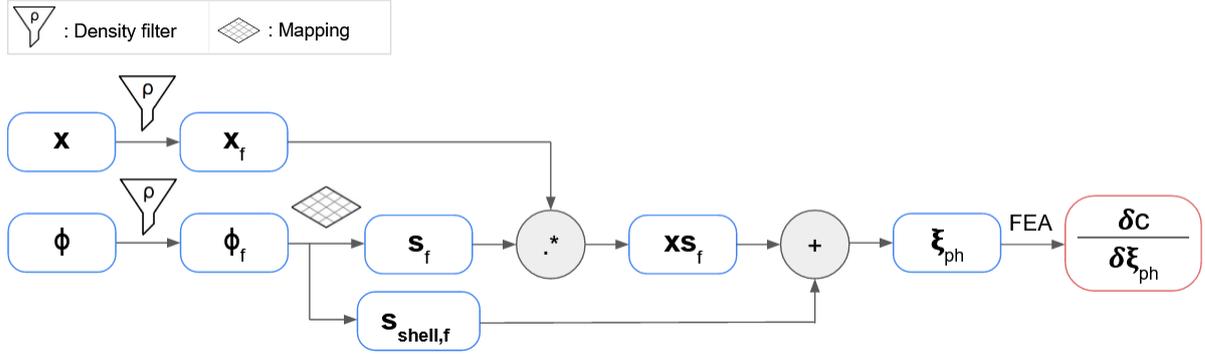


Figure 4.2: Simplified flowchart of the first steps taken for the hybrid method. The  $\phi_f$  field is mapped to the  $\mathbf{s}_f$  and  $\mathbf{s}_{shell,f}$  fields. The  $\mathbf{s}_f$  field is then combined with the  $\mathbf{x}_f$  field using element-wise multiplication. The result is a new field,  $\mathbf{x}\mathbf{s}_f$ . The  $\mathbf{s}_{shell,f}$  field is added to the  $\mathbf{x}\mathbf{s}_f$  field. This addition results in the final shelled hybrid field,  $\xi_{ph}$ . A finite element analysis is then performed and the sensitivities towards the objective  $c$  of  $\xi_{ph}$  are calculated.

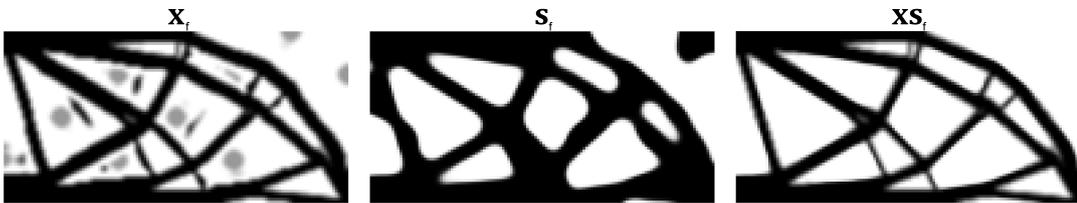


Figure 4.3: The combination of the different fields. Left: the density field  $\mathbf{x}_f$ . Middle: the mapped level-set field  $\mathbf{s}_f$ . Right: the combination of the fields  $\mathbf{x}\mathbf{s}_f$ . This field only consists of material present in both  $\mathbf{x}_f$  and  $\mathbf{s}_f$ .

are calculated, one for the  $\mathbf{s}_f$  field and one for the  $\mathbf{s}_{shell,f}$  field, this is done using two different Heaviside functions. Numerical integration is then used to calculate the densities.

The value of  $\phi_f$  inside each element, henceforth referred to as the local values, is evaluated at several points. The number of points in each direction is represented by  $n$ , making the total number of points  $n^2$ . This can be seen in Figure 4.4. The location of the points is chosen in such a way that each point represents a fraction of  $1/n$  of the total area of the element. This means that the distance of each point to the edge is  $1/2n$  and the distance of each point to the nearest points is  $1/n$ . Using a Gauss–Legendre quadrature with the corresponding points and weights would lead to a smaller error. However, due to time constraints and its more difficult implementation, this method has not been selected. The increase in the error from using a more basic method is expected to be negligible when enough sample points are used. To find

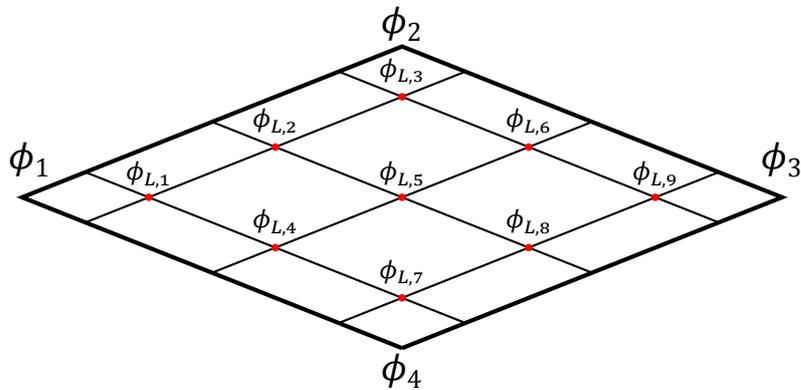


Figure 4.4: The location of the local phi variables. In this example, the number of points in each direction,  $n$ , is 3.

the value of  $\phi_f$  at each point the shape functions are used. The shape functions are bilinear interpolations and can be expressed as the following equations:

$$N_{\phi_1}(x, y) = \frac{(1-x)(1-y)}{4} \quad (4.4)$$

$$N_{\phi_2}(x, y) = \frac{(1-x)(1+y)}{4} \quad (4.5)$$

$$N_{\phi_3}(x, y) = \frac{(1+x)(1+y)}{4} \quad (4.6)$$

$$N_{\phi_4}(x, y) = \frac{(1+x)(1-y)}{4} \quad (4.7)$$

Each local value,  $\phi_{L,i}$ , can then be found using Equation 4.8. Here the variables  $x_i$  and  $y_i$  represent the coordinates of the sample point.

$$\phi_{L,i}(x_i, y_i) = N_{\phi_1}(x_i, y_i) + N_{\phi_2}(x_i, y_i) + N_{\phi_3}(x_i, y_i) + N_{\phi_4}(x_i, y_i) \quad (4.8)$$

The  $\phi_L$  values are used to calculate two different volume fractions. One for the  $\mathbf{s}_f$  field and one for the  $\mathbf{s}_{shell,f}$  field. This is done by applying two different smooth Heaviside functions. The smooth Heaviside function reduces some sensitivities in the level-set field by mapping a range of inputs to nearly identical outputs, which mainly takes place in areas where the level-set function is solid or void. This helps prevent changes in topology while still allowing for changes at the boundaries. These functions also adjust the values of  $\phi_f$  to be between 0 and 1 per the allowed density values. More information about the Heaviside function can be found in Appendix B. The Heaviside function for  $\mathbf{s}_f$  is shown as Equation 4.9. The Heaviside function for  $\mathbf{s}_{shell,f}$  is shown as Equation 4.10.

$$\phi_{H,i} = \frac{\tanh \beta + \tanh(\beta(\phi_{L,i} - \eta_1))}{\tanh \beta + \tanh(\beta(1 - \eta_1))} \quad (4.9)$$

$$\phi_{H,shell,i} = \frac{\tanh \beta + \tanh(\beta(\phi_{L,i} - \eta_0))}{\tanh \beta + \tanh(\beta(1 - \eta_0))} - \frac{\tanh \beta + \tanh(\beta(\phi_{L,i} - \eta_1))}{\tanh \beta + \tanh(\beta(1 - \eta_1))} \quad (4.10)$$

The variable  $\eta_1$  controls where the Heaviside threshold is. Inputs below this threshold are decreased, and inputs above the threshold are increased. The parameter  $\beta$  is used to control the steepness of the Heaviside function. These Heaviside functions are also visualized in a graph, Figure 4.5. In this figure, the Heaviside function of  $\mathbf{s}_{shell,f}$  is red, and  $\mathbf{s}_f$  is blue. The resulting local values can then be used to calculate the volume fractions using numeric integration, shown as Equation 4.11 and Equation 4.12.

$$\mathbf{s}_i = \frac{\sum_{j=1}^{n^2} \phi_{H,j}}{n^2} \quad (4.11)$$

$$\mathbf{s}_{shell,i} = \frac{\sum_{j=1}^{n^2} \phi_{H,shell,j}}{n^2} \quad (4.12)$$

Looking back at Figure 4.5, the colored areas show in which value of  $\phi_L$  contributes to which field. It is important to note that an element can have a value in both fields. Increasing the value of  $\beta$  reduces this overlap. Both  $\beta$  and  $\eta_1$  must have the same value in both equations. This parity ensures the total density will not be higher than 1 when adding each element of the

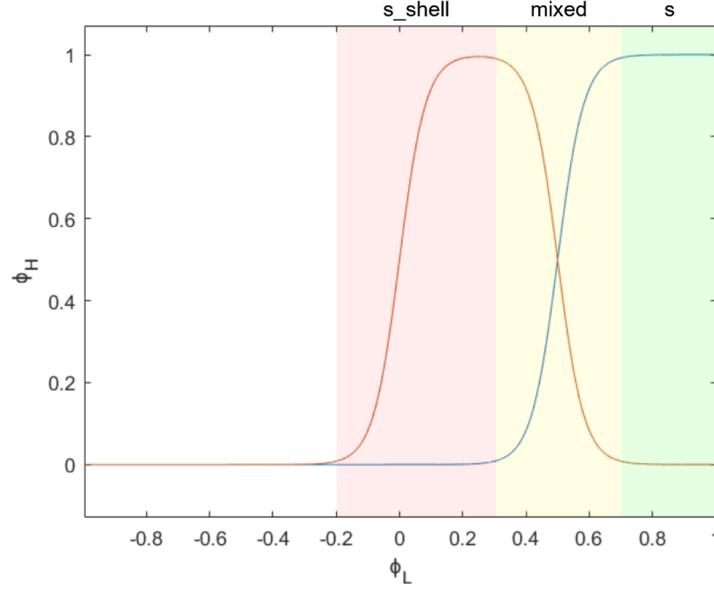


Figure 4.5: Graph visualising the split of  $\phi$  between the  $\mathbf{s}$  and  $\mathbf{s}_{shell}$  field. The Heaviside function for the shell, Equation 4.10, is shown in red. The Heaviside function for  $\mathbf{s}$ , Equation 4.9, is shown in blue. The colored areas denote the different fields where the material will end up. For this graph  $\beta = 12$ ,  $\eta_0 = 0$ , and  $\eta_1 = 0.5$

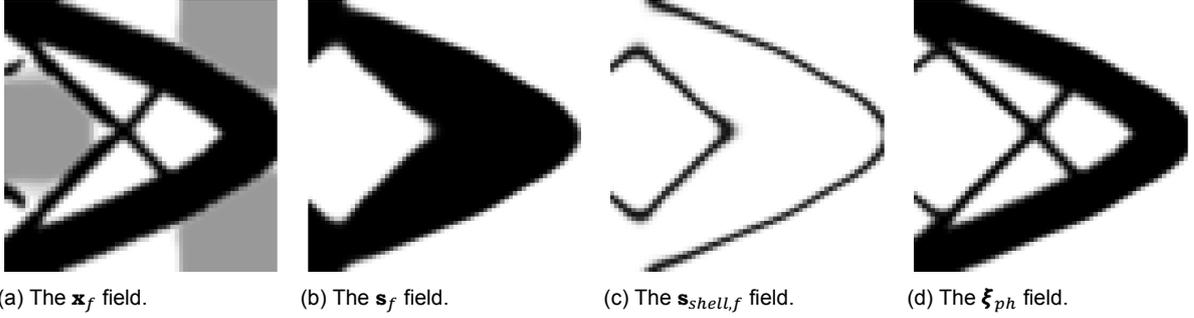


Figure 4.6: Four fields of a cantilever beam problem, optimized using the shell method. It can be seen that the  $\mathbf{s}_{shell,f}$  field surrounds the  $\mathbf{s}_f$  field when it is not connected to the edge of the design domain.

fields. The result of this operation is that material defined by the multiplication of  $\mathbf{x}_f$  and  $\mathbf{s}_f$  will always be surrounded by material of  $\mathbf{s}_{shell,f}$ , assuming reasonable values for the relevant parameters. The individual fields for a cantilever problem can be seen in Figure 4.6. It can be seen that the shell field in Figure 4.6c surrounds the  $\mathbf{s}_f$  field seen in Figure 4.6b.

#### 4.5. Analyzing the structure

After merging the fields, a finite element analysis (FEA) is performed on the shelled hybrid field,  $\xi_{ph}$ . The finite element analysis calculates the displacement of each element in the structure. The stiffness of each element is calculated using the modified SIMP approach. The optimization problem of the modified SIMP approach can mathematically be expressed as follows:

$$\begin{aligned}
 \min \xi : \quad & c(\xi) = \mathbf{U}^T \mathbf{K} \mathbf{U} = \sum_{i=1}^N E_i(\xi_i) \mathbf{u}_i^T \mathbf{k}_0 \mathbf{u}_i \\
 \text{s.t. :} \quad & \frac{V_\xi}{V_0} \leq f \\
 & \mathbf{K} \mathbf{U} = \mathbf{F} \\
 & \mathbf{0} < \xi < \mathbf{1}
 \end{aligned} \tag{4.13}$$

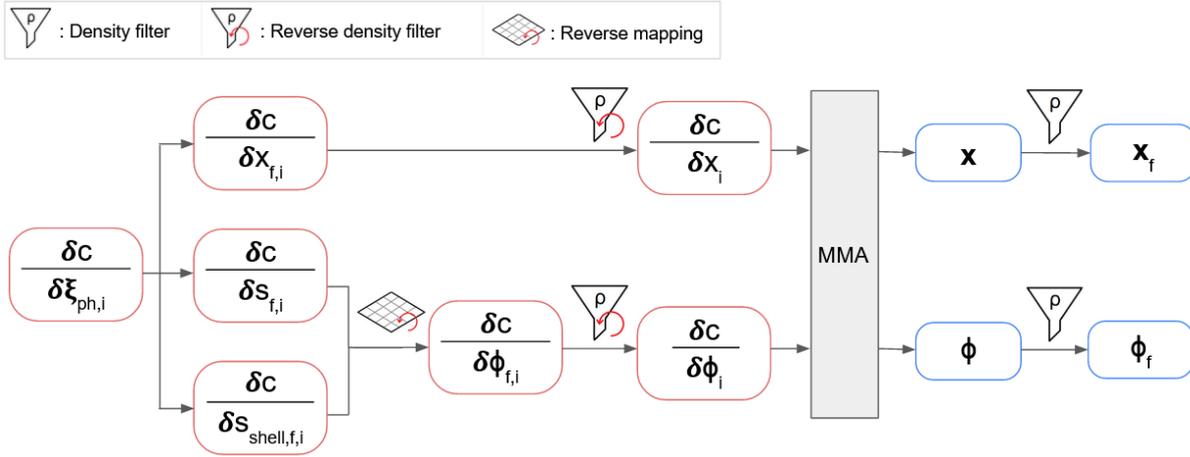


Figure 4.7: Simplified flowchart of the sensitivity calculations, the optimization, and the filtering steps. The sensitivities of  $\xi_{ph}$  are used to calculate the sensitivities of  $\mathbf{x}_f$ ,  $\mathbf{s}_f$ , and  $\mathbf{s}_{shell,f}$ . The sensitivities of  $\mathbf{s}_f$  and  $\mathbf{s}_{shell,f}$  are mapped back to the nodes and combined to calculate the sensitivities of  $\phi_f$ . The sensitivities of the design variables are filtered back and fed to the MMA optimizer. New  $\mathbf{x}$  and  $\mathbf{s}$  fields are found and filtered. The filtered design variables are then inserted back into the optimization loop until it is converged.

In this equation,  $c$  is the compliance. This value is a product of the global displacement vector  $\mathbf{U}$  and the global stiffness matrix  $\mathbf{K}$ . This product, in turn, is the result of the sum of the product of the Young's modulus of each element, the element displacement vector,  $\mathbf{u}_i$ , and the element stiffness matrix,  $\mathbf{k}_0$ .  $N$  is the total number of elements. The volume constraint is expressed using the volume fraction  $f$ , the total material volume  $V_\xi$ , and the total domain volume  $V_0$ . The vector  $\mathbf{F}$  is the global force vector. The force vector is part of the boundary conditions, together with a set of fixed elements. Using these, the stiffness matrix of a design can be calculated using a finite element analysis (FEA). For this, a rectangular Cartesian mesh is used. The implementation is identical to the one used in the top88 MATLAB program by Andreassen et al., 2011. For this implementation, Poisson's ratio is set to 0.3.

## 4.6. Calculating sensitivities

The objective sensitivities are calculated by differentiating the displacements with respect to the density. This differentiation results in the sensitivities of the  $\xi_{ph}$  field. This differentiation is performed for each element individually. This is expressed as the following equation:

$$\frac{\partial c_i}{\partial \xi_{ph,i}} = -p \cdot (E_0 - E_{min}) \cdot \xi_{ph,i}^{(p-1)} \cdot c_i \quad (4.14)$$

In this equation,  $\partial c_i / \partial \xi_{ph,i}$  expresses the sensitivity of an element of the hybrid field.  $c_i$  is the element objective. The subscript 'i' indicates that this is the value of a single element or node. The elements of  $\xi_{ph,i}$  can be expressed as follows:

$$\xi_{ph,i} = x_{f,i} \cdot s_{f,i} + s_{shell,f,i} \quad (4.15)$$

The next steps in the hybrid process are visualized in Figure 4.7. From this, the sensitivities of the  $\mathbf{x}_f$ ,  $\mathbf{s}_f$ , and  $\mathbf{s}_{shell,f}$  fields can be calculated. This is done for each element using equations 4.16, 4.17, and 4.18 respectively.

$$\frac{\partial c_i}{\partial x_{f,i}} = \frac{\partial c_i}{\partial \xi_{ph,i}} \cdot \frac{\partial \xi_{ph,i}}{\partial x_{f,i}} = \frac{\partial c_i}{\partial \xi_{ph,i}} \cdot s_{f,i} \quad (4.16)$$

$$\frac{\partial c_i}{\partial s_{f,i}} = \frac{\partial c_i}{\partial \xi_{ph,i}} \cdot \frac{\partial \xi_{ph,i}}{\partial s_{f,i}} = \frac{\partial c_i}{\partial \xi_{ph,i}} \cdot x_{f,i} \quad (4.17)$$

$$\frac{\partial c_i}{\partial s_{shell,f,i}} = \frac{\partial c_i}{\partial \xi_{ph,i}} \cdot \frac{\partial \xi_{ph,i}}{\partial s_{shell,f,i}} = \frac{\partial c_i}{\partial \xi_{ph,i}} \quad (4.18)$$

To calculate the  $\phi_{f,i}$  sensitivities from the  $s_{f,i}$  and  $s_{shell,f,i}$  sensitivities, the values have to be mapped back, as shown in Equation 4.19.

$$\frac{\partial c_i}{\partial \phi_{f,i}} = \frac{\partial c_i}{\partial s_{f,i}} \cdot \frac{\partial s_{f,i}}{\partial \phi_{f,i}} + \frac{\partial c_i}{\partial s_{shell,f,i}} \cdot \frac{\partial s_{shell,f,i}}{\partial \phi_{f,i}} \quad (4.19)$$

As mentioned before, the  $\mathbf{s}$  and  $\mathbf{s}_{shell}$  fields are the result of the sum of interpolation and a Heaviside function. The derivatives of the Heaviside equations 4.9 and 4.10 are shown as equations 4.20 and 4.21 respectively.

$$\frac{\partial \phi_H}{\partial \phi_L} = \beta \frac{\text{sech}^2(\beta(\phi_L - \eta_1))}{\tanh \beta + \tanh(\beta(1 - \eta_1))} \quad (4.20)$$

$$\frac{\partial \phi_{H,shell}}{\partial \phi_L} = \beta \left( \frac{\text{sech}^2(\beta(\phi_L - \eta_0))}{\tanh \beta + \tanh(\beta(1 - \eta_0))} - \frac{\text{sech}^2(\beta(\phi_L - \eta_1))}{\tanh \beta + \tanh(\beta(1 - \eta_1))} \right) \quad (4.21)$$

Using this result, the partial derivatives of  $\mathbf{s}_f$  and  $\mathbf{s}_{shell,f}$  with regards to  $\phi_f$  can be calculated using Equations 4.22 and 4.23. In this equation,  $N_{\phi_i}$  represents the 4 shape functions.

$$\frac{\partial s_{f,i}}{\partial \phi_{f,i}} = \sum_{i=1}^4 \sum_{j=1}^{n^2} \left( \frac{\partial \phi_{H,i,j}}{\partial \phi_{L,i,j}} \right) N_{\phi_i}(x_j, y_j) \quad (4.22)$$

$$\frac{\partial s_{shell,f,i}}{\partial \phi_{f,i}} = \sum_{i=1}^4 \sum_{j=1}^{n^2} \left( \frac{\partial \phi_{H,shell,i,j}}{\partial \phi_{L,i,j}} \right) N_{\phi_i}(x_j, y_j) \quad (4.23)$$

The second set of sensitivities that need to be calculated is the volume sensitivities. This is the impact the change in density has on the total volume. The volume sensitivities for  $\mathbf{x}_f$ ,  $\phi_f$  can be calculated per element and per node using Equation 4.24 and Equation 4.25 respectively.

$$\frac{\partial V_\xi}{\partial x_{f,i}} = \frac{s_{f,i}}{N_{el} \cdot f} \quad (4.24)$$

$$\frac{\partial V_\xi}{\partial \phi_{f,i}} = \left( \frac{\partial s_{f,i}}{\partial \phi_{f,i}} \cdot x_{f,i} + \frac{\partial s_{shell,f,i}}{\partial \phi_{f,i}} \right) \cdot \frac{1}{N_{el} \cdot f} \quad (4.25)$$

Where  $V_\xi$  is the volume,  $N_{el}$  represents the total number of elements, and  $f$  is the desired volume fraction. The calculated sensitivities have been verified using a finite difference comparison. This comparison can be found in Appendix C.

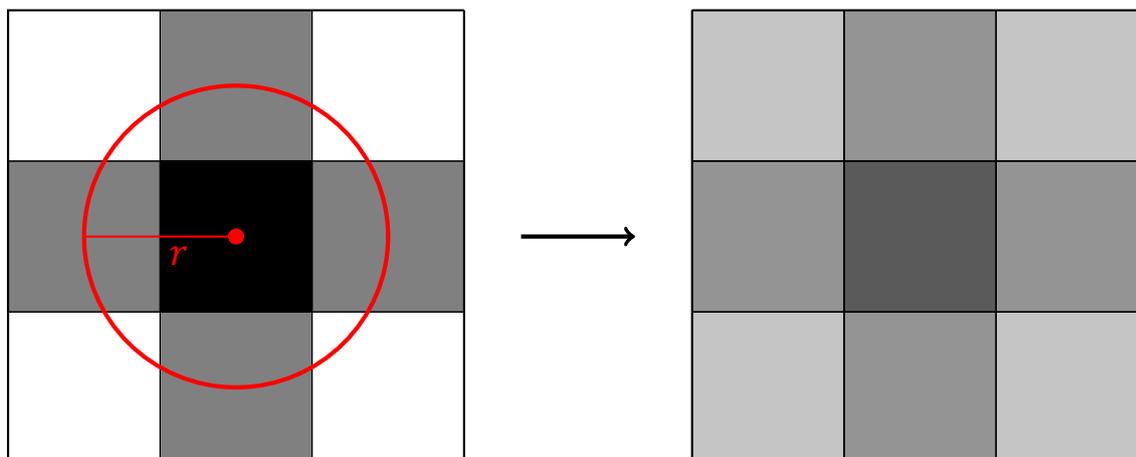


Figure 4.8: The effect of the density filter on a grid of 3 by 3 elements. On the left the grid consists of a solid element in the center surrounded by four intermediate-density elements, the corners are void. Shown in red is a filter with a radius  $r$ . On the right, the filtered densities are shown. The center element is no longer fully solid and the corner elements are no longer fully void.

## 4.7. Filtering

The shell method uses a density filter (Bruns and Tortorelli, 2001). This filter uses a mesh-independent area defined by a radius  $r$ . In this area, the weighted average of the design variables is used to calculate a new value. This calculation is performed for each element, creating a new design field. This process can be seen in Figure 4.8. It can be seen that the density filter increases the number of intermediate densities. This effect is visible at the boundaries between solid and void in the structure, resulting in blurry transitions between solid and void at these edges. It is important to note that the density filter preserves the volume of the elements that are filtered. In Figure 4.8 the density filter is demonstrated on element values, but it can be used on nodal values too. The filtered design field is used to calculate the sensitivities. The sensitivities have to be filtered back to obtain the sensitivities for the original design.

In the shell method, a density filter is applied to the  $\mathbf{x}$  and  $\phi$  fields, each with a different radius,  $r_x$ , and  $r_\phi$ . These design fields are filtered for different reasons. For the SIMP method, the usage of a filter is a known necessity. It is needed to prevent checkerboarding and it reduces mesh dependency by filtering out small artifacts. These features are not as relevant for the level-set method. In that case, the density filter helps convergence by smoothing out the transitions between solid and void elements. This smoothing ensures the existence of a boundary where significant sensitivities are present. This is necessary because the Heaviside function strongly reduces these sensitivities. Filtering also plays a role in controlling the thickness and convexity of the shell, this will be explained in Section 4.9.

## 4.8. The optimizer

The chosen optimizer for this project is the method of moving asymptotes (MMA). The MMA has first been proposed by Svanberg, 1987 for use in structural optimization, and it has become a very popular optimizer in topology optimization (Sigmund and Maute, 2013). This optimizer has been chosen because it is more versatile than the commonly used optimality criteria (OC) method from Bendsøe and Kikuchi, 1988. Unlike the OC method, the MMA is able to handle multiple constraints as-is. The main downside is that the MMA generally uses slightly more computational time than the OC method when solving a single constraint minimum compliance problem. However, the OC method uses a loop to satisfy the volume constraint. Because of

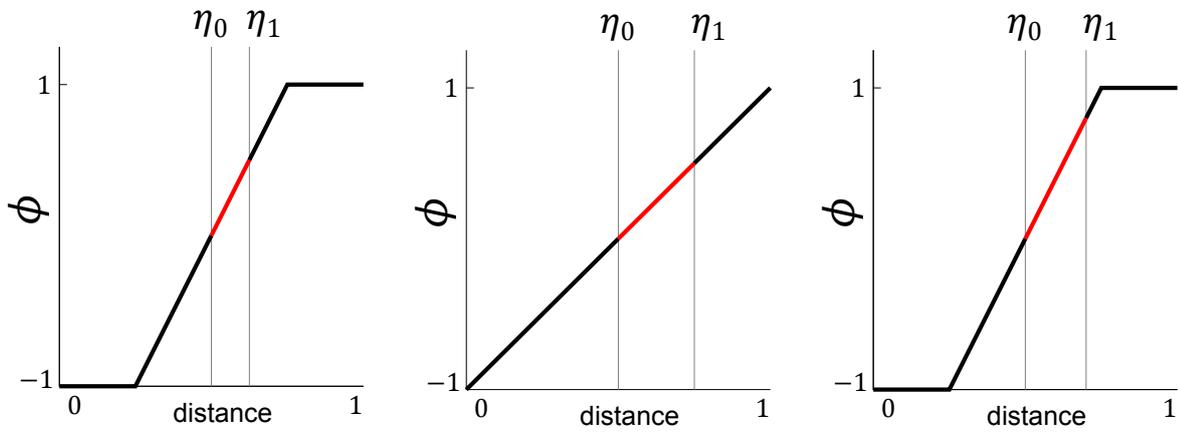
the complexity of the hybrid method this loop would be more expensive than usual, making the MMA the preferred choice. The implementation and parameters for the MMA optimizer are taken from Andreasen et al., 2020. The  $c$  parameter, for the penalization of the volume constraint, is set to the default value.

## 4.9. Controlling the shell

It can be necessary to exert a degree of control on the shell. The reason is that the structure might not meet the requirements for cleanability. There is no way for the algorithm to check whether the final design meets the cleanability requirements. Manually verifying the outcome is necessary. This means that the user ought to be able to change the inputs to achieve a satisfactory result when needed. It is also possible that a design that meets the requirements outlined in Chapter 2 might not be the best solution for the user. There could even be a different set of requirements. For that reason, parameters can easily be changed to control the thickness and shape of the shell.

The minimum shell thickness is affected by two parameters. It is not possible to set a maximum shell thickness. The first parameter is the filter radius of the density filter used on the level-set function  $\phi$ . This radius,  $r_\phi$ , will affect the maximum slope the level-set function can have. This reduction in slope results in a higher minimum of intermediate density values of  $\phi$  between -1 and 1. This higher minimum results, in turn, in more nodes having a value that is increased by the Heaviside function of the shell. This effect is visible when comparing the shell width in Figure 4.9a to the one in Figure 4.9b. The hard borders between shell and no shell shown in these figures are a simplification. There is a group of elements that have a value in both the  $\mathbf{s}$  and the  $\mathbf{s}_{shell}$  fields, as explained in Section 4.4. The second parameter is the cut-off of the Heaviside functions when mapping the level-set function,  $\eta_1$ . Increasing the value results in more material placed in  $\mathbf{s}_{shell}$  instead of  $\mathbf{s}$ . This can be seen by comparing Figure 4.9a to Figure 4.9c. It is also possible to achieve nearly the same effect by moving  $\eta_0$ . However, for the sake of simplicity, the value of  $\eta_0$  is always 0 for this project. The parameter  $\beta$  should also be mentioned here. Its value dictates the steepness of the Heaviside function. More information on  $\beta$  and the Heaviside function can be found in Appendix B. If the value of  $\eta_1$  is too low, the peak of the Heaviside function for the shell will be lower than 1. Thus, making it impossible to get fully solid shell elements. In that case, it is possible to increase  $\beta$  to create a steeper ascent in the function.

The second aspect of the shell that can be controlled is the curvature. The curvature is affected by the density filter radius on the level-set field,  $r_\phi$ . This is visualized in Figure 4.10. Due to the increased minimum distance between the fully solid material and the shell the corners cannot be as tight. The convexity will not necessarily be limited in areas where there is no need to place material close to the shell. Increasing the filter radius also impacts the thickness of the shell as explained earlier.



(a) A visualisation of the effect of  $\eta$  on the mapping function.

(b) The impact of the filter radius on the shell thickness. The more gradual slope results in a thicker shell.

(c) The impact of  $\eta_1$  on the shell thickness. A higher value for  $\eta_1$  means a thicker shell.

Figure 4.9: Comparison of the effect that  $\eta$  and the filter radius have on the thickness of the shell. The distance between  $\eta_0$  and  $\eta_1$  corresponds to the thickness of the shell. The distance values are arbitrary but identical for each figure. In most cases, the distance would span several elements.

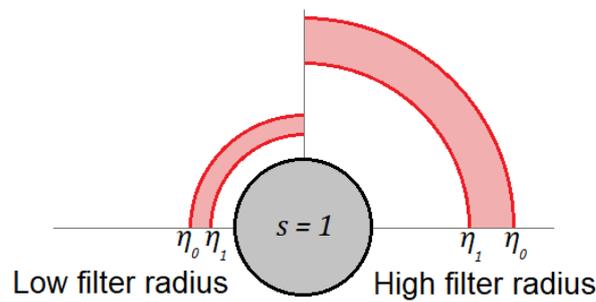


Figure 4.10: The impact of the filter radius on the width of the shell and the convexity. The circle in the center shows an area where the  $s$  field is 1. The shell field is shown in red, bounded between  $\eta_0$  and  $\eta_1$ . The low filter radius on the left allows for much sharper corners than the high filter radius on the right.



# 5

## Experiments

In this chapter, the results of multiple different experiments are presented. These experiments focus on the effect of the method on the final result. They have been performed on minimum compliance problems. Different sets of boundary conditions have been created to test the method. These sets are discussed in Section 5.1. In Section 5.2, the hybrid method is compared to the level-set and SIMP methods. The study into the behavior of the shell and how different parameters impact it is found in Section 5.3. Some more experiments have been performed to see if the method can successfully be applied to other boundary conditions. The results of these experiments are shown in Section 5.4. It is possible to create a design using a traditional method and add a shell afterward. The results of a comparison between this method and the shell method are shown in Section 5.5.

### 5.1. Boundary conditions

It is customary to demonstrate a topology optimization method using the boundary conditions for an MBB beam or a cantilever beam. However, these setups are not ideal for testing the behavior of the shell method. All sets of boundary conditions have the placement of fixed nodes and forces in the center of the design domain in common. This setup ensures that a shell can be freely formed from every side of the design. All fixed nodes have 0 degrees of freedom. A numbering system is used for the nodes to indicate the locations where forces are applied or nodes are fixed. This system is visualized in Figure 5.1. The numbering starts at 1 at the top left, this corresponds to the index numbers MATLAB uses for matrices. The coordinates start at 0 at the top left, the positive y-direction is downwards, and the positive x-direction is to the right. Two sets of boundary conditions have been used to extensively test the program. Both setups result in a design that should change by adding a shell. In this section, these setups and the expectations are briefly discussed.

#### 5.1.1. Four points setup

The first setup is the ‘four points’ setup. The design domain and boundary conditions can be seen in Figure 5.2a. The dots indicate points that are fixed in both directions, the arrows indicate forces. The exact locations of the nodes can be found in Table 5.1. The exact location, direction, and magnitude of the forces can be found in Table 5.2. The result of these boundary conditions can be seen in Figure 5.2b. This result is produced using the modified SIMP method. The fixed nodes and forces are indicated by the crosses and the arrows, respectively. The fixed points and the force on the top right are placed further from the center to ensure that the optimal design consists of two groups of solid elements. An additional shell is expected

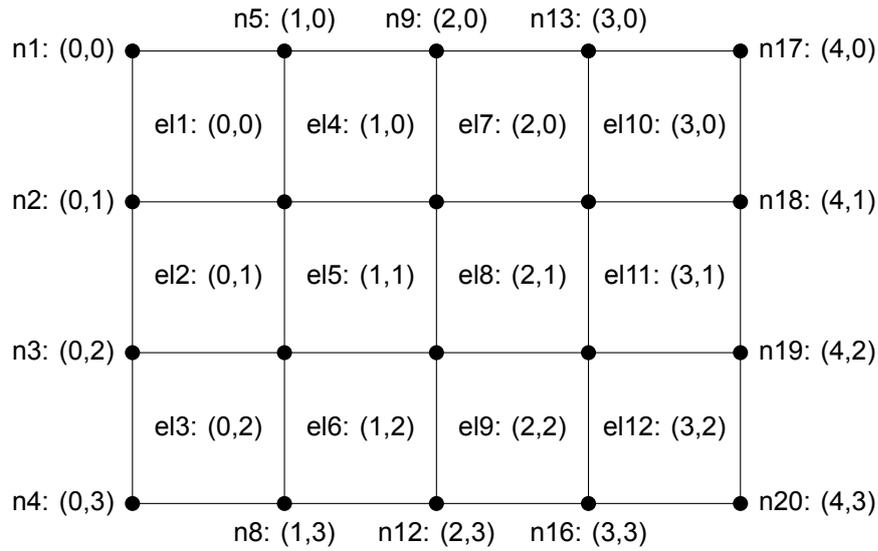


Figure 5.1: The numbering and coordinates of nodes and elements. The prefix 'n' denotes a node number, and 'el' denotes an element number. The numbering of the nodes and elements starts at the top left corner.

node number	coordinates (x,y)
813	12,32
828	12,47
1011	15,35
1026	15,50
1803	27,47
2001	30,50
3036	47,12
3266	50,15

Table 5.1: Node numbers and coordinates of the fixed nodes for the four points setup for a 64x64 design domain.

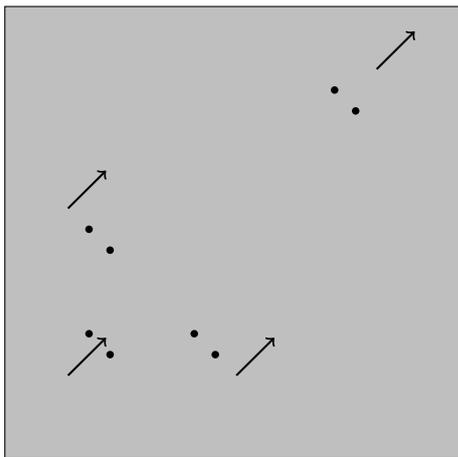
to connect these two groups, resulting in a single group of material elements. There is no requirement that a design must consist of a single unbroken area of the material. However, this is an expected result of the shell method. The expectation is also that a connection between different groups of material will often result in a more simple and cleanable design.

### 5.1.2. Iron setup

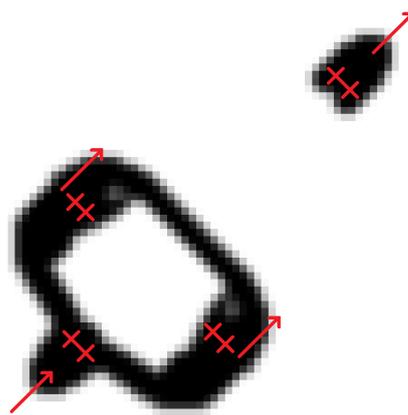
The second setup is the 'iron' setup. Named as such because the resulting image seen in Figure 5.3b somewhat resembles an iron. The design domain and boundary conditions can be seen in Figure 5.3a. The exact locations of the nodes can be found in Table 5.3. The exact location, direction, and magnitude of the forces can be found in Table 5.4. This design is nearly cleanable. Only the corner of the inset on the left is a bit too sharp and does not meet the requirements. The cleanability can be improved by reducing the number of inward-facing corners, as mentioned in Chapter 2. This design is well suited to test the control over the convexity of the shell.

## 5.2. Comparing the methods

In this section, comparisons are drawn between the optimization of the SIMP, level-set, and hybrid methods. The hybrid method is tested both with and without the shell field. The tests are



(a) The design domain of the four points setup. The fixed nodes are indicated by dots. These nodes are fixed in both directions. The forces are indicated by arrows and applied to the nodes located at the origin of the arrow.

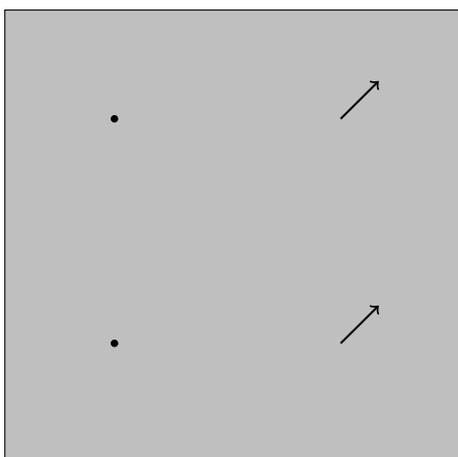


(b) The four points setup design including the boundary conditions. The design is created using the SIMP method. The fixed nodes are indicated by red crosses, the arrows indicate the forces.

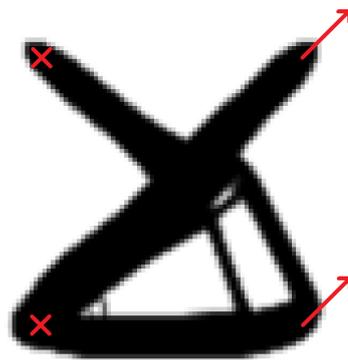
Figure 5.2: The boundary conditions of the four points setup.

node number	coordinates (x,y)	direction	magnitude
615	9,29	(+x,-y)	$\sqrt{2}$
639	9,53	(+x,-y)	$\sqrt{2}$
2199	33,53	(+x,-y)	$\sqrt{2}$
3455	53,9	(+x,-y)	$\sqrt{2}$

Table 5.2: Node numbers, coordinates, direction, and magnitude of the forces for the four points setup for a 64x64 design domain.



(a) The design domain of the iron setup. The fixed nodes are indicated by dots. These nodes are fixed in both directions. The forces are indicated by arrows and applied to the nodes located at the origin of the arrow.



(b) The iron setup design including the boundary conditions. The design is created using the SIMP method. The fixed nodes are indicated by red crosses, the arrows indicate the forces.

Figure 5.3: The boundary conditions of the iron setup.

node number	coordinates (x,y)
4031	31,95
4095	31,31

Table 5.3: Node numbers and coordinates of the fixed nodes for the iron setup for a 128x128 design domain.

node number	coordinates (x,y)	direction	magnitude
12287	95,31	(+x,-y)	$\sqrt{2}$
12351	95,95	(+x,-y)	$\sqrt{2}$

Table 5.4: Node numbers, coordinates, direction, and magnitude of the forces for the iron setup for a 128x128 design domain.

performed by generating the solutions to the boundary conditions mentioned in Section 5.3.1. The parameters used to perform each test in this section are identical. These parameters will be further explained and tested in Section 5.3. Their values can be found in Table 5.7. The initial level-set function for each setup is a square that is slightly larger than the final design. The shell field converges quite slowly. Therefore the change in the objective or design variables must be minimal to consider the problem converged. The stopping criteria for all experiments are an objective smaller than 0.01, or a maximum change of a single element in any field smaller than 0.003.

### 5.2.1. Four points setup

First is the four points setup. The results can be seen in Figure 5.4, the objective values are found in Table 5.5. The SIMP approach, Figure 5.4a, finds a slightly better solution than the level-set, Figure 5.4b, method with an objective value of 18.56 as opposed to 20.07. The level-set method might perform better when using a different initial design. The hybrid method, Figure 5.4c, finds a nearly identical solution to the one found by the SIMP approach. The hybrid method achieves a slightly lower objective value, 18.19 as opposed to 18.56. This slight increase in performance is most likely due to its sharper edges. The addition of the shell field results in the solution seen in Figure 5.4d. This addition results in the predicted behavior. Both areas of material are connected via the shell. This connection comes at the expense of performance regarding the objective. The objective is 20.18 as opposed to 18.19 of the hybrid method. In Figure 5.5, the individual fields of the shell method are shown. It is not visible in the final result, but the shell encloses the entire structure. It is merged with the other material where possible, effectively replacing the combination of density and level-set material. This behavior is needed to ensure the shell is not placed in locations where it serves no purpose.

Method and corresponding objective value	
SIMP	18.56
Level-set	20.07
Hybrid (no shell)	18.19
Hybrid	20.18

Table 5.5: The objective values of different methods for the four points setup

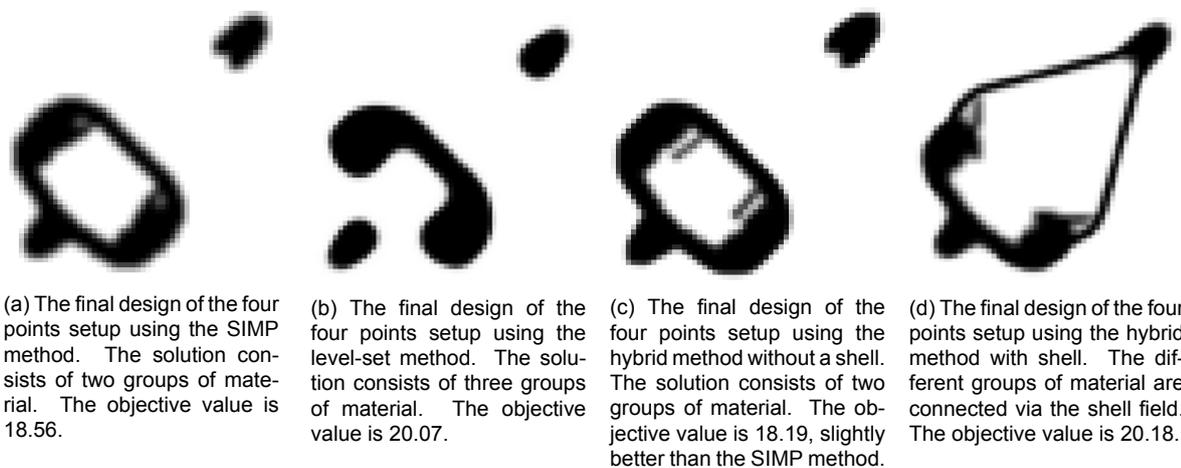


Figure 5.4: The final design of the four points setup using the different methods.

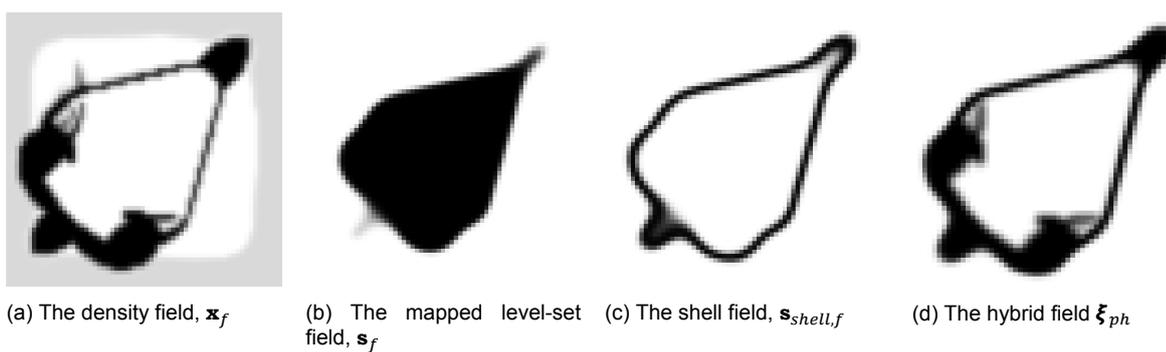


Figure 5.5: The different fields which compose the final result.

Method and corresponding objective value

SIMP	68.55
Level-set	67.42
Hybrid (no shell)	67.50
Hybrid	67.61

Table 5.6: The objective values of different methods for the iron setup



(a) The final design of the iron setup using the SIMP method. The objective value is 68.55.

(b) The final design of the iron setup using the level-set method. The objective value is 67.42.

(c) The final design of the iron setup using the SIMP method. The objective value is 67.50.

(d) The final design of the iron setup using the level-set method. The objective value is 67.61.

Figure 5.6: The final design of the iron setup using the different methods.

### 5.2.2. Iron setup

The second series of tests have been performed on the iron setup. The results of this setup can be seen in Figure 5.6, the objective values are found in Table 5.6. Both the SIMP and the level-set method, Figures 5.6a and 5.6b, end up with similar solutions. The level-set method is less detailed but achieves a slightly better objective value of 67.42 compared to 68.55. The hybrid method, Figure 5.6c, again finds a nearly identical solution to the SIMP approach. It achieves a slightly lower objective value of 67.50, which is very close to the level-set value. The addition of the shell field results in the solution seen in Figure 5.6d. The result hardly differs from the results of the other methods. This result is due to the shell closely wrapping itself around the structure. It replaces the material placed with the density method. This result demonstrates that the additional shell does not always lead to a different solution. The final design does not meet the requirements because the inwards-facing corner on the left is unchanged.

### 5.2.3. Convergence

The convergence of each method differs significantly. The convergence has been tested using different setups but only the four points setup will be discussed as it has been found that there are only minimal differences in the convergence between different setups. The SIMP method converges quickly and steadily, as can be seen in Figure 5.7. Full convergence is achieved after 70 iterations. The level-set method seen in Figure 5.8 is faster still and fully converges after 40 iterations. It can be seen that the level-set undershoots the volume constraint at the start. This undershoot is caused by the high maximum step size that the MMA can take for the level-set method. More information on this step size can be found in Appendix D. The reason this step size is so high can be found in the convergence of the shell method in Figure 5.10. The shell method needs 150 iterations to fully converge, much more than the other methods. This increase is due to the slow convergence of the shell. The convergence of the interior is not slower than it is for the hybrid method, seen in Figure 5.9. However, the shell itself needs

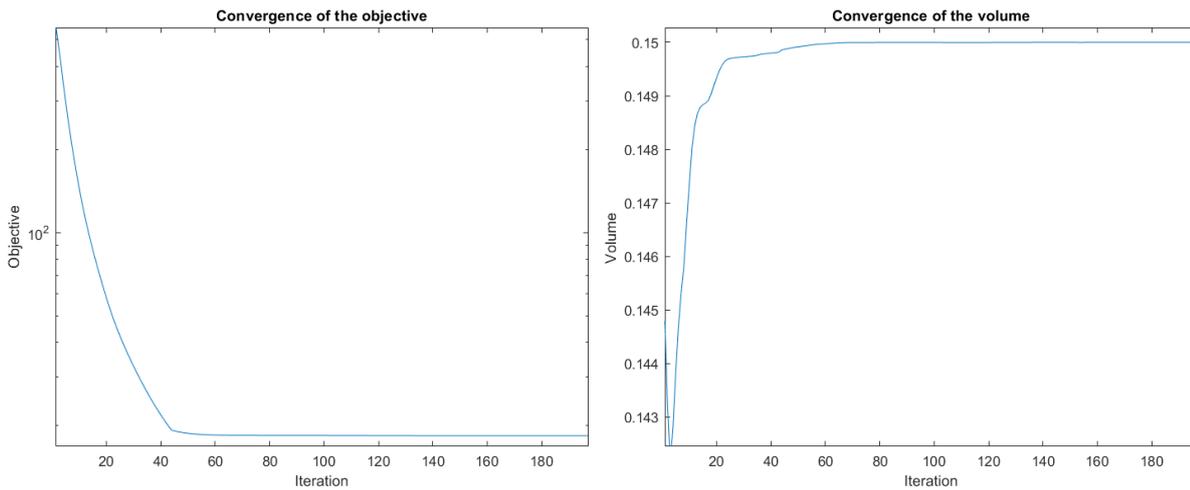


Figure 5.7: Convergence of the SIMP method for the four points setup.

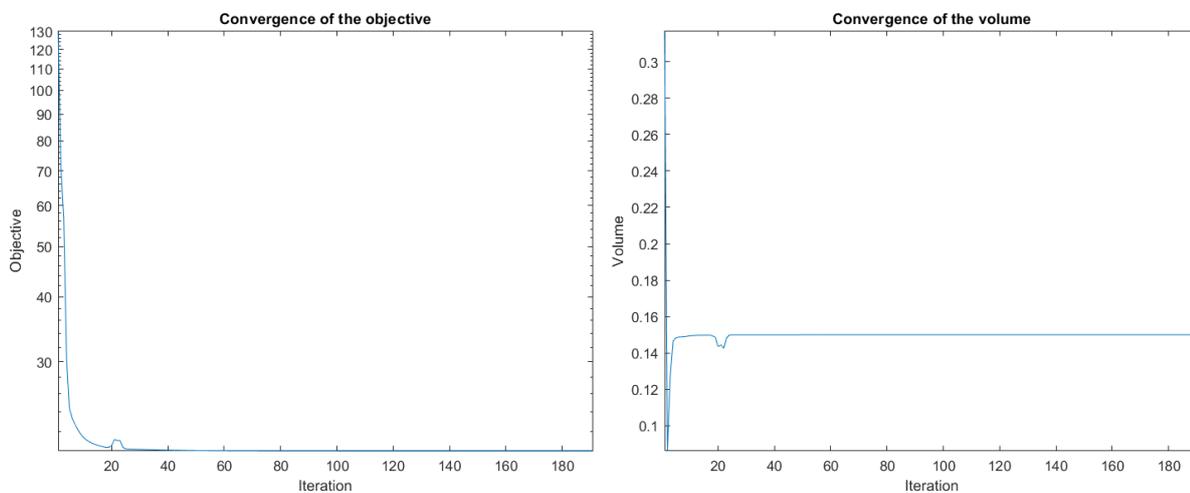


Figure 5.8: Convergence of the level-set method for the four points setup.

over 100 iterations on top of this. Taking smaller steps would further increase the number of required iterations. The overshoot can be seen during the shell method too. The shell shrinks until it reaches the rest of the material and then overshoots. This results in the small dips that can be seen around iterations 35, 80, and 125.

## 5.3. Controlling the shell

The program gives users control over several parameters. The impact of these parameters will be discussed in this section using the setups discussed in Section 5.1. Most parameters have little effect on the result or an impact that has already been researched in other works. For these parameters, a default value has been chosen. They are discussed in Subsection 5.3.1. The most interesting parameters are the ones impacting the behavior of the shell,  $r_\phi$  and  $\eta_1$ . The impact of these parameters has been tested. The results of these experiments are discussed in Subsection 5.3.2.

### 5.3.1. Default values

The default values for all the parameters are shown in Table 5.7. These are the values with which the experiments are performed unless otherwise mentioned. The mesh sizes are dic-

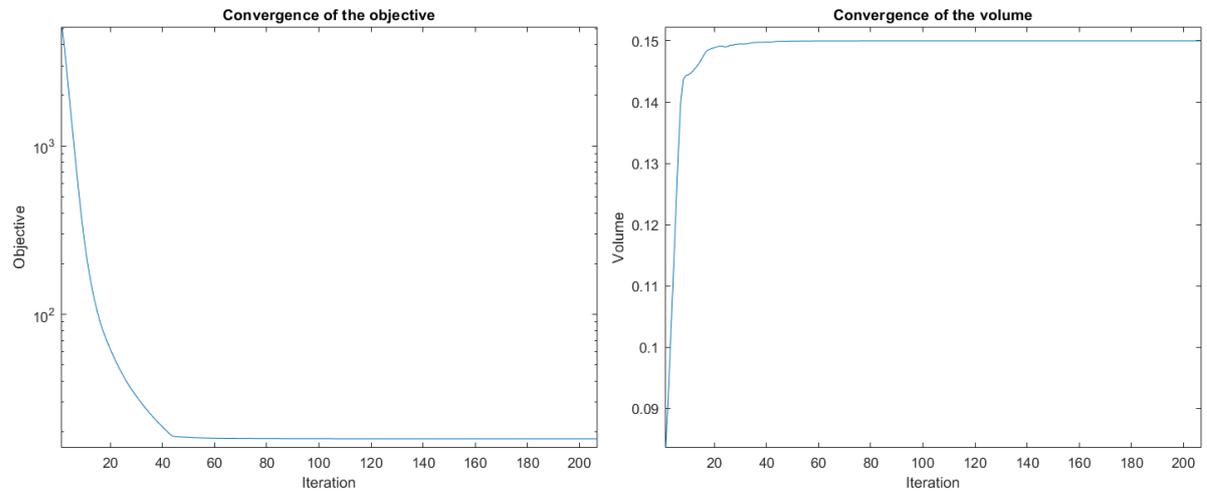


Figure 5.9: Convergence of the hybrid method for the four points setup.

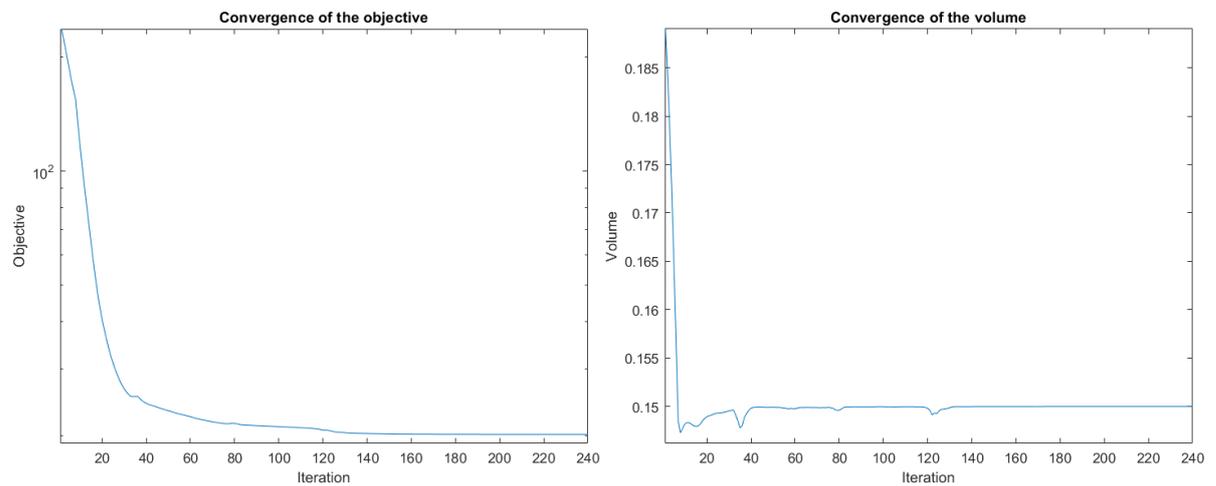


Figure 5.10: Convergence of the shell method for the four points setup.

tated by the limited computational power available. The chosen sizes, 64x64 and 128x128, offer a good balance between the computational time and the level of detail. The volume fraction of 0.15 has been chosen to encourage the creation of void space. This space allows for better insights into the behavior of the shell. A SIMP penalty of 3 is often used, strongly discouraging intermediate densities without causing further problems. This value will also be used here. As mentioned in the previous chapter,  $\eta_0$  is set to 0 to limit the number of options. The impact of this parameter is too similar to  $\eta_1$  to justify further testing. Information about the other parameters can be found in Appendix D.

Mesh size four points setup	64x64
Mesh size iron setup	128x128
Volume fraction	0.15
SIMP Penalty, $p$	3
Filter radius $\mathbf{x}$ , $r_x$	1.5
Filter radius $\phi$ , $r_\phi$	8
Number of mapping sampling points, $n^2$	49
Heaviside steepness $\beta$	12
Heaviside threshold $\eta_0$	0
Heaviside threshold $\eta_1$	0.4

Table 5.7: The default values for the parameters. All experiments use these values unless otherwise mentioned.

### 5.3.2. Shell control parameters

Two parameters have a large impact on the shape of the shell. The Heaviside cutoff,  $\eta_1$ , and the density filter radius on the level-set field,  $r_\phi$ . The way these parameters influence the shell thickness is explained in Section 4.9. It can be expected that increasing  $\eta_1$  will increase the thickness of the shell. Increasing  $r_\phi$  will increase the thickness of the shell while increasing the number of intermediate densities. It will also increase the minimum convexity of the shell. The parameters have been tested separately and together to see how they interact. The stopping criteria for all experiments are an objective smaller than 0.01, or a maximum change of a single element in any field smaller than 0.003.

#### The impact of $\eta_1$

The parameter investigation of  $\eta_1$  has been performed with a filter radius,  $r_\phi$ , of 8. This value allows for relatively fast convergence. The increase in shell thickness when  $\eta_1$  is increased is visible. It is shown in Figure 5.11a that the shell is thin and consists of nothing but intermediate densities when  $\eta_1$  equals 0.1. The impact on the objective compared to the hybrid method without a shell is also small, an increase of 0.32. The shell features more solid elements when  $\eta_1$  is increased to 0.3. This can be seen in Figure 5.11b. This solution looks easily cleanable with a relatively small impact on the objective, an increase of 1.01. Increasing  $\eta_1$  to 0.4 a very different solution appears, as shown in Figure 5.11c. The material cost of the shell means less material can be used in other locations. This change is paired with a relatively large increase in objective compared to the  $0.3\eta_1$  design, 20.18 instead of 19.20. This trend continues, with a value of 0.7 leading to a solution that is little more than just a shell, Figure 5.11d. This  $\eta_1$  comes with an objective of 21.86, the worst performance so far. The iron setup is minimally affected by changes of  $\eta_1$  because the shell merges with the rest of the structure. For that reason, these results are not shown here. It is worth mentioning that during the optimization, the same changes in the shell observed for the four points setup can be seen in the iron setup.

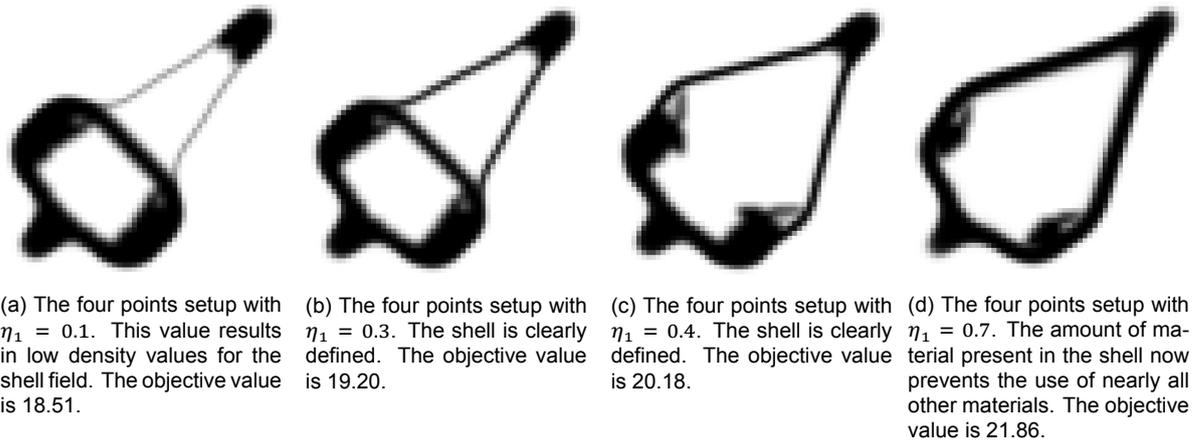


Figure 5.11: The comparison between different values of  $\eta_1$  in the four-point setup.

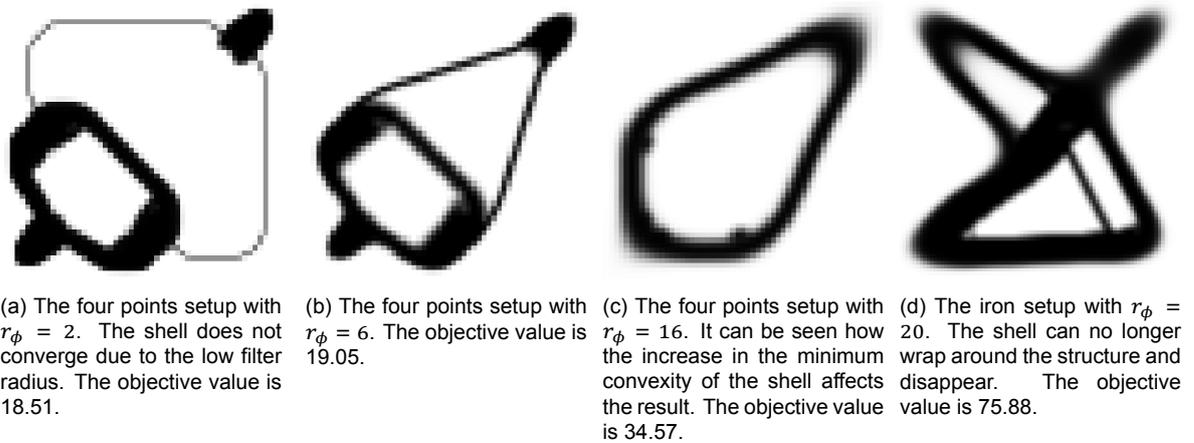


Figure 5.12: The comparison between different values of  $r_\phi$ .

### The impact of $r_\phi$

The parameter investigation of  $r_\phi$  has been performed with an  $\eta_1$  of 0.4. It can be seen in Figure 5.12a that a filter radius of 2 results in a thin shell. The stopping criteria are met with very little movement of the shell. The values of the sensitivities for the level-set function are too low. The filter radius is required to be at least 4 for the shell to converge at all. The difference in the amount of material used in the shell leads to a different topology. This can be observed when comparing Figure 5.12b to Figure 5.11c. This change is consistent with the change seen with an increase of  $\eta_1$ . The increase in the thickness of the shell is visible when comparing the figures. The increase in the minimum convexity is not. This increase becomes visible when the filter radius is increased to 16. The result is shown in Figure 5.12c. The shell no longer fits the sharp corner on the top right. For the iron setup, the results are largely the same, a low filter radius results in a lack of convergence. A high filter radius is interesting because it can ensure the shell can no longer completely merge with the density material. A filter radius of 20 achieves this, as can be seen in Figure 5.12d. Due to its thickness, the shell takes over a significant part of the construction.

### Combining $\eta_1$ and $r_\phi$

An interesting interaction occurs when a high filter radius value is combined with a low value for  $\eta_1$ . This combination results in a more rounded shell without increasing the thickness too much. The results of using these parameters for the four points setup can be seen in Figure



(a) The four points setup with  $r_\phi = 16$  and  $\eta_1 = 0.15$ . The limited convexity of the shell is seen at the top right and bottom left. The objective value is 21.92.

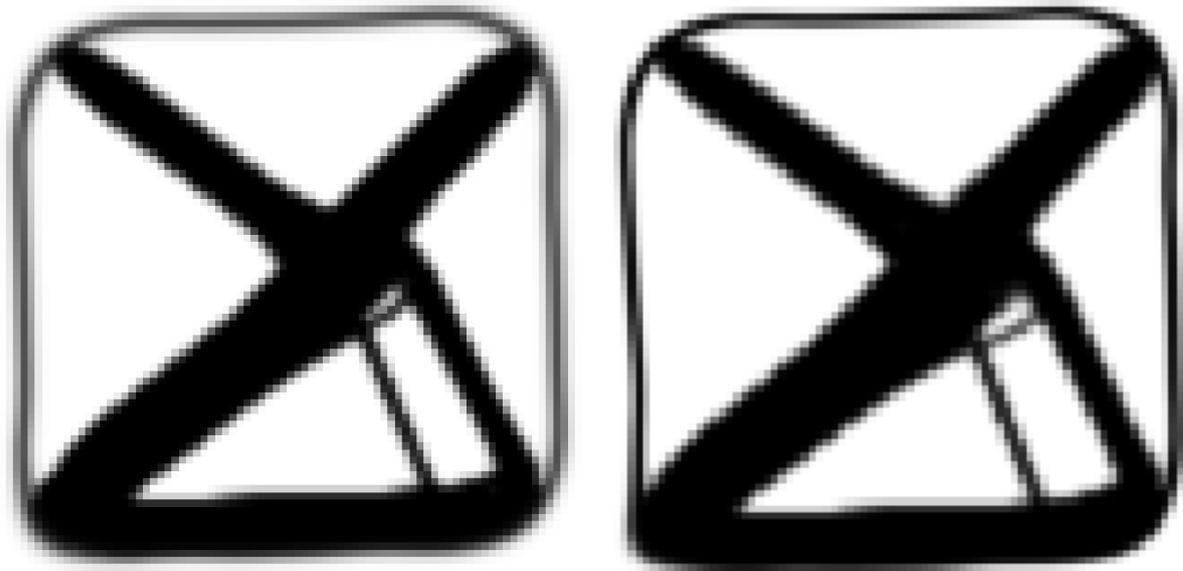
(b) The four points setup with  $r_\phi = 16$  and  $\eta_1 = 0.15$  and  $\beta = 24$ . The increase of  $\beta$  from 12 to 24 results in what is again a sharper and more solid shell. The objective value is 21.14.

Figure 5.13: The result of combining a high value for  $r_\phi$  with a low value for  $\eta_1$  for the four points setup.

5.13. Combining a filter radius of 16 with an  $\eta_1$  of 0.15 results in Figure 5.13a. The shell takes on a different, more convex, shape. The low value of  $\eta_1$  does result in shell elements that are not fully solid. The densities in the shell can be increased by picking a higher value for  $\beta$ . This can be seen in 5.13b. Increasing  $\beta$  also significantly decreases the objective value. The results for the iron setup can be seen in Figure 5.14. Here, using a  $r_\phi$  of 20 with an  $\eta_1$  of 0.15 results in a rectangular shell around the structure as can be seen in Figure 5.14a. This combination allows the iron setup to meet the cleanability requirements. This shell has a thickness of several elements, but its densities are not close to 1. Increasing the  $\beta$  results in a more solid shell in this case too as can be seen in Figure 5.14b. This increase in  $\beta$  has little effect on the results in these cases. However, it should be noted that when the filter needs to be increased even further, new problems could arise. One example is the convergence problem described in Appendix D.

## 5.4. Shell creation using different boundary conditions

In the previous section, the behavior of the shell method has been demonstrated using two setups. In this section, the impact of different combinations of parameters on a design is demonstrated. The four sets of parameter combinations can be found in Table 5.8. These combinations result in a very similar minimal shell thickness, but the minimum curvature differs. This represents a scenario in which the shell thickness is known, and a suitable shape needs to be found. The goal of these experiments is to demonstrate how difficult it may be to get the desired result. Four more sets of boundary conditions are used to test these parameter combinations. The setups seen in Figures 5.15 and 5.16 already meet the requirements. However, the number of inward-facing corners can be reduced. The setup in Figure 5.17 does not meet the requirements due to an acute corner near the top right. The last setup, Figure 5.18, consists of two areas of material. The tables containing the relevant nodes can be found in Appendix E. The initial level-set function for each setup is a square that is slightly larger than



(a) The iron setup with  $r_\phi = 20$  and  $\eta_1 = 0.15$ . The shell now encloses the structure without becoming part of it. The objective value is 77.09.

(b) The iron setup with  $r_\phi = 20$  and  $\eta_1 = 0.15$  and  $\beta = 30$ . The increase of  $\beta$  from 12 to 30 results in what is again a sharper and more solid shell. The objective value is 76.03.

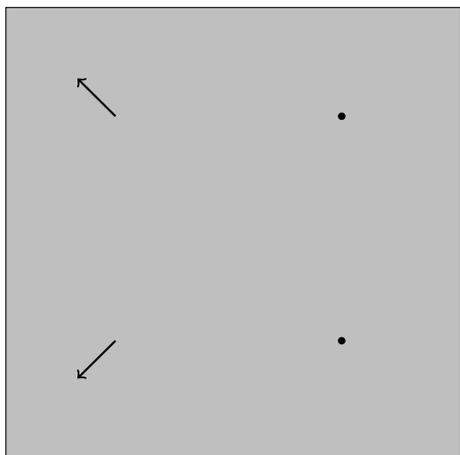
Figure 5.14: The result of combining a high value for  $r_\phi$  with a low value for  $\eta_1$  for the iron setup.

the final design. A mesh of 128x128 has been used for every setup. The other parameters are identical to the ones found in Table 5.7. The stopping criteria for all experiments are an objective smaller than 0.01, or a maximum change of a single element in any field smaller than 0.003.

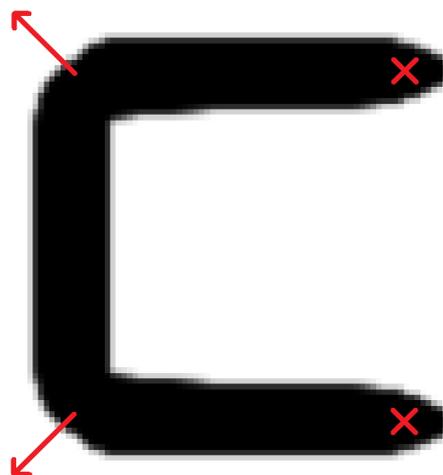
Combination	$r_\phi$	$\eta_1$	$\beta$
1	8	0.4	12
2	12	0.274	18
3	16	0.208	24
4	20	0.167	30

Table 5.8: The four different combinations of variables.

The first setup is the C setup, the resulting designs can be seen in Figure 5.19. It can be seen that the results for each set of parameters are very similar. Each solution is cleanable. The increase in filter radius does lead to a slight increase in the objective value, from 21.28 to 21.37. This is a small increase compared to the objective value without a shell, which is 20.76. The X setup produces two distinct solutions, these can be seen in Figure 5.20. The first three parameter combinations yield a result that is very similar to the result produced by the SIMP method in 5.16. The final combination seen in Figure 5.20d does feature a shell. This shell is relatively costly for the objective value, increasing it from 34.42 to 40.62. The results of the three points setup can be seen in Figure 5.21. The solutions are quite similar for each parameter combination, all can be considered cleanable too. The increase in curvature is visible for a filter radius of 20. The objective value does not change much compared to the uncleanable SIMP design. The value increases from 14.98 to 15.05 for the small filter radius and 15.13 for the large filter radius. The equals sign setup yields four nearly identical solutions, these can be seen in Figure 5.22. Both areas of material are connected. The objective value is slightly increased compared to the SIMP design. From 11.54 to 11.90 for

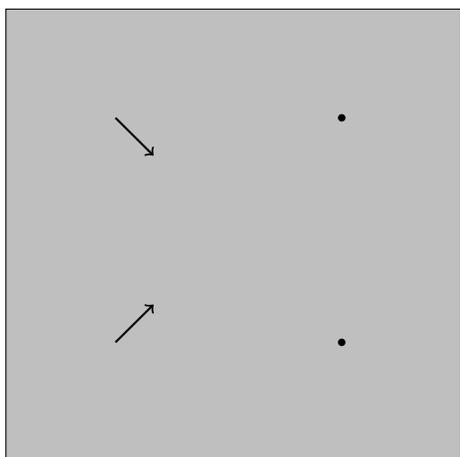


(a) The design domain of the c setup. The fixed nodes are indicated by dots. These nodes are fixed in both directions. The forces are indicated by arrows and applied to the nodes located at the origin of the arrow.

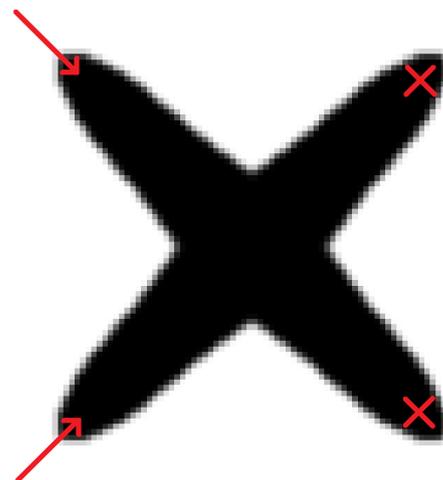


(b) The c setup, created using the SIMP method. The objective value is 20.76.

Figure 5.15: The boundary conditions of the c setup

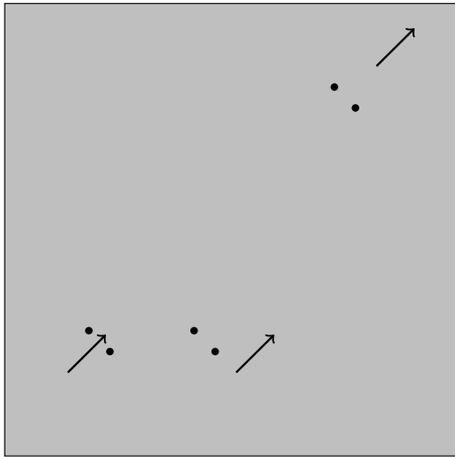


(a) The design domain of the x setup. The fixed nodes are indicated by dots. These nodes are fixed in both directions. The forces are indicated by arrows and applied to the nodes located at the origin of the arrow.

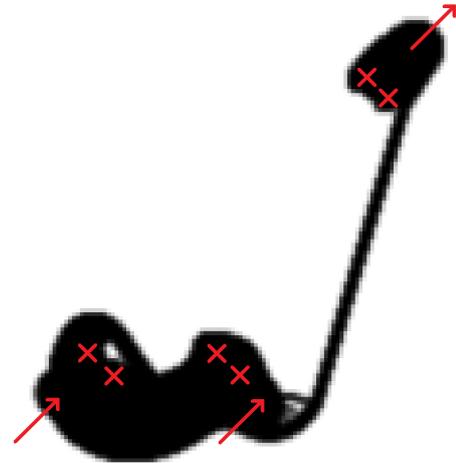


(b) The x setup, created using the SIMP method. The objective value is 34.42.

Figure 5.16: The boundary conditions of the x setup

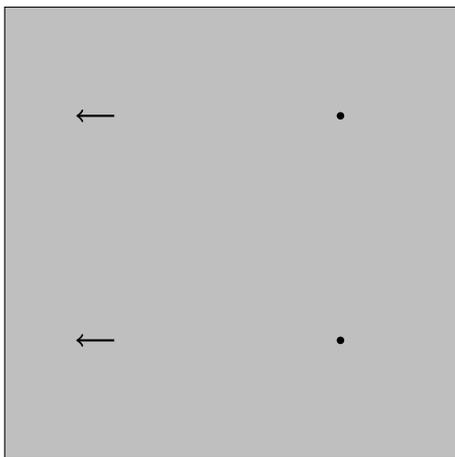


(a) The design domain of the three points setup. The fixed nodes are indicated by dots. These nodes are fixed in both directions. The forces are indicated by arrows and applied to the nodes located at the origin of the arrow.

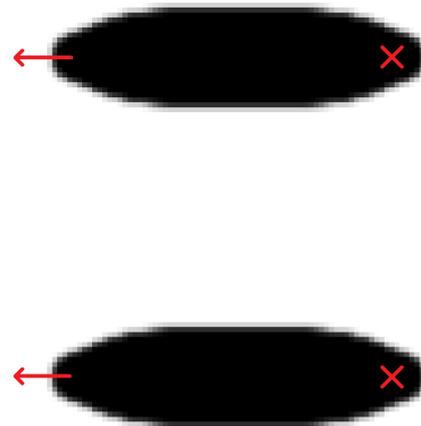


(b) The three point setup, created using the SIMP method. This result does not meet the cleanability requirements. The objective value is 14.98.

Figure 5.17: The boundary conditions of the three points setup

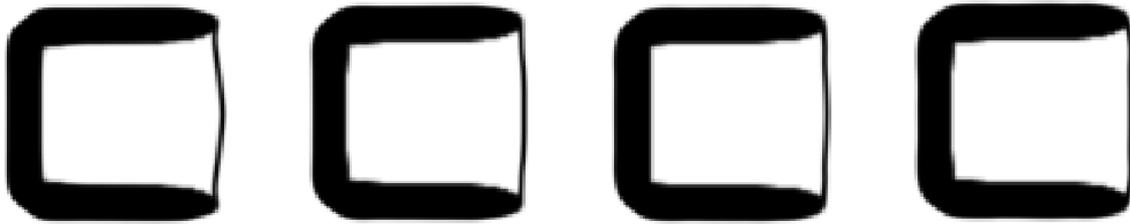


(a) The design domain of the equals sign setup. The fixed nodes are indicated by dots. These nodes are fixed in both directions. The forces are indicated by arrows and applied to the nodes located at the origin of the arrow.



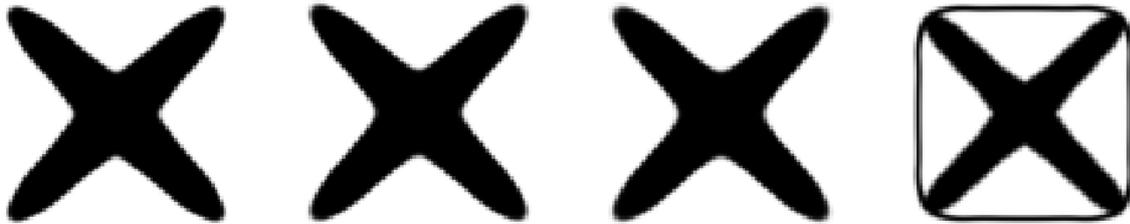
(b) The equals sign setup, created using the SIMP method. The objective value is 11.54.

Figure 5.18: The boundary conditions of the equals sign setup



(a) The C setup with parameter combination 1. The objective value is 21.28. (b) The C setup with parameter combination 2. The objective value is 21.32. (c) The C setup with parameter combination 3. The objective value is 21.34. (d) The C setup with parameter combination 4. The objective value is 21.37.

Figure 5.19



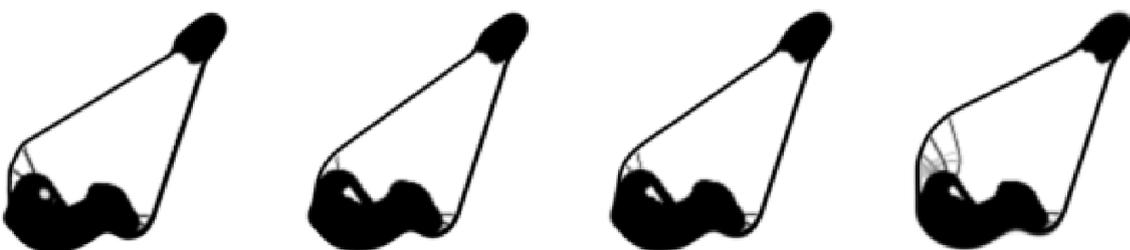
(a) The X setup with parameter combination 1. The objective value is 34.10. (b) The X setup with parameter combination 2. The objective value is 34.19. (c) The X setup with parameter combination 3. The objective value is 34.34. (d) The X setup with parameter combination 4. The objective value is 40.62.

Figure 5.20

parameter combination 1 and 11.96 for parameter combination 4. It can be seen that in most cases, the shell method finds a solution that improves cleanability. In nearly every case, the method also produces very similar results for different parameter combinations. The X setup is the exception, where a high filter radius is required to produce a different result. It can be concluded that in most cases the shell method will produce a desirable result for a lower filter radius and  $\beta$ . This offers a good starting point for an optimization process because it also results in the lowest objective value. If the result is not satisfactory, the filter radius can be increased until a cleanable design emerges.

### 5.5. Comparison to adding a shell

A simple alternative to this method could be to add a shell to an existing design. In this chapter, this two-step approach is compared to the hybrid method with a shell, which will be referred to as the shell method. The steps taken for the two-step method are illustrated in Figure 5.24 to



(a) The three point setup with parameter combination 1. The objective value is 15.05. (b) The three point setup with parameter combination 2. The objective value is 15.08. (c) The three point setup with parameter combination 3. The objective value is 15.08. (d) The three point setup with parameter combination 4. The objective value is 15.13.

Figure 5.21

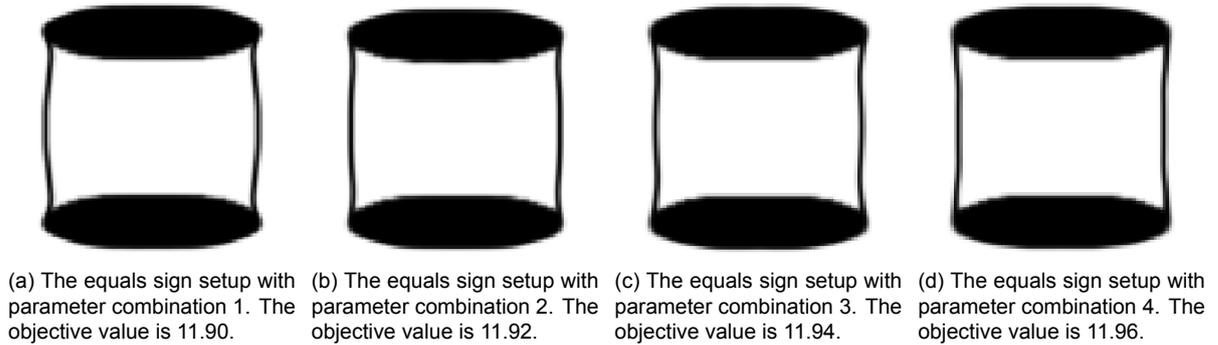


Figure 5.22

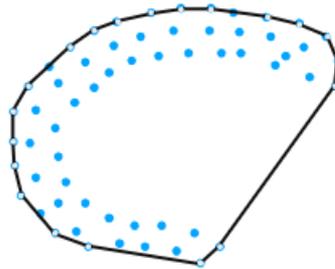


Figure 5.23: An example of a convex hull. From Moreira and Santos, 2007

help visualize the method. The first step in the two-step method is finding a solution for a lower volume fraction than the one used for the shell method, using the density or hybrid method without a shell. The remaining volume budget is then used for the shell itself. The convex hull is used to decide where the shell should be placed. A convex hull is the smallest convex shape that can be placed around a set of points. An example is shown in Figure 5.23. For the two-step method, the convex hull is placed around all the points with a density higher than 0.9. This value ensures that the shell will be directly connected to the structure. The densities of all the elements that make up the convex hull are then set to 1. Next, the parts of the convex hull that directly border the material of the original design are removed. This removal ensures that the convex hull is not added as an extra layer on top of the existing structure in locations where it is unneeded. After this removal, it is possible to increase the thickness of the hull. This increase may be desirable depending on the size of the mesh. Then, a density filter is applied to the hull. This filter makes the hull smoother and ensures there will be no small artifacts. Finally, the hull volume is adjusted for the design to meet the original constraint. The stopping criteria for all experiments are an objective smaller than 0.01, or a maximum change of a single element in any field smaller than 0.003.

The first experiment is performed using the four points setup. The initial volume constraint for the designs is 0.12. That means a volume fraction of 0.03 is reserved for the convex hull shell. The shell method uses roughly twice the material for the shell field. However, this shell field also takes over some of the material placed by the  $\mathbf{x}$  and  $\mathbf{s}$  fields. The volume budget of 0.03 for the convex hull shell results in an effective shell of roughly the same thickness. The hull has been created using a filter radius of 1.5 on a single-element thickness. The results are found in Figure 5.25. The density method with a convex hull shell has the highest objective value, 21.00. A better result is achieved using the hybrid method with a convex hull shell, with an objective value of 20.38. This increase in performance is consistent with the earlier results without the convex hull. The shell method has the best performance with an objective of 19.20. The designs that use the convex hull have some shell material on the bottom left

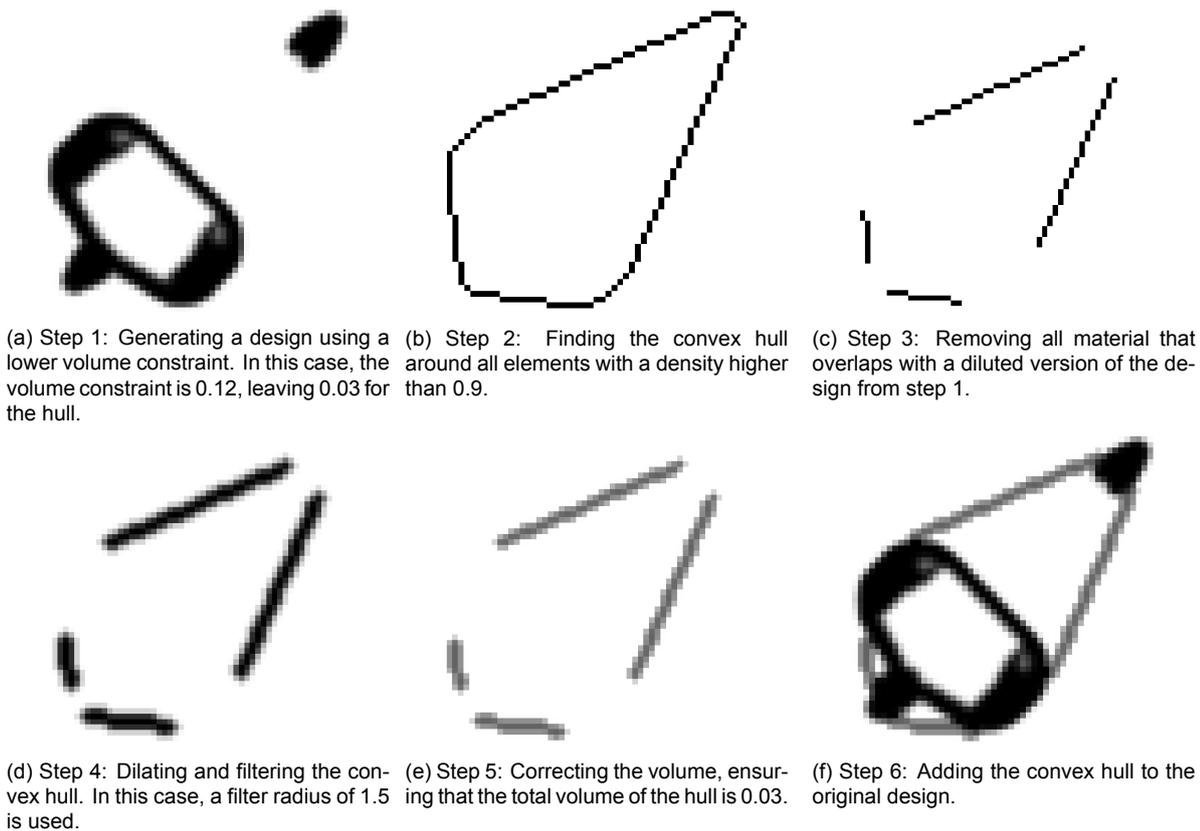


Figure 5.24: The six steps taken to add the convex hull shell to the design for the two-step approach.

that is not present in the hybrid method with a shell. The shell connection between the two areas of material is also slightly longer in the convex hull designs.

The second experiment is performed using the iron setup. The initial volume constraint for the designs is 0.13. That means a volume fraction of 0.02 is reserved for the convex hull shell. The shell method uses 0.028 volume for the shell fields. Of this volume, roughly a quarter is merged with the structure, so this should be a fair comparison. The hull has been created using a filter radius of 3 on a hull that is 3 elements thick. The results are found in Figure 5.26. The results are similar to the results of the four points setup. The density method with a convex hull shell has the highest objective value in this setup as well, 79.69. A better result is achieved using the hybrid method with a convex hull shell, with an objective value of 78.12. The shell method has the best performance with an objective of 77.09. The relative difference in the objective value between these designs is smaller than it was for the four points setup. Despite the filtering, the convex hull shell is not a completely straight line on the left side, small jumps are visible. The hybrid convex hull also places some material near the bottom. This material is most visible in the hybrid method design.

The results show that the hybrid method with a shell performs slightly better than adding the convex hull shell afterward. The implementation of the addition of this shell could be better. It can be improved by creating a smoother hull and removing it from more locations where it adds little to the objective. It is unlikely that this method will outperform the shell method even with these improvements. It is also more difficult to get a consistent thickness and density for the shell when using the convex hull method.



(a) The density method with added convex hull shell. The objective value is 21.00.

(b) The hybrid method with added convex hull shell. The objective value is 20.38.

(c) The hybrid method with shell. The objective value is 19.20.

Figure 5.25: Comparison between the different methods to create a shell using the four points setup. It can be seen that the shell method (c) performs significantly better than the density method (a) and the hybrid method with an added convex hull (b).



(a) The density method with added convex hull shell. The objective value is 79.81.

(b) The hybrid method with added convex hull shell. The objective value is 78.12.

(c) The hybrid method with shell. The objective value is 77.09.

Figure 5.26: Comparison between the different methods to create a shell using the iron setup. It can be seen that the hybrid method with shell (c) performs better than both the density method (a) and the hybrid method (b) with the added convex hull.

# 6

## Discussion

### 6.1. Definition of cleanability

In Chapter 2, some assumptions have been made concerning cleanability. A cleanable structure has been defined as a structure with an exterior of which every location is reachable. Moreover, no acute angles may be present on the exterior. These assumptions have been made based on very little research. A deeper dive into what makes a structure cleanable might generate new insights. Another consideration is that there are many ways to clean objects. Each method will have its requirements. The shell method deals with physical cleanability in a more general sense. Designing with a single, well-defined, cleaning method in mind will probably lead to a more applicable result.

### 6.2. The degree of control over the shell

Two parameters have a large impact on the way the shell behaves,  $\eta_1$  and  $r_\phi$ . Tweaking these values gives a reasonable degree of control over the shell. However, it would be an improvement to have more ways of controlling the shell. One good addition would be a way to control the curvature of the shell without impacting its other properties. This control is currently exerted by changing  $\eta_1$ ,  $r_\phi$ , and  $\beta$ . However, it has been shown that problems arise when using a low value for the filter radius or a high value for  $\beta$ . Because of this, different combinations of curvature and thickness cannot be used. The minimum convexity that is enforced using these parameters could also be achieved by controlling the feature size for the level-set function, as has been done by, for example, Guo et al., 2014. A different possibility might be to ensure that the interface of the level-set function is convex. This can prevent the behavior shown in Figure 3.1d without further controlling the curvature. A method to achieve this has been proposed by Shi and Li, 2021.

It is also possible to implement a minimum surface area of the level-set function. This can be done by using a mass conservative, or volume preserving, level-set function. Different versions of a mass conservative level-set function have been proposed by, for example, Kuzmin, 2014 and Basting and Kuzmin, 2014. This can also prevent the behavior shown in Figure 3.1d without controlling the curvature.

### 6.3. The use of local level-set functions

The level-set field  $\phi$  consists of local FEM basis functions. This method is easy to implement but any change only affects a single node. Therefore more iterations are needed to fully converge (Van Dijk et al., 2013). The convergence plots have shown that most of the time spent optimizing is dedicated to the shell. It would have been valuable to explore different

basis functions and compare the results. The use of a different type would likely reduce the number of iterations and save time in the long run.

# 7

## Conclusion

The main objective of this research is to create a topology optimization method that can generate two-dimensional optimized structures that meet the cleanability requirement. The results have shown that this objective has been achieved on several sets of boundary conditions. The shell method also performs better than adding a shell to an existing structure.

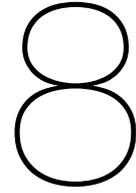
The concept of cleanability has been narrowed down to two requirements. The first is that all exterior surfaces of a structure must be visible. The second requirement is that no acute corners are present on this surface. Beyond that, reducing the number of inward-facing corners is assumed to improve cleanability, but this is not a requirement. Structures can be covered by a cleanable shell, resulting in a cleanable exterior. This shell must become part of the structure and the optimization process to ensure it contributes towards the objective.

A method is proposed to generate these shelled structures. This method uses a combination of the modified SIMP method and the level-set method, referred to in this project as the hybrid method. The level-set function acts as a boundary. The SIMP method can place material inside this boundary. A shell can be created by placing material near the edge of the level-set function. The nodal values of the level-set are mapped to the elements of the mesh to combine the two methods. This mapping is done via two separate fields,  $\mathbf{s}$  and  $\mathbf{s}_{shell}$ , by using different Heaviside functions. The  $\mathbf{s}$  field is always enclosed by the shell field, resulting in a shell around the structure.

This method has been tested for several boundary conditions, consisting of both cleanable and uncleanable designs. At least one cleanable solution has been found for each set of boundary conditions. The effect of the parameters on the thickness and convexity of the shell has been demonstrated. The shell method produces better results than adding a convex hull shell to an existing design.

The method is not guaranteed to produce a result that meets the requirements for cleanability as it cannot verify whether the final result is cleanable. The cleanability needs to be confirmed by the user. If the requirements have not been met, changing the parameters and trying again will likely yield a satisfactory result.





## Future work

### **8.1. Finding a way to quantify cleanability**

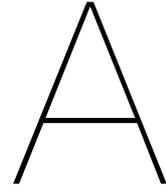
The main weakness of the proposed method is the lack of quantification of cleanability. The program is unaware of the secondary objective that has been implicitly added. It simply generates the structure with the least compliance. The method steers toward such a cleanable result, but there is no way to guarantee a cleanable structure as output. If cleanability were to be quantified it could be used as either an objective or a constraint. This would open up the possibility to make it part of the optimization process and would allow for cleanability to become an objective or constraint, and part of the optimization process. Only then would it be possible to guarantee cleanable structures or to easily weigh the additional cost of cleanability.

### **8.2. Extending to the 3rd dimension**

For this method to be applied in the industry, an extension to the third dimension is needed. With the addition of this dimension, the number of design variables and the computational time required would increase drastically. It would therefore be advisable to find ways to reduce this. It is advisable to use a faster programming language such as C++ to create this program.

There is little reason to think this method will not work in 3D but one problem that could arise is the inconsistency of the method. As mentioned, using this method does not always lead to a cleanable result. Different parameter settings are needed for different boundary conditions. This problem could be amplified in 3D. The increase in variables can make results less consistent and raise the need for more ways to control the shell.





# Density and level-set flowcharts

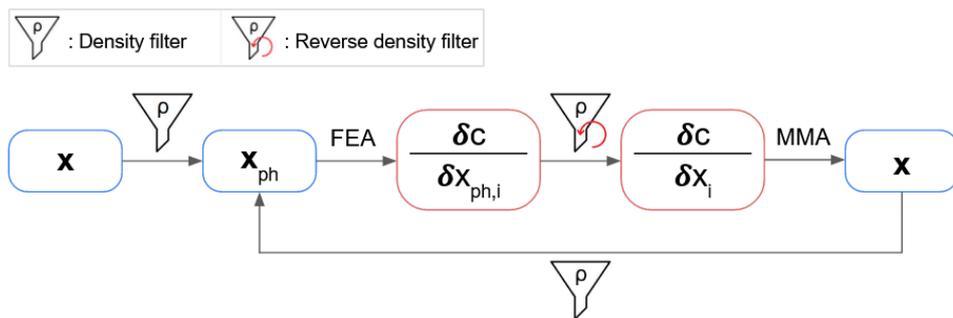


Figure A.1: Simplified flowchart of the steps taken for the density method.

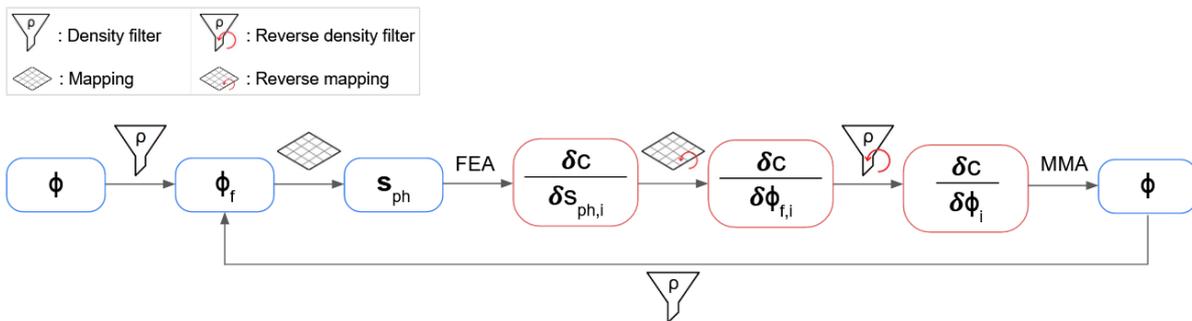


Figure A.2: Simplified flowchart of the steps taken for the level-set method.



# B

## The Heaviside function

The Heaviside function is inspired by the Heaviside step function. This is a discrete step function with a cutoff. With this function, any number between 1 and 0 is transformed into either 1 or 0. A continuous version of this function is first used in topology optimization by Guest et al., 2004. Many different versions have been used. Eventually, this function has been turned into the volume-preserving continuous equation by Wang et al., 2011 that can be seen in Equation B.1. This is the version used in this project.

$$\tilde{\rho}_i = \frac{\tanh \beta \eta + \tanh \beta (\rho_i - \eta)}{\tanh \beta \eta + \tanh \beta (1 - \eta)} \quad (\text{B.1})$$

The equation appears in Section 4.4 as the Equation which transforms  $\phi_L$  into  $\phi_H$ , also seen here as Equation B.2.

$$\phi_{H,i} = \frac{\tanh \beta + \tanh (\beta (\phi_{L,i} - \eta_1))}{\tanh \beta + \tanh (\beta (1 - \eta_1))} \quad (\text{B.2})$$

In this equation, the value of  $\phi_H$  is the result of three variables. The input  $\phi_L$ ,  $\eta_1$ , and  $\beta$ . The value of  $\eta_1$  shows where the threshold is. Values below this threshold will be rounded down, those above will be rounded up. The parameter  $\beta$  is used to control the steepness of the function. This is visualized in Figure B.1. It can be seen that a higher value of  $\beta$  results in a steeper function. An infinitely high  $\beta$  would result in a discrete step function.

The Heaviside function can be derived. This is necessary for the calculation of the sensitivities as explained in Section 4.6. The derived Heaviside function is shown as Equation B.3. This equation also appears in Section 4.4 as Equation 4.20.

$$\frac{\partial \phi_H}{\partial \phi_L} = \beta \frac{\text{sech}^2 (\beta (\phi_L - \eta_1))}{\tanh \beta + \tanh (\beta (1 - \eta_1))} \quad (\text{B.3})$$

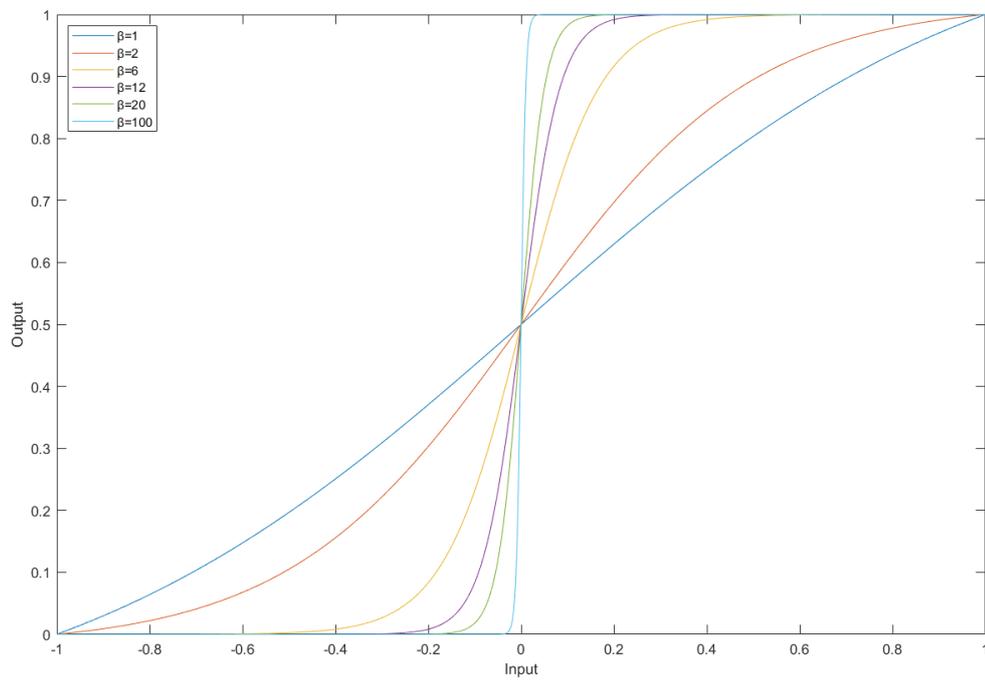
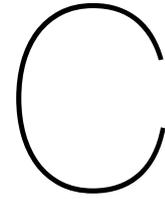


Figure B.1: The continuous Heaviside function for different values of  $\beta$ . The value of  $\eta_1$  is set to 0.



## Finite difference results

In this chapter, the finite difference (FD) results are presented. These results are created using a finite difference check. The FD check is performed to verify the objective and volume sensitivities. The goal of this verification is to ensure that the calculation of the sensitivities works as intended. The check is done by comparing the effect of a small perturbation of a single node or element to the calculated sensitivity. A separate check has been performed for the SIMP method, the level-set method, and the shell method. The four points setup from Section 5.1 has been used for these checks. The parameters are the default parameters from Table 5.7. For each check, six different elements or nodes have been selected. Two void points, one near the area of material on the bottom left, and one near the top right. Two intermediate points on the edge between the material and the void, one on the bottom left and one on the top right. And lastly, two solid points, one on the bottom left and one on the top right. The node numbers are displayed in the graphs, the explanation of these numbers can be found in Figure 5.1.

The results of the SIMP method FD check can be seen in Figure C.1. The error is displayed on the y-axis, and the magnitude of the perturbation is displayed on the x-axis. It can be seen that the objective error of all elements besides the void elements is small. The void elements are not shown because their error is 'infinite'. This is due to the FD check finding a sensitivity of  $10^{12}$  or lower while the sensitivity is calculated to be 0. The volume errors are smaller than the objective errors. The errors for the void elements are bigger but remain within an acceptable range.

The results of the level-set method FD check can be seen in Figure C.2. It can again be seen that the objective and volume errors of all nodes besides the void nodes are small. The error in the void nodes is caused by the use of the Heaviside function. This operation results in very low sensitivities in the order of  $10^{12}$  or lower. The absolute errors in these nodes are thus very small, despite their large relative error. The relative error in the solid nodes is not as large because these nodes are not as close to 1 as the void nodes are to -1.

The same elements and nodes have been checked for the shell method. The results can be seen in Figures C.3 and C.4 respectively. The errors are somewhat bigger for the shell method. The method results in bigger errors for intermediate element 3271. This increase is likely due to the combination of the density, level-set, and shell at this point. For that reason, this error is considered to be acceptable. The combination of the level-set and density fields also further increases the errors in the void nodes. The absolute errors in these points are even smaller than they are for the individual methods, in the order of  $10^{14}$  or lower.

From this data, it can be concluded that the calculation of the sensitivities works as intended for the three methods.

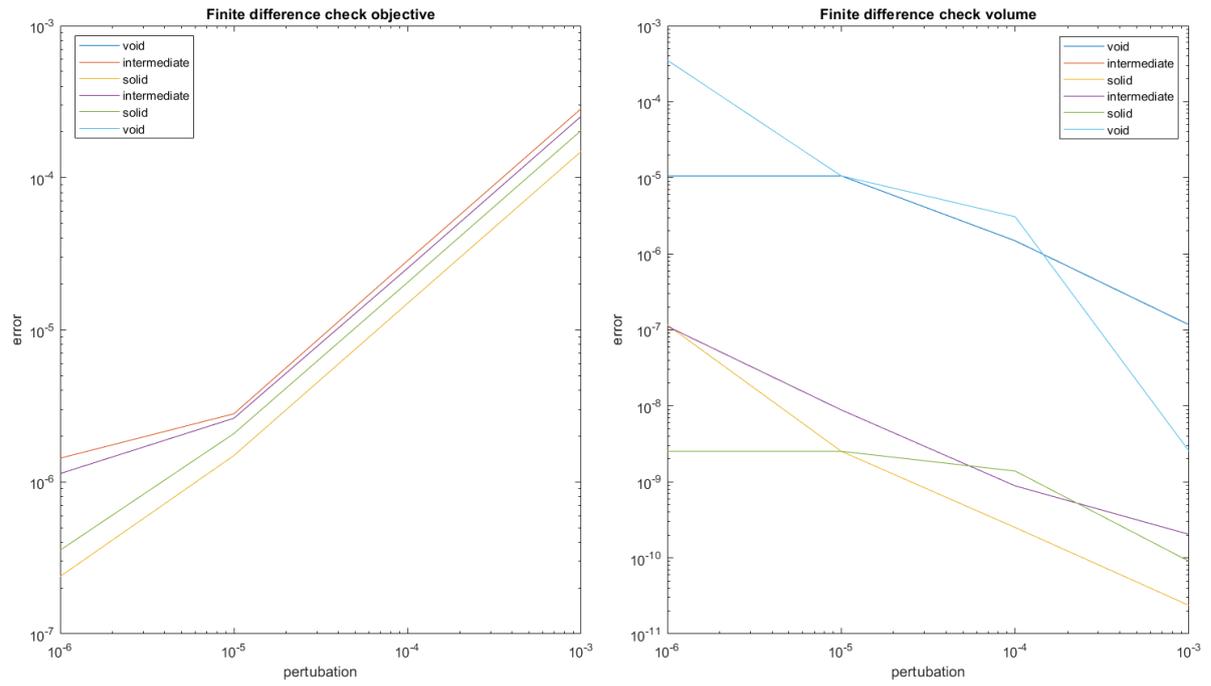


Figure C.1: Finite difference check of the SIMP method.

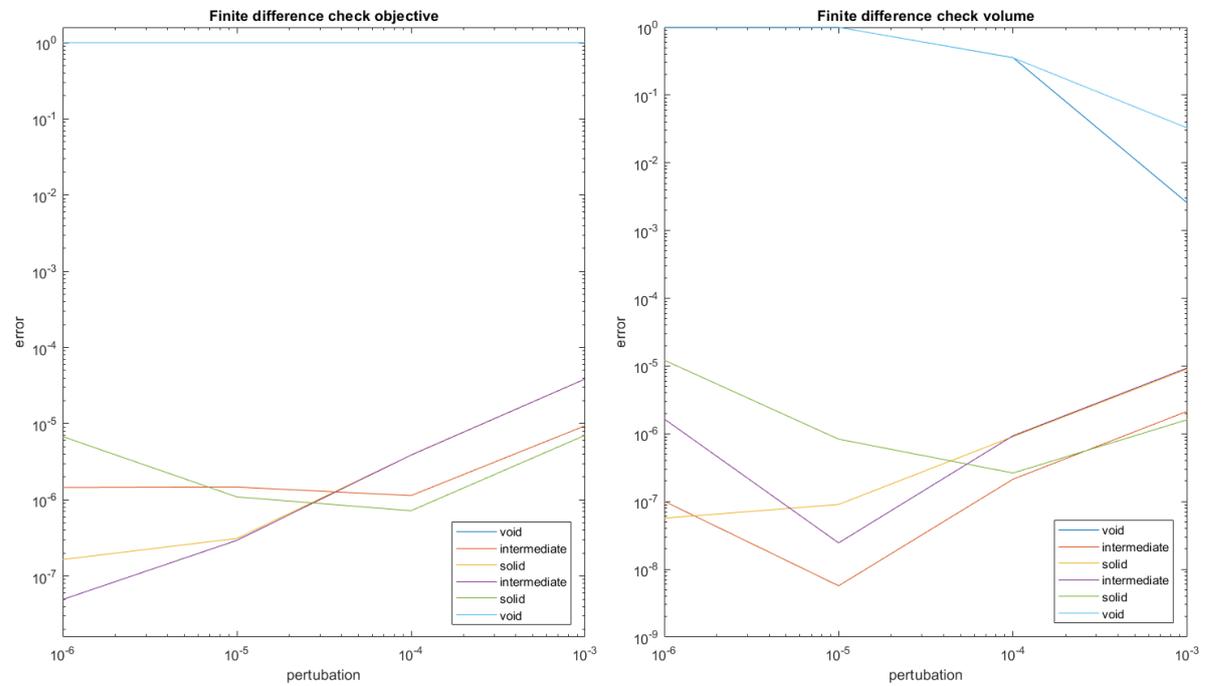


Figure C.2: Finite difference check of the level-set method.

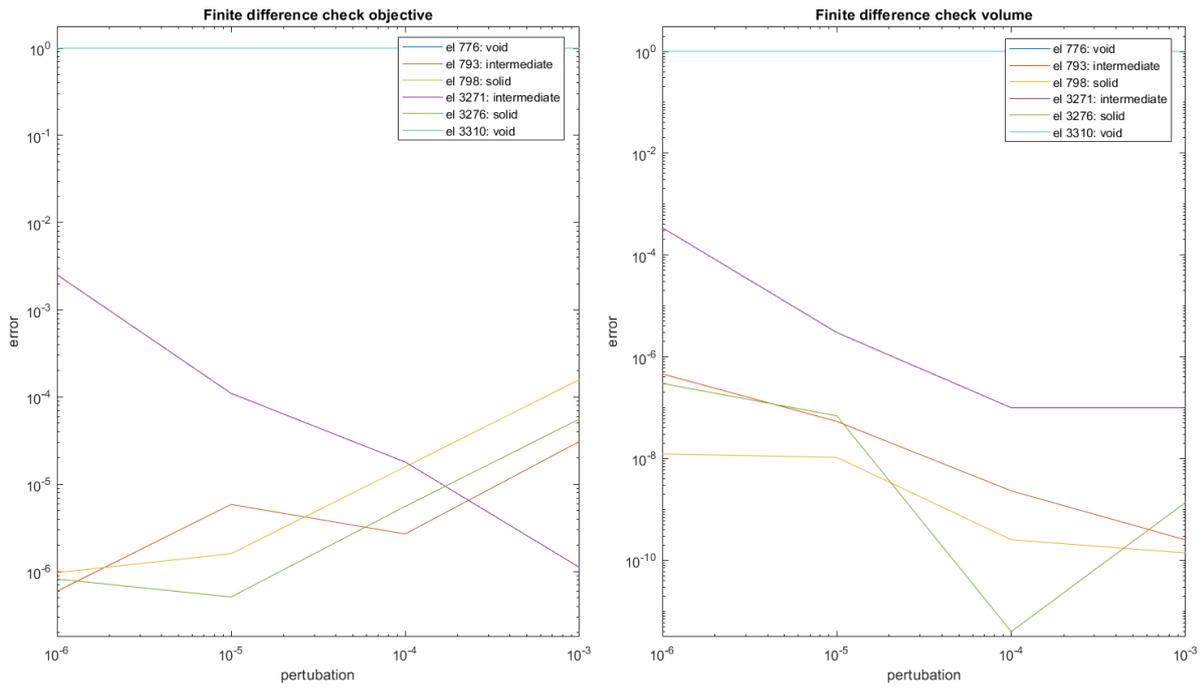


Figure C.3: Finite difference check of the elements of the shell method.

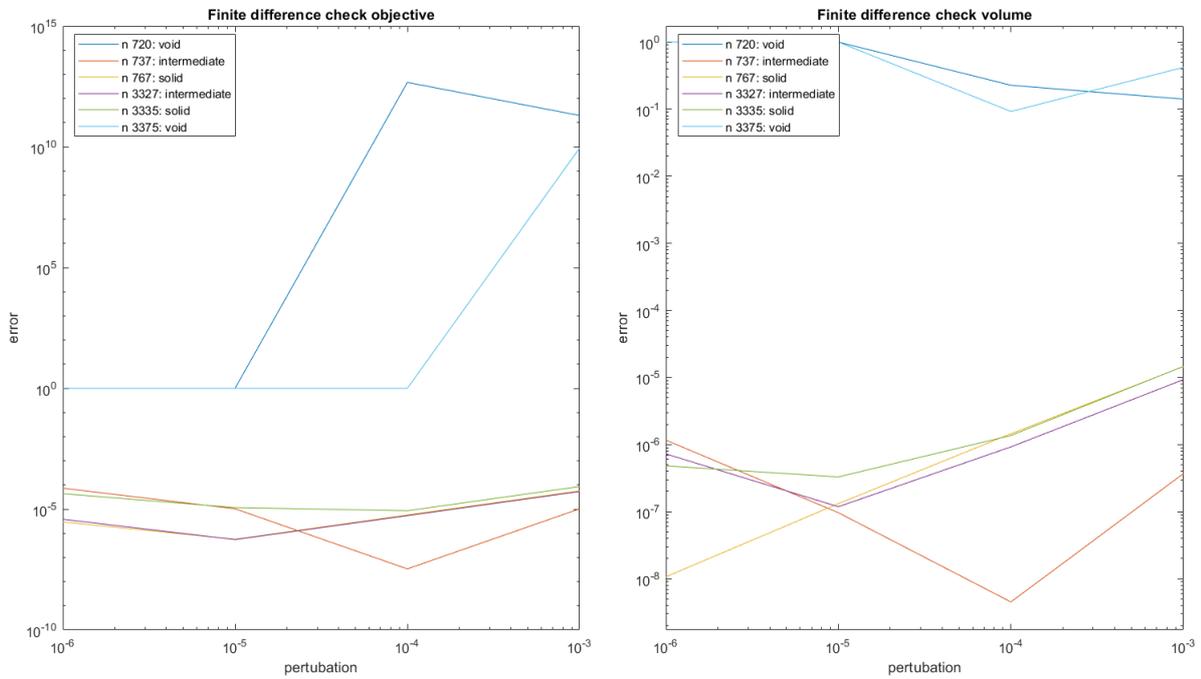
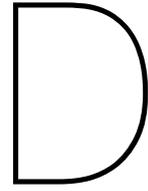


Figure C.4: Finite difference check of the nodes of the shell method.





# Parameters

To pick the right values for different parameters, the effect of each parameter has been investigated. In this appendix, the maximum step sizes, the steepness of the Heaviside function, and the number of sample points are discussed.

## D.1. Maximum step sizes

The maximum step size controls the size of the steps the optimizer is allowed to make for a design variable. There is a separate limit for both the  $\mathbf{x}$  field and the  $\phi$  field. The higher this limit is the faster the optimizer can converge. However, if the limit is too high the optimizer can overshoot and even fail to find the best solution. The level-set field is much slower to converge than the density field. The boundary of the level-set field has to move whereas the SIMP method can place or remove material anywhere. This makes the move limit of the  $\phi$  field the main driver of the convergence speed. For this reason, the move limit of  $\mathbf{x}$  has been set to a fairly conservative value of 0.02. Higher values have a too small impact on the speed of the hybrid method to be of much use. The maximum step size of the level-set field is especially interesting due to the impact it has on the shell. The shell is usually the last part of the structure to converge. The impact of the step size on the number of iterations needed to converge the shell has been investigated. The four points setup from Section 5.1 has been used for this. The experiment has been performed with the default parameters from Table 5.7. The stopping criteria for all experiments are an objective smaller than 0.01, or a maximum change of a single element in any field smaller than 0.003. Figure D.1 shows the convergence when using a step size of 0.05. The quick initial convergence can be seen after which the convergence of the shell continues. The shell is fully converged after about 350 iterations. This number can be reduced to about 210 by increasing the maximum step size to 0.1, as can be seen in Figure D.2. Further increasing the step size to 0.3 and 0.5 results in about 150 and about 130 iterations. This can be seen in Figures D.3 and D.4 respectively. It is clear that as the step size is further increased, the returns diminish. When increasing the value to 0.5, it has been found that the program might not properly converge. This has been the case when running the four points setup with parameters  $\eta = 0.15$ ,  $r_\phi = 16$ , and  $\beta = 24$ . This can be seen in Figure D.5. This behaviour has not been encountered for a value of 0.3. For this reason, the level-set move limit has been set to 0.3.

## D.2. Beta

The  $\beta$  parameter impacts the steepness of the Heaviside step function, as explained in Appendix B. When it comes to the mapping, a higher value of  $\beta$  will lead to sharper edges.

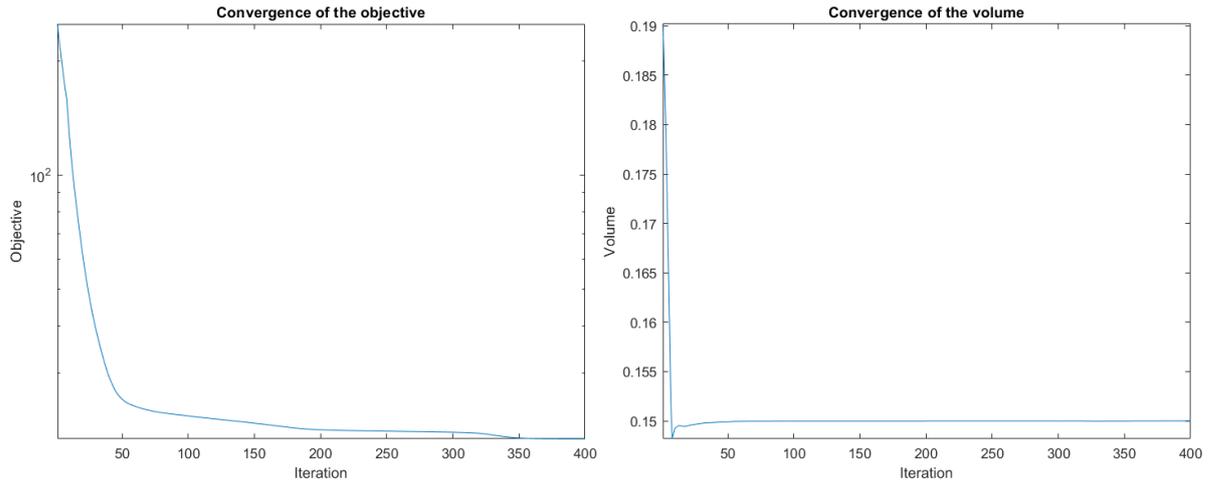


Figure D.1: The convergence for a step size of 0.05. The shell is fully converged after about 350 iterations.

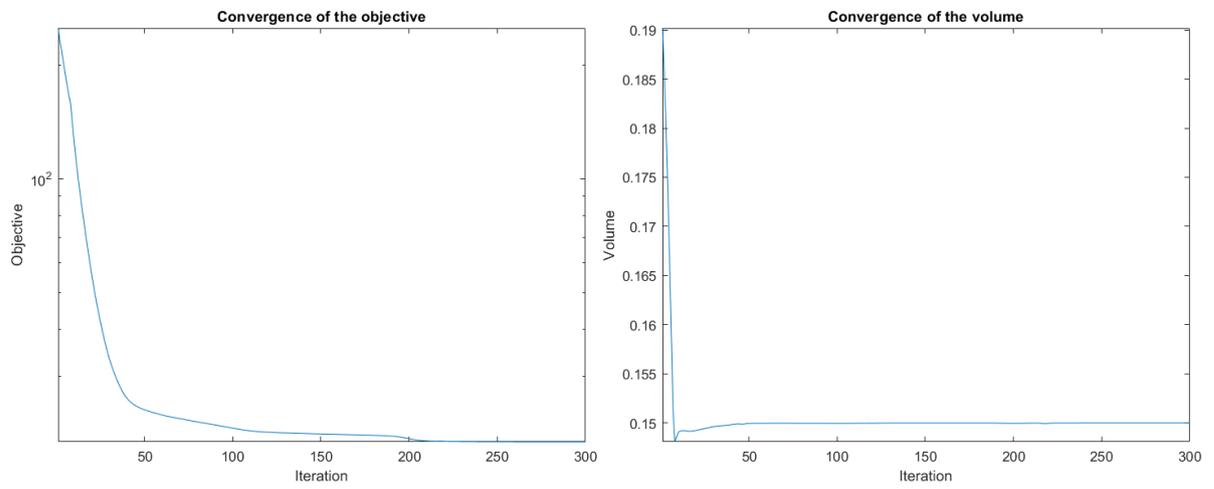


Figure D.2: The convergence for a step size of 0.1. The shell is fully converged after about 210 iterations.

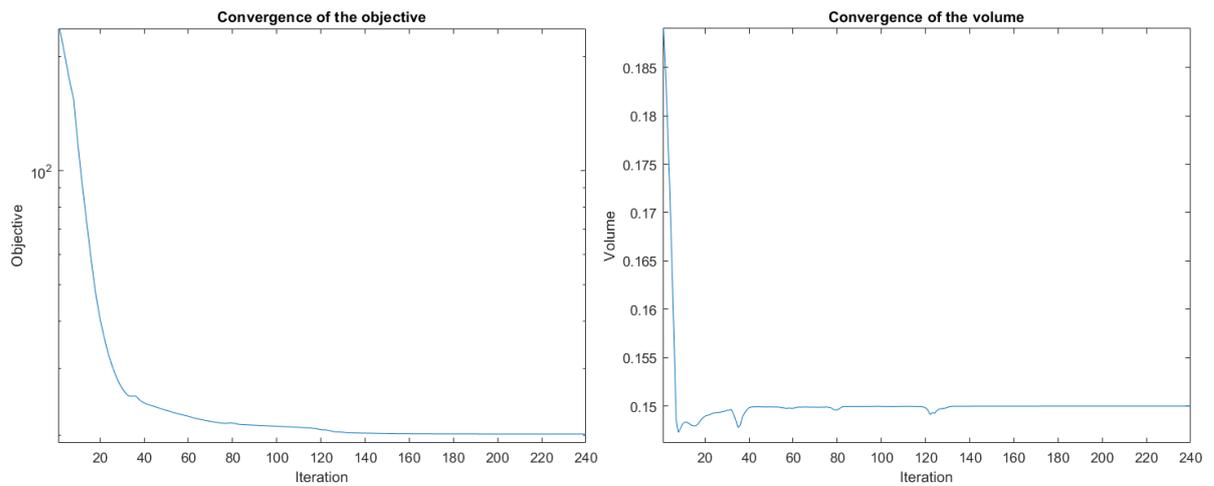


Figure D.3: The convergence for a step size of 0.3. The shell is fully converged after about 150 iterations.

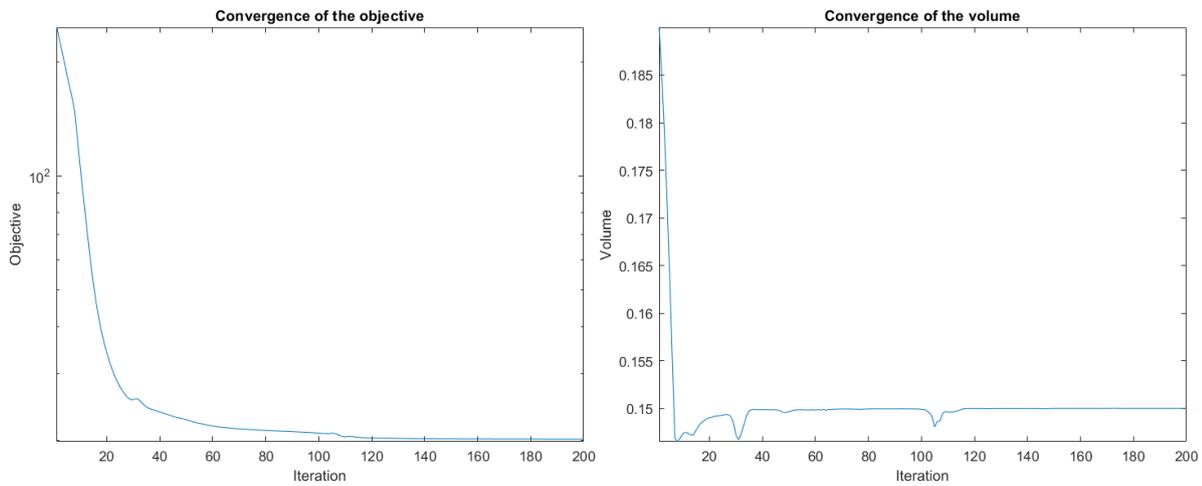


Figure D.4: The convergence for a step size of 0.5. The shell is fully converged after about 130 iterations.

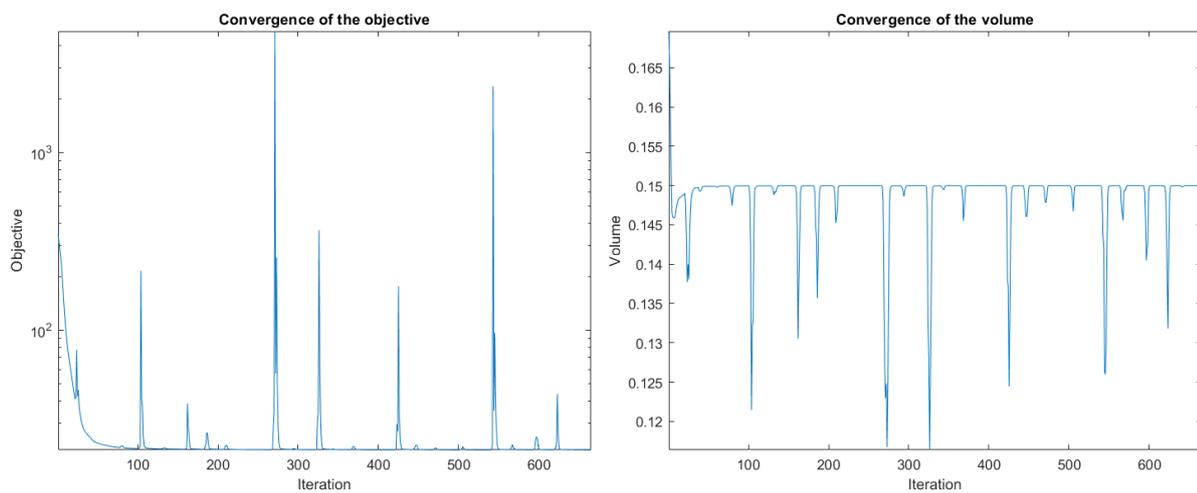


Figure D.5: The lack of proper convergence for a step size of 0.5 with parameters  $\eta = 0.15$ ,  $r_\phi = 16$ , and  $\beta = 24$ .

Figure D.6: The four points setup with  $\beta = 2$ Figure D.7: The four points setup with  $\beta = 6$ 

Moreover, it will reduce the sensitivities further from the edges of the level-set function. These are both desirable features but high values can lead to problems with convergence. Values below 4 result in very blurry edges, this is especially problematic for the shell, rendering it effectively useless. This can be seen in Figure D.6. Values between 4 and 8 do result in a more solid shell but they still have some blurry edges, as can be seen in Figure D.7. On the higher end of the scale, other problems appear. Values above 16 already need significantly more iterations due to slower convergence. It is for that reason that the value of 12 has been chosen as a default. It should be noted that situations might arise in which the sensitivities inside a structure are high enough that the topology is changed by the level-set. This could be a reason to pick a higher value for  $\beta$ , despite convergence problems.

### D.3. Number of sample points

As explained in Section 4.4, a number of local  $\phi$  variables is used to calculate the density of an element. As mentioned in Section 4.4, the chosen numerical integration is not as accurate as a method that uses a Gaussian Quadrature rule. This inaccuracy can be overcome by using more sample points. However, using more sample points does increase the computation time. A simple experiment has been performed to test the effect that different values of  $n$  have. The parameter  $n$  represents the number of sample points for a single direction. Thus, the total number of sample points is  $n^2$ . This experiment gives some insight into the effect the number of sample points has on the computational time and final result. The four points setup has been used to test the time it takes to converge using values 1 to 24 for  $n$ . The stopping criteria for all experiments are an objective smaller than 0.01, or a maximum change of a single element in any field smaller than 0.003. The results can be seen in Figure D.8. It is clear that increasing  $n$  increases the time required to find the solution. The slight dips at the  $n$  values of 5, 9, 15, and 21 are caused by the program not converging as much as it has done for other values, stopping early. This is indicated by an increase in the objective value which can be seen in Figure D.9. A lower objective value can be seen for  $n < 4$ . This decrease is caused by a reduction in the number of intermediate values in the final result. It can be concluded that the value for  $n$  should be at least 3. Because the numerical integration method is not the most accurate one, the value that has been selected for  $n$  is 7. This value is very much on the safe side without costing too much computational time.

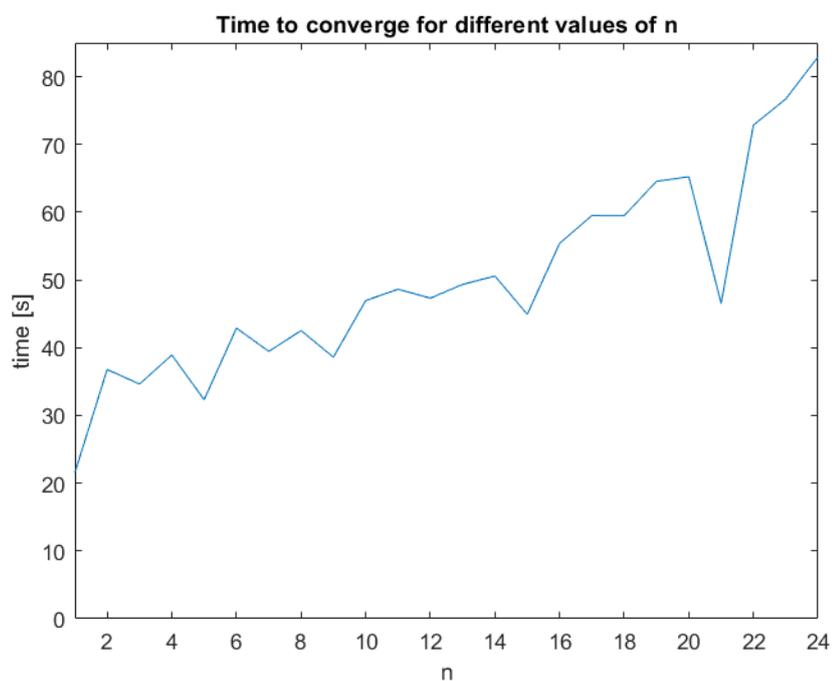


Figure D.8: A graph of the impact that  $n$  has on the convergence time.

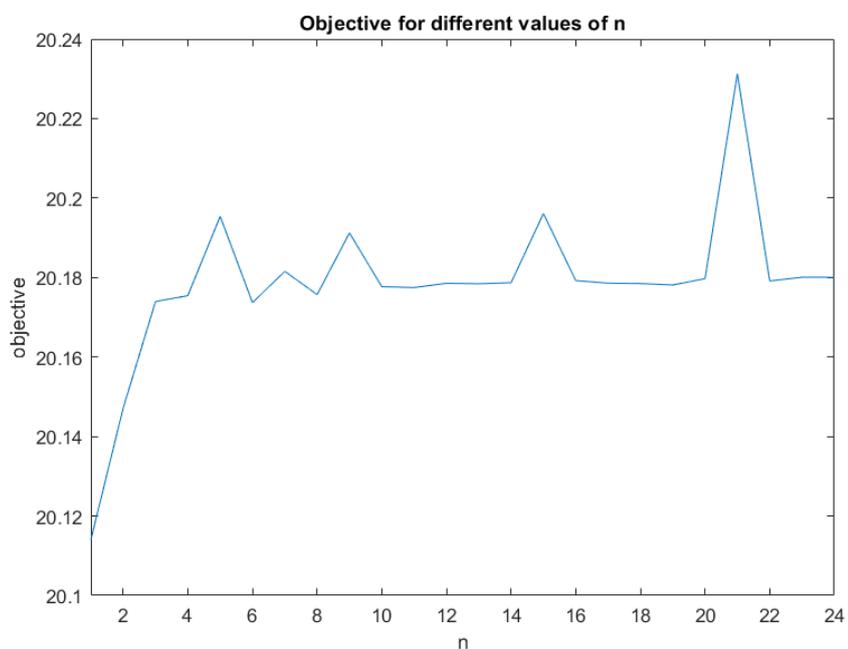
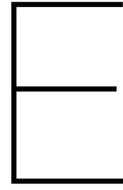


Figure D.9: A graph of the impact that  $n$  has on the objective value.





## Boundary condition tables

node number	coordinates (x,y)
12287	95,31
12351	95,95

Table E.1: Node numbers and coordinates of the fixed nodes for the c setup for a 128x128 design domain.

node number	coordinates (x,y)	direction	magnitude
4031	31,95	(-x,-y)	$\sqrt{2}$
4095	31,31	(-x,+y)	$\sqrt{2}$

Table E.2: Node numbers, coordinates, direction, and magnitude of the forces for the c setup for a 128x128 design domain.

node number	coordinates (x,y)
12287	95,31
12351	95,95

Table E.3: Node numbers and coordinates of the fixed nodes for the x setup for a 128x128 design domain.

node number	coordinates (x,y)	direction	magnitude
4031	31,95	(-x,+y)	$\sqrt{2}$
4095	31,31	(-x,-y)	$\sqrt{2}$

Table E.4: Node numbers, coordinates, direction, and magnitude of the forces for the x setup for a 128x128 design domain.

node number	coordinates (x,y)
3321	25,95
4101	31,101
7191	55,95
7971	61,101
12281	95,25
13061	101,31

Table E.5: Node numbers and coordinates of the fixed nodes for the three points setup for a 128x128 design domain.

node number	coordinates (x,y)	direction	magnitude
2431	18,108	(+x,-y)	$\sqrt{2}$
8881	68,108	(+x,-y)	$\sqrt{2}$
13951	108,18	(+x,-y)	$\sqrt{2}$

Table E.6: Node numbers, coordinates, direction, and magnitude of the forces for the three points setup for a 128x128 design domain.

node number	coordinates (x,y)
12287	95,31
12351	95,95

Table E.7: Node numbers and coordinates of the fixed nodes for the equals sign setup for a 128x128 design domain.

---

node number	coordinates (x,y)	direction	magnitude
4031	31,95	(-x)	1
4095	31,31	(-x)	1

Table E.8: Node numbers, coordinates, direction, and magnitude of the forces for the equals sign setup for a 128x128 design domain.



# Bibliography

- Andreasen, C. S., Elingaard, M. O., & Aage, N. (2020). Level set topology and shape optimization by density methods using cut elements with length scale control. *Structural and Multidisciplinary Optimization*, 62. <https://doi.org/10.1007/s00158-020-02527-1>
- Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., & Sigmund, O. (2011). Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43. <https://doi.org/10.1007/s00158-010-0594-7>
- Basting, C., & Kuzmin, D. (2014). Optimal control for mass conservative level set methods [Fourth International Conference on Finite Element Methods in Engineering and Sciences (FEMTEC 2013)]. *Journal of Computational and Applied Mathematics*, 270, 343–352. <https://doi.org/https://doi.org/10.1016/j.cam.2013.12.040>
- Bendsøe, M. P. (1989). Optimal shape design as a material distribution problem. *Structural Optimization*, 1. <https://doi.org/10.1007/BF01650949>
- Bendsøe, M. P., & Kikuchi, N. (1988). Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71. [https://doi.org/10.1016/0045-7825\(88\)90086-2](https://doi.org/10.1016/0045-7825(88)90086-2)
- Bendsoe, M. P., & Sigmund, O. (2003). *Topology optimization: Theory, methods, and applications*. Springer Science & Business Media.
- Bruns, T. E., & Tortorelli, D. A. (2001). Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190(26), 3443–3459. [https://doi.org/https://doi.org/10.1016/S0045-7825\(00\)00278-4](https://doi.org/https://doi.org/10.1016/S0045-7825(00)00278-4)
- Cavazzuti, M., Baldini, A., Bertocchi, E., Costi, D., Torricelli, E., & Moruzzi, P. (2011). High performance automotive chassis design: A topology optimization based approach. *Structural and Multidisciplinary Optimization*, 44(1), 45–56.
- Dijk, N. P. V., Maute, K., Langelaar, M., & Keulen, F. V. (2013). *Level-set methods for structural topology optimization: A review*. <https://doi.org/10.1007/s00158-013-0912-y>
- Galjaard, S., Hofman, S., & Ren, S. (2015). *New opportunities to optimize structural designs in metal by using additive manufacturing*. [https://doi.org/10.1007/978-3-319-11418-7\\_6](https://doi.org/10.1007/978-3-319-11418-7_6)
- Geiss, M. J., & Maute, K. (2018). Topology optimization of active structures using a higher-order level-set-xfem-density approach. *2018 Multidisciplinary Analysis and Optimization Conference*. <https://doi.org/10.2514/6.2018-4053>
- Giele, R., Groen, J., Aage, N., Andreasen, C. S., & Sigmund, O. (2021). On approaches for avoiding low-stiffness regions in variable thickness sheet and homogenization-based topology optimization. *Structural and Multidisciplinary Optimization*, 64. <https://doi.org/10.1007/s00158-021-02933-z>
- Giele, R., van Keulen, F., & Langelaar, M. (2022). Design for drainability in density-based topology optimization. *Structural and Multidisciplinary Optimization*, 65. <https://doi.org/10.1007/s00158-022-03272-3>
- Guest, J. K., Prévost, J. H., & Belytschko, T. (2004). Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering*, 61. <https://doi.org/10.1002/nme.1064>
- Guo, X., Zhang, W., & Zhong, W. (2014). Explicit feature control in structural topology optimization via level set method. *Computer Methods in Applied Mechanics and Engineering*, 272, 354–378. <https://doi.org/https://doi.org/10.1016/j.cma.2014.01.010>

- Haber, R. B., & Bendsøe, M. P. (1998). Problem formulation, solution procedures and geometric modeling: Key issues in variable-topology optimization. *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. <https://doi.org/10.2514/6.1998-4948>
- Kuzmin, D. (2014). An optimization-based approach to enforcing mass conservation in level set methods. *Journal of Computational and Applied Mathematics*, 258, 78–86. <https://doi.org/https://doi.org/10.1016/j.cam.2013.09.009>
- Li, W., Garg, S., & McMains, S. (2009). 2d accessibility analysis for waterjet cleaning. *Computer-Aided Design and Applications*, 6. <https://doi.org/10.3722/cadaps.2009.231-241>
- Merulla, A., Gatto, A., Bassoli, E., Munteanu, S. I., Gheorghiu, B., Pop, M. A., Bedo, T., & Munteanu, D. (2019). Weight reduction by topology optimization of an engine subframe mount, designed for additive manufacturing production. *Materials Today: Proceedings*, 19, 1014–1018.
- Moreira, A., & Santos, M. Y. (2007). Concave hull: A jt-nearest neighbours approach for the computation of the region occupied by a set of points. *GRAPP 2007 - 2nd International Conference on Computer Graphics Theory and Applications, Proceedings, GM*. <https://doi.org/10.5220/0002080800610068>
- Muyl, F., Dumas, L., & Herbert, V. (2004). Hybrid method for aerodynamic shape optimization in automotive industry. *Computers & Fluids*, 33(5-6), 849–858.
- Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79. [https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2)
- Papoutsis-Kiachagias, E. M., & Giannakoglou, K. C. (2016). Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives of Computational Methods in Engineering*, 23. <https://doi.org/10.1007/s11831-014-9141-9>
- Shi, X., & Li, C. (2021). Convexity preserving level set for left ventricle segmentation. *Magnetic Resonance Imaging*, 78, 109–118. <https://doi.org/https://doi.org/10.1016/j.mri.2021.02.003>
- Sigmund, O. (2001). A 99 line topology optimization code written in matlab. *Structural and multidisciplinary optimization*, 21(2), 120–127.
- Sigmund, O. (2007). Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33. <https://doi.org/10.1007/s00158-006-0087-x>
- Sigmund, O., & Maute, K. (2013). *Topology optimization approaches: A comparative review*. <https://doi.org/10.1007/s00158-013-0978-6>
- Svanberg, K. (1987). The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24. <https://doi.org/10.1002/nme.1620240207>
- Wang, F., Lazarov, B. S., & Sigmund, O. (2011). On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization*, 43. <https://doi.org/10.1007/s00158-010-0602-y>
- Wei, P., Li, Z., Li, X., & Wang, M. Y. (2018). An 88-line matlab code for the parameterized level set method based topology optimization using radial basis functions. *Structural and Multidisciplinary Optimization*, 58. <https://doi.org/10.1007/s00158-018-1904-8>
- Zhu, J. H., Zhang, W. H., & Xia, L. (2016). Topology optimization in aircraft and aerospace structures design. *Archives of Computational Methods in Engineering*, 23. <https://doi.org/10.1007/s11831-015-9151-2>