Development of a Three-Dimensional Topology Optimization Algorithm
for Mass-Optimized Cast Glass Components

Eva Schoenmaker

Course	MSc in Architecture, Urbanism & Building Sciences (Building Technology Track)
Studio	Building Technology Gradution Studio
Mentors	Dr. Charalampos Andriotis Dr. Faidra Oikonomopoulou
Student	Eva Schoenmaker
Student Number	4693906
Submitted On	03-11-2023

In memory of Amber
I know you would have been proud

Abstract

This thesis is part of a larger ongoing research project at the TU Delft regarding structural glass as a novel construction material. Here, the focus is directed towards the casting of glass to create complex free-form geometries. Thus far, only smaller components have been created, but a multitude of previous student work explored how the current limitations in the fabrication process could be overcome by using topology optimization to design larger cast glass components. This thesis continues this research by integrating the limitations and remarks concluded from previous work to further explore the potential of designing large cast glass components using topology optimization.

While commercial topology optimization software exists, the application of those for the design of optimizing glass structures is hampered by their use of von Mises stresses. Glass is a brittle material, with considerably higher compressive strength than tensile strength. The usage of von Mises stresses makes it impossible to utilise both the compressive and tensile strength. A two-dimensional optimization algorithm has been created that integrates the two strength constraints as well as design-specific constraints related to cast glass as annealing and manufacturing criteria.

This thesis intends to contribute to the existing research by creating a tool for the design of 3D-optimized cast glass components, which can take into consideration the structural properties of glass as well as manufacturing, annealing and specific design criteria. The tool is created in Matlab, using the SIMP optimization method with 8-noded hexahedral finite elements for the structural model.

The literature review encompasses a wide range of topics. First, the topics regarding the structural application of glass will be covered. These encompass the structural properties of glass, various casting methods and mould types. This exploration leads to an understanding of the constraints involved in designing with glass.

Next, the review explores topics related to structural optimization. Different structural optimization methods will be covered. Additionally, it discusses various finite element types, including one-dimensional, two-dimensional, and three-dimensional models, to determine the most suitable choice for setting up the structural model. Furthermore, different formulations of the optimization problem were considered including, stress, compliance and volume based approaches, to identify the most appropriate objective for the optimization problem. Lastly, existing topology optimization algorithms are analysed to gain an understanding of how to establish an efficient algorithm.

From the literature review it was concluded that the SIMP methodology is best suited the requirements of this project. For the structural model, 8-noded hexahedral elements were selected. Two algorithms were developed: one with a compliance based objective and a second with a volume based objective.

The algorithm's performance was tested on different domains. To facilitate comparison with existing literature, the algorithm was initially calibrated with a domain width of one element, effectively creating a two-dimensional optimization. Subsequently, its performance was assessed through the application of a small three-dimensional problem. Finally, the algorithm is used to design a case study involving the structural slab of a small pedestrian bridge located inside the British Museum.

The results validate the benefits of the 3D algorithm. The geometries optimized in three dimensions have a higher structural stiffness and offer more spatial insight. Moreover, the volume optimizations results in a larger material reduction than those achieved in two-dimensional optimizations. For this reason, the volume-optimized component was selected for post-processing and implementation in the final design.

Contents

1
Research Framework

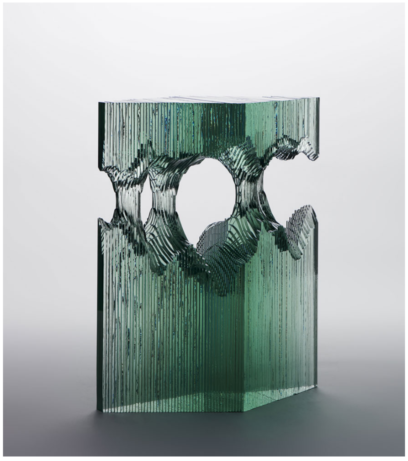
This chapter will give an introduction to the topic discussed in this thesis. First, a general introduction to the topic of glass as a structural material will be given. Then the problem statement will be explained, from which the goal and research questions for this thesis follow. The relevance of this research will be explained alongside the methodology that will be used to answer these research questions. The chapter will end with the research outline and proposed timeline.

1.1 Introduction: Glass as a structural material

Glass is a versatile and widely-used material in the built environment. Its applications range from the stained glass windows in churches to the curtain walls in modern skyscrapers. Advancements in glass technologies and engineering have shown that glass has more potential than a simple infill material. It has a high strength in compression, which makes it suitable for load-bearing applications. Several examples exist where glass has been used as a structural material. Most notable are the complete glass structures designed by EOC engineers depicted in figure 1.1a[1]. The pursuit of maximum transparency has led to larger glass sheets and the desire for fewer connections.

Nevertheless, as most of the glass industry for the built environment is centered around the float glass industry, architects can find themselves constrained by limited possibilities in terms of design, shapes and dimensions. The inherent nature of float glass dictates that structures primarily rely on two-dimensional, planar elements. While art installations, as demonstrated in figure 1.1b, have successfully achieved three-dimensional forms through lamination, this approach can compromise structural integrity.

(a) Apple store Fifth Avenue



(b) Laminated glass artwork by Ben Young

Figure 1.1: Two examples of float glass

Cast glass, on the other hand, offers the possibility to achieve fully transparent structures with almost unlimited freedom in shape. Its potential has been shown in projects like the Crystal Houses facade (Oikonomopoulou, 2019). Further experiments have shown that it is possible to use the shape freedom to create complex and accurate interlocking elements, depicted in figure 1.2, that can be used for structural applications.

While not limited in geometry, these elements are limited in their dimensions. This is due to long annealing process required for larger elements. Annealing reference to the process after casting, where elements are cooled down slowly to resolve any stresses in the component. The duration of this period depends largely on the mass and shape of the elements. As a result, larger elements will take more time, energy and money to produce.

1.2 Problem Statement

Previous works have researched the potential of creating massive structural cast glass components using Topology Optimization (TO). Topology optimization is a method to reduce the volume and thus mass of structural elements while keeping the stiffness. When applied to glass elements, this can greatly reduce the mass and the annealing time of elements. Increasing the possibility of creating larger cast glass elements. Damen (2019) showed that topology optimization can be used as a tool to limit the mass of glass components and thus reduce the annealing time. Furthermore, Bhatia (2019) proved the feasibility of using 3D printed sand moulds to create complex TO geometries.

[1] For sources images see list of figures

(a) (b)

Figure 1.2: (a) Qaammat Pavilion in Greenland, constructed out of adhesively-bonded glass bricks. Photo credits:Julien Lanoo; (b)Interlocking cast glass components made from clear glass and recycles mouth-blown coloured glass

Nonetheless, the research into the topology optimization of cast glass components has been limited by constraints in commercial software. Existing topology optimization algorithms in commercial software are oriented towards ductile materials like metals, that exhibit similar stress constraint in tension and compression. Glass however has a large difference in tensile and compressive stress, with the allowable compressive stress being a factor 100 higher. As a result, the optimized geometries generated with current software solutions underutilize glass's compressive strength. Furthermore, the current workflow requires a post-optimization step to evaluate the compressive stress values within the structure

Last year, a thesis resulted in the development of a TO algorithm that is specifically tailored to glass. It includes formulas for the specific properties of glass, eliminating the need for previously mentioned additional steps. This algorithm can effectively optimize 2D structures, achieving designs with a 70% mass reduction. This raises the question if further minimization of the mass and thus annealing time can be achieved by an algorithm that can directly optimize in three dimensions.

1.3 Research goal

This thesis intends to contribute to the existing research by creating a tool for the design of 3D-optimized cast glass components, which can take into consideration the structural properties of glass as well as manufacturing, annealing and specific design criteria.

To add to this main goal, this research has the following secondary goals:

- Analyze how different objective functions and constraints influence the design generated by the proposed algorithm.
- Analyze how the proposed algorithm can be applied as a design tool by generating a design for a topologically-optimized slab for a case study.
- Evaluating the algorithm by comparison of the results to predecessors derived with the 2D algorithm.

1.4 Research questions

Main question:
What are the main aspects and limitations of a three-dimensional topology optimization for the design of massive cast glass structures?

Sub-questions:

1. What are the design constraints posed by the chosen case study?

2. What are the structural and manufacturing constraints for cast glass, and where do they differ for 2D and 3D optimization?
3. Which algorithm methodology is most suitable for the topology optimization process?
4. Which finite element type is most suitable for the topology optimization process?
5. How can the computational time be limited to ensure a feasible calculation?
6. How do different objective functions influence the outcome of the topology optimization?
7. How do the results of the 3D optimization compare to the outcome of the 2D optimization in terms of structural perfomance, material utilization and geometry?
8. What are the main benefits and limitations of the developed 3D algorithm compared to existing 2D algorithms for cast glass components?

1.5 Methodology

To answer the research questions and main objectives of this thesis, a combination of different methods is applied. This research consists of three main phases: exploration, development and evaluation.

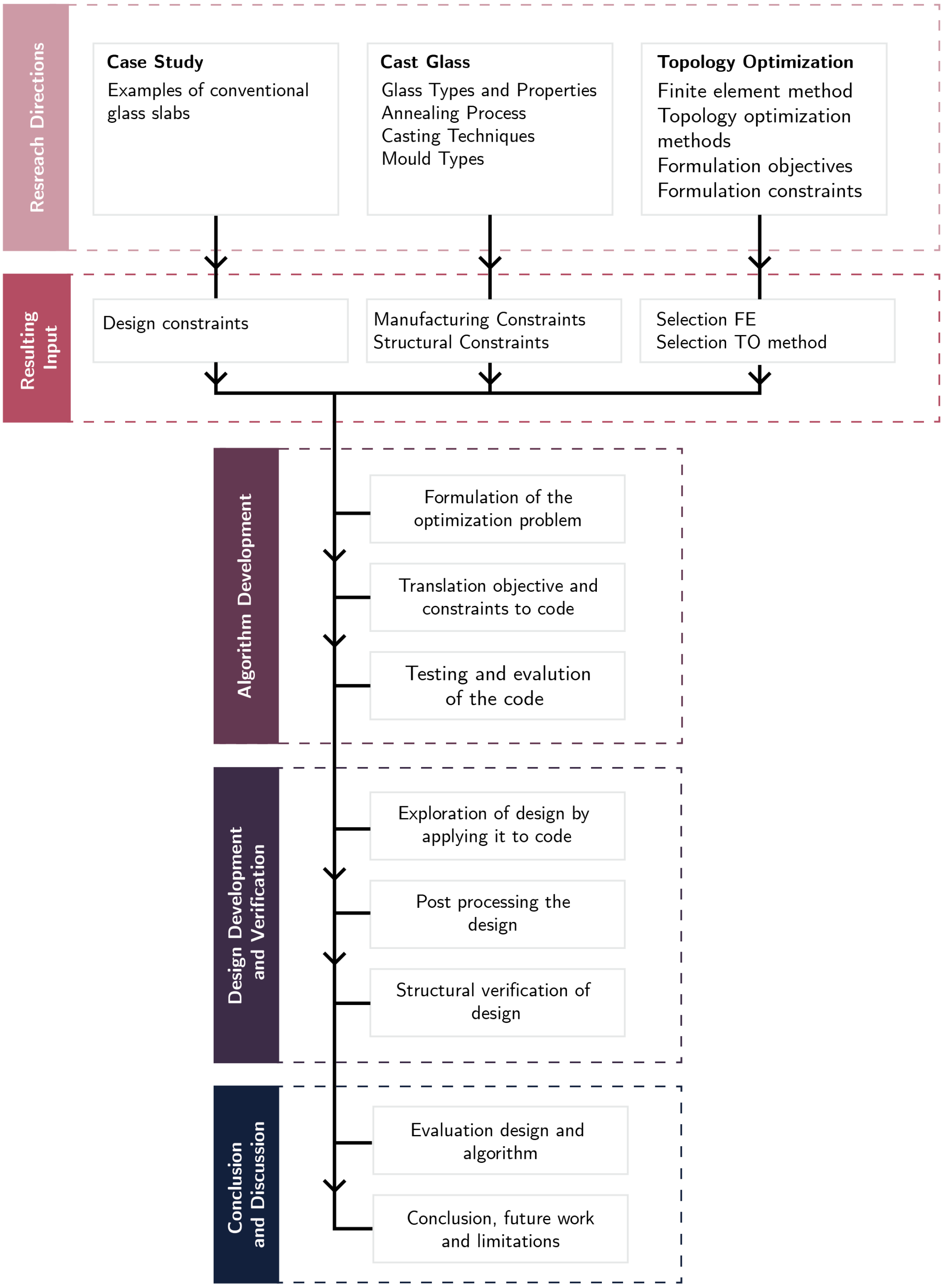
The research starts with an exploration of the topic, involving gathering and analyzing relevant literature. This exploration results in a literature review, which encompasses the scientific studies and the theses which have been done on similar topics. The two main topics of this review are structural glass and structural optimization methods. In addition to this, the literature review discusses a set of different case studies and their respective characteristics.

The information collected on the properties of glass and the process of casting glass will be used for defining the structural, manufacturing and annealing criteria. The review of structural optimization methods will be used to define the combination of type, solver and objective that will be applied in the algorithm. The analysis of the different case studies will be used in defining any design constraints that follow out of the location and use of the structure.

The next step will focus on the algorithm development. This phase will start with the formulation of the optimization problem and translating the objectives and constraints following the literature research into an algorithm. As it is expected that the algorithm will take a considerable amount of computational time, testing will first be done on smaller domains. If the code functions as expected, it will be applied to the design domain as given by the case study. In turn, both the problem formulation and the algorithm will be updated as required until a desired outcome is reached.

When a final geometry is determined, the connection details, construction and assembly sequence of the design will be worked out. Lastly, an evaluation of the design in comparison to previous projects (derived with the 2D algorithm) will be done, to draw conclusions on the performance of the new algorithm.

1.6 Research outline



2
Case study

This chapter serves the purpose of introducing the case study. The redesign of a conventional glass structure was chosen as method to test the performance of the topology optimization algorithm through application. The selected structure is a glass bridge in the British Museum. The chapter will focus on the main characteristics of the case study and explain the reasoning behind the selection of this case study.

2.1 Examples of Glass Bridges

In various architectural contexts, achieving transparency of floors, is a focus of the design concept. These floors create a sense of unique experience and are often implemented when there are impressive or important views. Most of these floors and bridges consist of a walking surface constructed out of laminated tempered glass panels that are supported by a metal or concrete substructure. The size of the glass sheets is limited by the deformation of the glass sheet. However, this substructure can fragment the view. As can be seen in the examples of the glass floors in the Acropolis museum (figure 2.1). This specific floor was constructed to show the findings of archaeological excavation on the site of the museum.

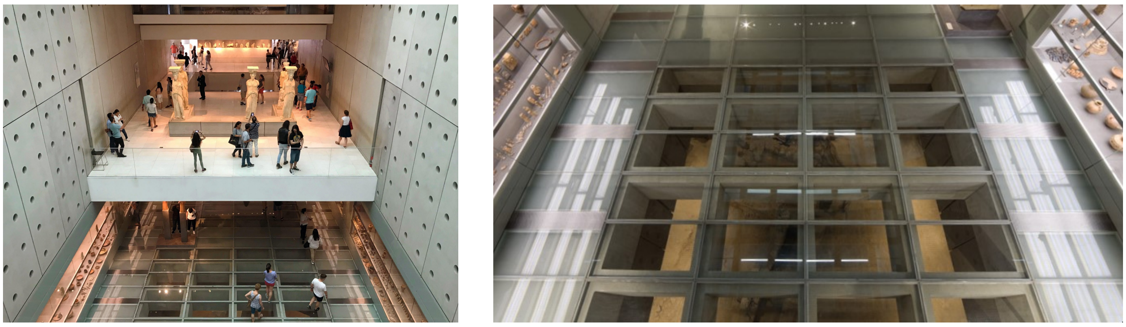


Figure 2.1: Glass floor at the Acropolis museum

A second approach is to the design of glass floors is to replace the supporting substructure with laminated glass beams and fins. This creates fully glass structures, as can be seen in the example of the glass cube in New York (figure 2.2a). While this kind of structure achieves more optical clarity, it is not possible to completely avoid visible steel connections.

Instead of having simply layered glass beams, engineers of ABT have experimented with corrugated glass sheets (figure 2.2b). By bending the sheets into a half circle, additional stiffness is created. Making a larger span of 6 meters possible. [1]

(a) Fully transparent Apple Cube in New York, Engineering by EOC Engineers.

(b) Glass bridge in the Municipal Office Vlaardingen, Design by ABT

Figure 2.2: Two examples of fully glass structures

2.2 Selection of the Case Study

For the selection of the case study, multiple glass slabs were considered. The decision to exclusively consider slabs was based on two factors. First, slabs place substantial demands on material tensile strength, given that their tensile stresses exceed those of components primarily experiencing compressive loads, such as columns or arches. Secondly, these designs often represent an example of a typical glass structure, featuring laminated sheets supported by a metal substructure. Redesigning one such conventional

[1] This example has not been realised fully in glass, as one of the components broke during installation, needing to be replaced by a steel version as the project needed to be finished.

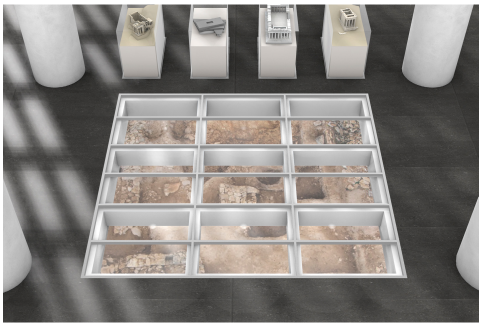
structure will showcase the structural capabilities of large cast glass components. Additionally, it will show the novel architectural language that these cast glass elements can introduce.

For the case study the focus was put on slabs that have already been redesigned using topology optimization methods. These include the redesign of the glass floor in the acropolis museum by Stefanaki (2020) and the redesign of the glass bridge in the British Museum by Koniari (2022). Choosing a case study that has a previous redesign allows for a comparative analysis between the performance of the to be developed algorithm and commercial software in case of Stefanaki (2020) or a two dimensional optimization in case of Koniari (2022).

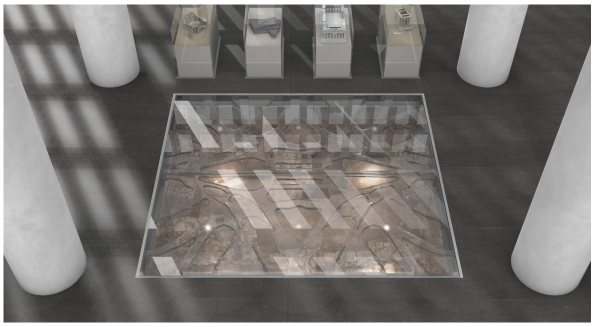
2.2.1 Acropolis Museum, Athens

The Acropolis Museum is designed by the Bernard Tschumi Architects and the construction was completed in 2007. The museum is built over archaeological excavations that were found on site. To show these findings multiple glass floors have been integrated into the museum's ground floor. This allows visitors to observe these archaeological findings. On the ground floor, there are two main glass floors. For one of the floors, the concrete beams are not only carrying the glass floor but also function as a bracing between two columns. For this reason, the other glass floor is a more feasible case study (Stefanaki, 2020).

The current floor has been redesigned by Stefanaki (2020), using the topology optimization software integrated into ANSYS workbench. Figure 2.3 shows that replacing the original structure with a topology optimized slab greatly increases the visibility of the archaeological findings. However, when glass has an organic shape the refraction of light changes with the geometry. Resulting in a possible distortion when viewing the artefacts. With this in mind, this case study offers the possibility to investigate the insurance of a clear view as an extra design constraint.



(a) Render current glass floor



(b) Render redesigned glass floor by Stefanaki (2020)

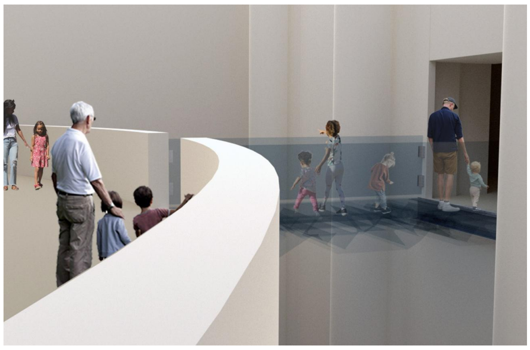
Figure 2.3: Glass floor in the Acropolis Museum

2.2.2 British Museum, London

The redesign of the Great Court of The British Museum by Forster + Partners included a small pedestrian bridge. The bridge crosses from the Reading Room to the rest of the exhibition spaces. The bridge is an example of a glass bottomed bridge. The slab is made from float glass sheets that are supported by a metal substructure. The substructure is in turn connected to the neighbouring walls (Koniari, 2022). The total span of the bridge is approximately 2.30m * 4.20 m * 0.20 m (width, length, height).

As mentioned previously, the bridge has been redesign by Koniari (2022). The design was made using a two dimensional topology optimization specifically created for the design of cast glass structures. This algorithm includes constraints for the maximum annealing time as well as constraints for the specific material properties of glass. Figure 2.4 shows the original structure and the redesign. Having this redesign as a basis allows for a comparative analysis between the performance of two-dimensional optimization and three-dimensional optimization.

(a) Current pedestrian bridge



(b) Redesigned TO bridge by Koniari (2022)

Figure 2.4: Pedestrian bridge British Museum

2.2.3 Case Study Selection and Limitations

In the end, the glass bridge in the British museum is chosen as case study for this thesis. Redesigning this bridge using a the new algorithm allows for a direct comparative study between the performance of the two dimensional algorithm and the three dimensional algorithm. The design only encompasses the monolithic slab, as this is the structural part of the design (figure 2.5). The railing is not integrated in the algorithm and will be added after the design of the slab is finished. For the supports, it is assumed that the slab will be fixed to the neighbouring walls at the short side of the domain. These supports are taken to be fully fixed. There will be two layers of float glass placed on top of the monolithic structure. These will protect the large components from accidental damage and local contact stresses. The bridge narrows in its width. However, to simplify the calculation the domain is always set to a rectangular shape. Thus the slightly curved edges are assumed to be straight.

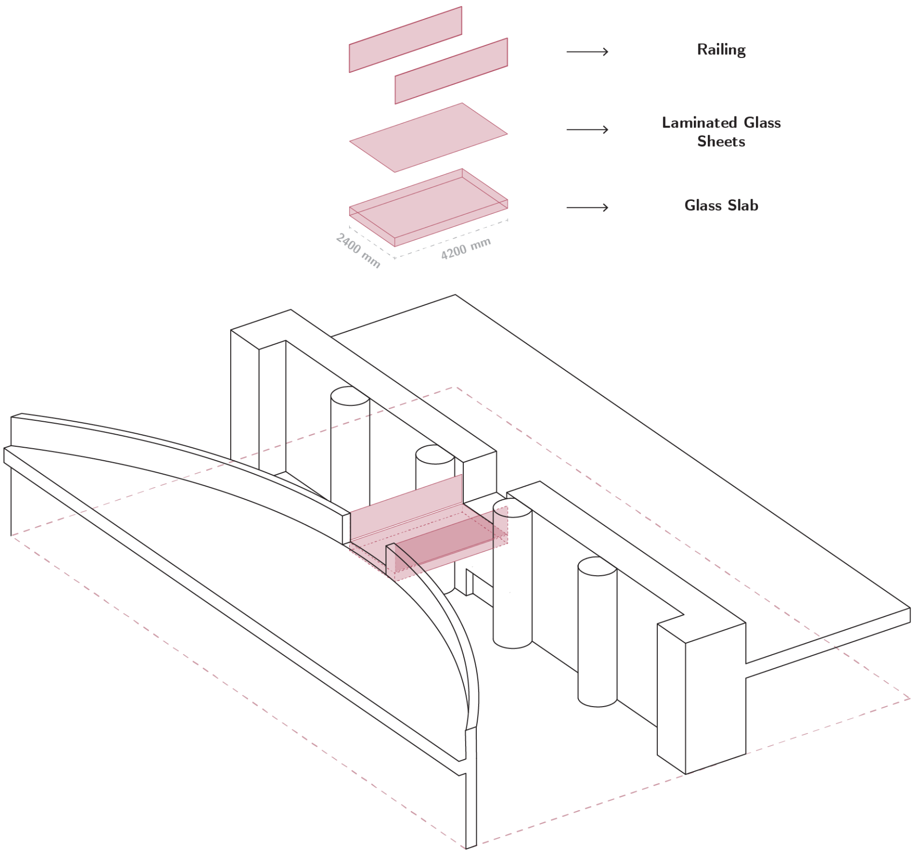


Figure 2.5: Isometric diagram illustrating the design elements

2.3 Load Case and Boundary Conditions

To ensure that the structure is able to support the intended loads and usage, several load cases should be taken into consideration. One scenario is the combination of a dead load with a distributed live load. The dead load consists of the weight of the floor itself, including any finishes. The distributed live load is the weight of people using the floor. For a museum (Category C3) the live load is set as 3,0 to 5,0 $[kN/m^2]$. A second scenario would be the asymmetric loading of the structure, here the live load is just present on one side of the structure. Furthermore, the loads have to be multiplied by a safety factor. For permanent loads this is 1,2. Short term loads, like the live load, have to be multiplied by 1,5.

For the allowable deformation, there are two different norms. One regards the deflection of the component during the construction and placement. The other is stricter and refers to the allowable deformation during use (Koniari, 2022).

	Symbol	Units	Input values
Deflection during construction phase	d_c	m	l / 250
Deflection for serviceability	d_s	m	l / 500

Table 2.1: Design values for deflection in glass components (Koniari, 2022).

3 Glass

This chapter serves the purpose of addressing two sub-questions. Firstly, it focuses on the specific properties of glass. This involves an exploration of different glass types as well as the structural behaviour of glass. Secondly, the chapter will focus on the production process of glass. This involves different mould types and their respective casting methods. Based on the findings of the literature review, the respective structural, annealing and manufacturing criteria will be defined. These will be used as input for the algorithm.

3.1 Glass Properties

Glass is an amorphous material, meaning that on a molecular level it does not have a crystalline structure. The amorphous structure arises from the way glass is formed. When glass is melted and then cooled rapidly, the viscosity of the material changes and the molecules become too slow to reorganise themselves into a crystal structure. The random molecular structure gives glass its transparency. However, as a result, glass is also unable to deform plastically. Making it a brittle material that will yield suddenly without showing plastic behaviour. Furthermore, due to the lack of plastic behaviour, glass is unable to reduce local peak stresses (Oikonomopoulou, 2019).

The following sections will describe how the chemical composition of glass influences the properties of different glass types and the specific structural properties of glass.

3.1.1 Glass Compositions

The properties of glass depend on the chemical composition. The purest form of glass has a molecular structure completely consisting of silica oxide. The addition of other elements can change the behavior of the material. The material composition has the most pronounced effect on the temperature-related properties, such as thermal expansion, viscosity and working temperature. For example, the addition of alkali to pure silica oxide will lower the working temperature by 110 to 410 degrees Celsius. The most commonly used chemical compositions can be divided into six types as summed up in table 3.1.

Soda-lime glass is the most commonly produced type of glass. It is used for glazing, bottles and jars. Its composition causes a low working temperature, which results in low manufacturing costs. This makes soda-lime glass the most economical choice for a lot of applications. It, however, has a limited capacity to deal with quick temperature changes that can cause thermal shock and requires a longer annealing time than other compositions of glass like borosilicate glass (Oikonomopoulou, 2019).

Borosilicate glass is produced by the addition of boron oxide to the glass composition. This gives the glass a lower thermal expansion coefficient, which results in a high resistance to thermal shock. Making this glass type suitable for applications where termal changes are expected, like laboratory glassware as well as household ovenware. The lower thermal expansion coefficient also reduces the annealing time. This makes borosilicate glass type attractive for large scale castings. It is, however, more expensive than soda-lime or lead silicate glass.

Lead silicate glass has the lowest annealing point and is after soda-lime the most economical option. It is softer than the other glass types, making it easier to grind, shape and polish. However, this also makes it prone to scratching, limiting the usage in architectural applications and making it more suitable for art (Oikonomopoulou, 2019).

Aluminosilicate, 96% silica and fused-silica are glass types that are not favoured for structural components. Due to their composition, they require high temperatures for production and annealing. This greatly increases the manufacturing costs. These types of glass are more suited for specialised circumstances.

	Mean melting point at 10 Pa.s	Softening Point	Annealing Point	Strain Point	Density	Coefficient of Expansion 0C - 300C	Young's Modulud
Glass type	[°C]	[°C]	[°C]	[°C]	kg/m^3	10-6/°C	Gpa
Soda-lime	1350 - 1400	730	548	505	2460	8.5	69
Borosilicate	1450 - 1550	780	525	480	2230	3.4	63
Lead Silicate	1200 - 1300	626	435	395	2850	9.1	62
Aluminosilicate	1500 - 1600	915	715	670	2530	4.2	87
Fused-silica	>> 2000	1667	1140	1070	2200	0.55	69
96% silica	>> 2000	1500	910	820	2180	0.8	67

Table 3.1: Approximate properties of different glass types (Oikonomopoulou, 2019)

3.1.2 Structural Properties

The theoretical strength of glass can be calculated with Orowan's stress formula, and is estimated to be between 6000 and 10000 MPa (Haldimann, 2006). However, this strength is rarely achieved due to material flaws and geometric defects. These imperfections introduce stress concentrations that significantly lower the the structural performance of glass. As glass is unable to deform plastically, these stress concentrations can not be reduced by distributing them throughout the material.

These localized peak stresses can surpass the theoretical strength of glass and cause the glass to break (Oikonomopoulou, 2019). When subjected to a compression load, flaws and cracks within the glass can not grow because they are effectively being closed by elastic deformation. However, when subjected to tensile stress, these flaws and cracks are prone to propagation, ultimately leading to the failure of the glass component. This difference in behavior results in the higher compressive strength of glass compared to its tensile strength. In practice, glass rarely breaks solely due to exceeding its compression limit. Even under a comprehensive compression load, the emergence of tensile stresses is inevitable due to factors such as buckling or deformation (Damen, 2019).

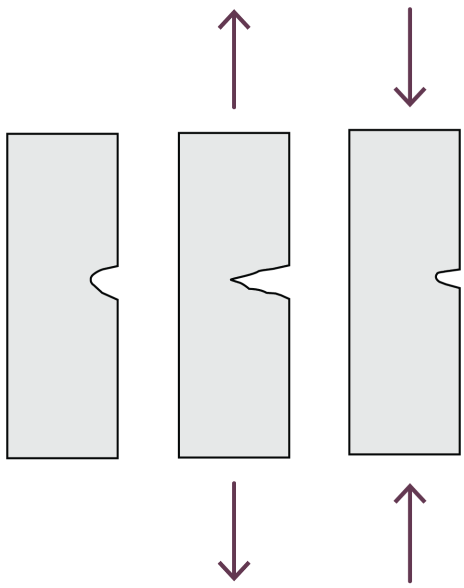


Figure 3.1: Structural behaviour of brittle material with a surface flaw.

There are two types of flaws in glass elements (figure 3.2). First there are inclusions, like bubbles stones or cords. These flaws will cause peak stresses inside the glass structure due to their different mechanical properties. More important for the strength of glass are the edge and surface flaws. These defects are usually the result of the machining process but also accumulate due to scratching, debris and thermal changes that the object will be subjected to during handling, transportation and use (Oikonomopoulou, 2019).

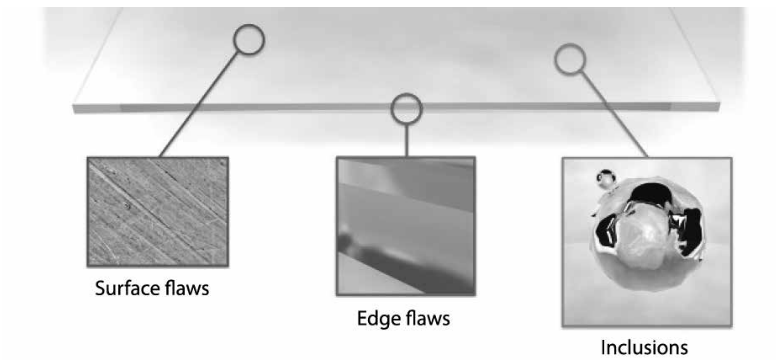


Figure 3.2: Types of flaws in glass elements

As cast glass elements have a larger volume and there is no automated control of flaws during the manufacturing process, it is not possible to directly use the characteristic values of borosilicate float glass. To extract the tensile strength of borosilicate cast glass, Koniari (2022) references the flexural strength of borosilicate glass that resulted from experiments done by Bristogianni et al. (2020). With the assumption that the flexural ans tensile strength are proportional for float and cast glass, the tensile strength of cast glass is determined by:

$$f_{t,cast} = \frac{f_{t,float}}{f_{fl,float}} f_{fl,cast}$$

Furthermore, to derive a safe value for the tensile strength of cast glass, Koniari (2022) references regulations for float glass structures. As there are currently no guidelines or norms specified for the design of cast glass structures. The German standard (DIN 18008) states that the tensile design strength of glass can be calculated by:

$$R_{dfloat} = \frac{K_{mod}\, k_c\, f_t}{\gamma_m}$$

where:
K_{mod} : Coefficient for consideration of the load duration of annealed float glass (0.4 for medium loads)
k_c : Coefficient for consideration of the type of construction (1.0 for horizontal float flass construction)
γ_m : Partial safety factor of resistance of the material (1.8 for float glass)

Table 3.2 shows an overview of the characteristic values of borosilicate float glass, the flexural strength of borosilicate cast glass as found by Bristogianni et al. (2020) and lastly the characteristic tensile strength and resulting design strength derived by Koniari (2022).

	Flexural strength (f_{fl})	Tensile strength (f_t)	Design tensile strength (f_c)	Compressive strength (f_c)
	[MPa]	[MPa]	[MPa]	[MPa]
Float glass	69	45	-	500
Cast glass	44	29	6.4	500

Table 3.2: Characteristic and design values for float and cast borosilicate glass (Bristogianni et al., 2020; Koniari, 2022; Oikonomopoulou, 2019)

3.2 Casting Process

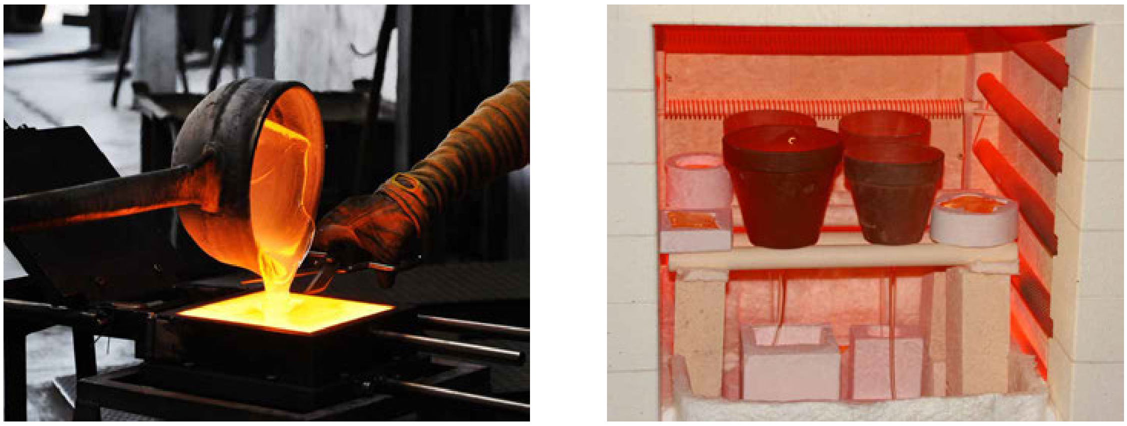
Casting glass allows for the production of complex three dimensional building components. The process of casting glass is similar to that of casting metals. First, the material is heated to a liquid state, after which it is poured into a mould of the required shape. The material is kept in the mould to cool down and solidify, after which it is released from the mould. Due to the properties of glass, additional steps have to be taken in the casting process. The following sections will describe the different casting methods and the annealing process that follows casting.

3.2.1 Casting Methods

Casting glass can be done either from raw materials (primary casting) or reheating solid pieces (secondary casting)(Oikonomopoulou, 2019). Both methods require different processes.

Primary casting is the process in which the glass melt is created from the raw ingredients in a furnace. The molten glass is then poured into a mould, after which it is placed back into a second furnace for the annealing process. Primary casting is the preferred method for large scale production.

Secondary casting is a process in which the glass melt is created from reheating solid glass pieces like sheets, grains, powder etc. With secondary casting the melting as well as the annealing happen within one kiln. The glass pieces are placed in a holder from which the glass can flow into the moulds when sufficiently heated. The use of already formulated solid pieces requires lower temperatures. This in combination with only requiring one furnace makes it the favoured method for customized components.



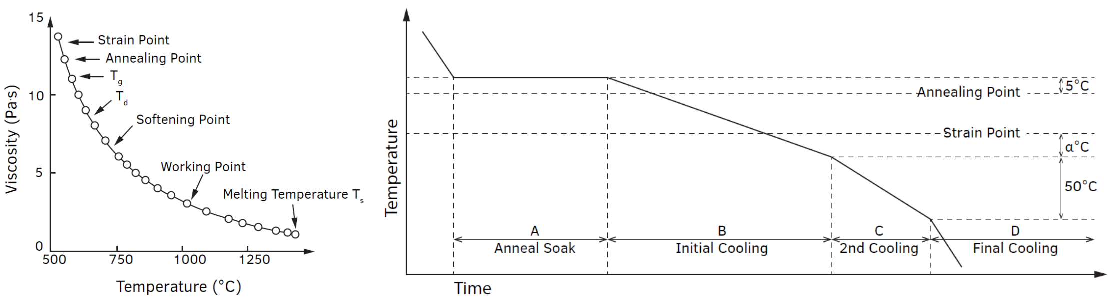
(a) Primary casting method (hot forming) **(b)** Secondary casting method (kiln-casting)

Figure 3.3: casting methods

3.2.2 Annealing Process

The annealing process of glass consists of keeping the glass melt withing a certain temperature range to eliminate strain and prevent the formation of residual stress during further cooling of the element. When glass is cast, either by primary or secondary casting, the temperature of the glass melt lies within its melting temperature and working point. Here the glass is viscous enough to flow into the mould (figure 3.4a). After the mould is filled a process called quenching is started. This is done by rapidly cooling the glass until it is below the softening point and preventing the formation of a crystal molecular arrangement. When the temperature of the glass melt has reached the softening point the annealing process starts. At this temperature, the glass has enough viscosity that the rearrangement of molecules can happen which in turn will relieve stress in the element. As Oikonomopoulou (2019) states, "At the annealing point stress relief can occur within a few minutes; whilst toward the strain point it requires a few hours" (p. 69).

After the element has a temperature below the strain point, cooling can happen faster. However, it still needs to be slow enough to limit the risk of thermal shock. Figure 3.4b shows the typical annealing schedule for soda-lime glass.



(a) **(b)**

Figure 3.4: (a) Typical curve for viscosity as a function of temperature for soda-lime-silica melt;(b) Typical annealing scheme for commercial soda-lime glasses.

The shape of an element has a direct correlation to the internal stresses that persist during annealing. The remaining stresses are largely caused by in-homogeneous shrinkage due to temperature differences in the glass. These differences can occur as a result of the amount of cooling surface, the type of glass and the thickness of the sections. This has led to a set of preferences for cast glass shapes (figure: 3.5). Uneven volume distribution should be avoided, as well as sharp and pointy edges. Round or ellipsoid shapes with equal mass distribution lead to a limited risk of residual stress (Damen, 2019).

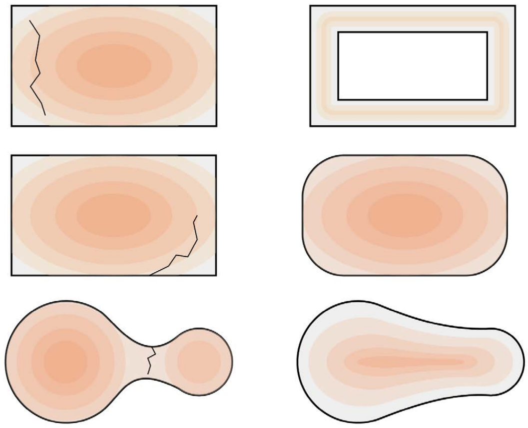


Figure 3.5: Three primary design principles to reduce annealing time and prevent internal stresses.

3.2.3 Approximating the annealing time

The time that elements need to properly anneal depends on a combination of factors. The element size has a considerable influence, but the thermal properties of the glass type and the geometry of the component also affect the annealing duration. There are formulas that use these aspects to approximate the annealing time. However, there are a multitude of factors that can not be taken into consideration in these formulas. For example, the annealing time is also influenced by the number of sides directly exposed and the presence of other elements in the furnace (Oikonomopoulou, 2019).

One of these formulas, as mentioned in (Shand & Armistead, 1958), has been adapted and used by Koniari (2022). This equation is considered the most suitable as it calculates the cooling rate h. Furthermore, it incorporates the material properties defined by M and a shape factor b. It should be mentioned that this formula was originally derived for the annealing in a lehr. This is a temperature-controlled kiln, in which the objects are moved through a temperature gradient. Objects annealed in a lehr are mostly cooled down from the top and a limited amount from the sides.

$$h = \frac{\sigma}{Md^2b} \qquad M = \frac{E\alpha_{ex}\rho c_p}{(1-\mu)\lambda}$$

Which when combined leads to:

$$h = \frac{\sigma}{\frac{E\alpha_{ex}\rho c_p}{(1-\mu)\lambda}\ d^2\ b}$$

In this equation, σ refers to the maximum allowable permanent stress in the object. The characteristic dimension that is calculated is given by d. Furthermore, E, α_{ex} and ρ refer to the Young's modulus, thermal expansion coefficient and the density of the material respectively. The other material properties that influence the annealing are the Poisson's ratio (μ), specific heat (c_p) and the thermal conductivity (λ).

3.3 Mould types

Glass mould types can be divided into single or multiple use moulds. Permanent moulds are moulds that can be used for a large number of castings. Disposable moulds are fabricated from brittle materials and are only suitable for one time use or small amounts of castings. The selection of a suitable mould for a project depends on factors such as the shape of the component, the desired production volume, the required precision of the final product and the type of glass used. The following sections will describe the different mould types.

3.3.1 Permanent Moulds

Permanent moulds are moulds that can be used for a large number of castings. Often times they are made from stainless steel, making them resistant to the high temperatures needed for casting glass. As a result, they are appropriate for the primary casting process. In addition, they result in a glossy surface as well as highly accurate products. However, permanent moulds are unsuitable for topologically optimized shapes as these moulds are expensive to manufacture especially when the shape becomes more complex (Oikonomopoulou, 2019).

3.3.2 Disposable Moulds

Disposable moulds are fabricated from brittle materials and are only suitable for one time use or small amounts of castings. This makes them most suitable for projects that require small batches and in kiln-casting processes. Due to the nature of the mould materials, the glass elements need to be post-process to get rid of any surface roughness imprinted on the glass by the mould. Disposable moulds can be made from a multitude of materials, each resulting in a different level of accuracy as well as higher or lower manufacturing costs. Silica-plaster moulds are relatively cheap but offer a low level of accuracy. Alumina-silica moulds, on the other hand, result in highly accurate castings, however this goes paired with a high manufacturing cost. 3D printed sand moulds offer a high level of accuracy, with a relatively low manufacturing costs.

Silica Plaster Moulds

Silica plaster moulds are fabricated with the lost wax technique. A positive wax model of the desired geometry is created, this is then used to create a refractory mould by dipping it or coating it in clay or another material. After solidifying, the wax is melted and drained away from the mould, leaving a cavity in the shape of the desired geometry. The mould is then used for casting, after which the mould is broken to release the casting.

While traditionally the wax model had to be created manually, advancements in additive manufacturing make it possible to use a wax-based filament to 3D print the model. This results in an increased accuracy as well as an easier production of complex shapes.

Limitations of the lost wax technique are linked to the properties of the silica plaster. Silica plaster has a low maximum firing temperature, making it only suitable for kiln casting. Furthermore, the process of creating the mould is laborious as it involves a lot of steps and the use of a oven at two times (Ioannidis, 2023).

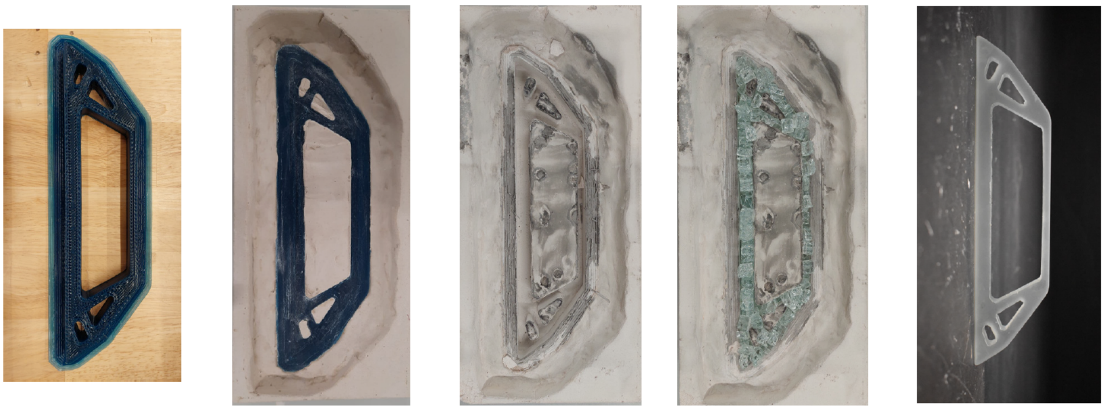


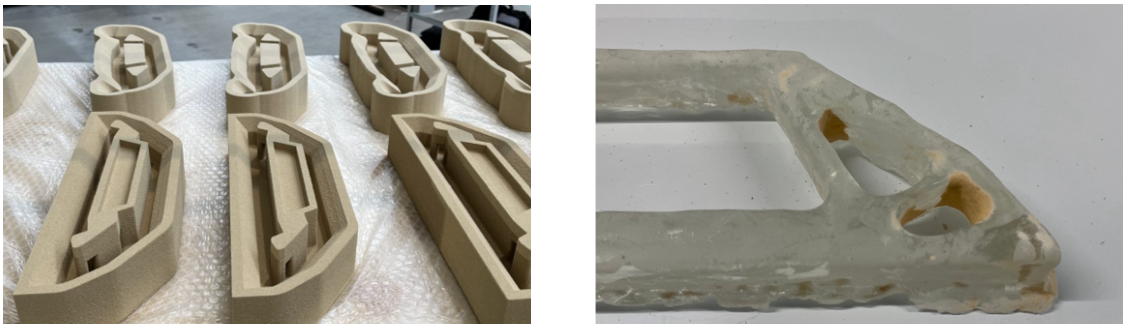
Figure 3.6: Steps of the lost wax technique, left to right: wax model; casting; de-waxing; glass placement; cast glass component

3D printed Sand Moulds

3D printed sand moulds (3DPSM) are fabricated using additive manufacturing. A digital 3D model of the desired geometry is sliced into layers. The mould is then produced by applying thin layers of sand on the printing table and having a computer-controlled head apply a liquid binder on each layer in the negative space of the geometry. The print is then dried and excess sand is removed.

Due to their relatively low manufacturing cost, high level of precision and low production time, 3DPSM are very promising for glass castings. However, 3DPSM have a size restrictions imposed by the manufacturing process. The largest printer, the voxeljetprinter VX4000, has a maximum printing dimension of 4x2x1m (Oikonomopoulou et al., 2020). When elements of a larger size are desired, the mould has to be divided into smaller parts that can be assembled into one mould (Koniari, 2022).

Furthermore, experiments done with these moulds also show other limitations. As the moulds need to withstand high temperatures during the casting and certain mould types leave a rough finish on the element. However, choosing the appropriate binders as well as coatings can ensure good results (Ioannidis, 2023; Oikonomopoulou et al., 2020).



(a) 3D printed sand moulds   **(b)** Component casted in 3DPSM

Figure 3.7: Example of glass castings with 3DPSM

3.3.3 Comparison Mould Types

For the casting of topologically optimized geometries, 3D-printed sand moulds are the best option. This follows out the comparison of the different mould types, as shown in table 3.3. Sand moulds offer high accuracy for a lower cost than the other mould types. Additionally, the sand can be dissolved in water and reused to print new moulds, increasing the sustainability (Bhatia, 2019).

Additional constraints are placed on any mould type used for the realization of cast glass elements due to the manufacturing process. Auxiliary paths need to be added to the mould design in order for air trapped in the mould to escape (Bhatia, 2019). Furthermore, the glass will cause pressure on the mould during the casting process. To withstand this force the mould needs to have sufficient thickness. This thickness is depended on the size of the voids in the structure. Thus a minimum void size is necessary to be able to create a sufficiently strong mould (Koniari, 2022)

(a)   **(b)**   **(c)**

Figure 3.8: (a) Glass TO node fabricated with the lost-wax technique; (b) Glass bricks with permanent precision moulds; (c) Part of TO glass column fabricated with 3D-printed sand mould

<table>
<tr><th>Characteristics</th><th colspan="8">Mould type</th></tr>
<tr><td>Reusability</td><td colspan="3">Disposable</td><td colspan="5">Permanent</td></tr>
<tr><td>Material</td><td>Silica plaster</td><td>Alumina silica</td><td>Sand</td><td colspan="3">(Stainless) Steel</td><td colspan="2">Graphite</td></tr>
<tr><td>Adjustability</td><td>Fixed</td><td>Fixed</td><td>Fixed</td><td>Adjustable</td><td>Fixed</td><td>Pressed</td><td>Adjustable</td><td>Fixed</td></tr>
<tr><td>Production method</td><td>Investment casting/ lost wax technique</td><td>Milling</td><td>3D printing</td><td colspan="3">Milling / Cutting and welding</td><td colspan="2">Milling / grinding</td></tr>
<tr><td>Manufacturing costs</td><td>Low</td><td>High</td><td>Low</td><td colspan="3">Moderate to high</td><td colspan="2">High</td></tr>
<tr><td>Top Temperature</td><td>900 - 1000 C</td><td>~1650 C</td><td>Unknown</td><td colspan="3">~1200 C / 1260 C</td><td>Unknown</td><td>Unknown</td></tr>
<tr><td>Glass annealing method</td><td colspan="3">Mould not removed for annealing</td><td colspan="3">Mould usually removed for annealing/ only maintained if high accuracy is required</td><td colspan="2">Mould not removed for annealing</td></tr>
<tr><td>Release method</td><td>Immerse in water</td><td>Water pressure</td><td>Unknown</td><td colspan="3">Release coating necessary</td><td colspan="2">Release coating necessary</td></tr>
<tr><td>Level of precision</td><td>Low / moderate</td><td>High</td><td>High</td><td>Moderate/high</td><td>High</td><td>Very high</td><td>Moderate / High</td><td>High</td></tr>
<tr><td>Finishing surface</td><td colspan="3">Translucent / rough</td><td colspan="3">Glossy</td><td colspan="2">Glossy with surface chills</td></tr>
<tr><td>Post-processing requirements</td><td colspan="3">Grinding and polishing required to restore transparency</td><td colspan="3">Minimum or no post-processing</td><td colspan="2">Minimum or moderate post-processing</td></tr>
<tr><td>Applicability</td><td colspan="3">Single component / low volume production</td><td colspan="3">High volume production</td><td colspan="2">High volume production</td></tr>
</table>

Table 3.3: Mould types and their properties. Reformatted; original from (Oikonomopoulou, 2019), edited by (Koniari, 2022)

3.4 Conclusion and Input Values

From the information mentioned in the previous sections, the specific input values and criteria for the algorithm can be determined. As mentioned in section 3.1 borosilicate glass has the best properties for structural elements. Thus it is preferred over soda-lime glass even though it is more expensive. The design principles derived from the annealing are translated into a maximum annealing time, element dimensions and ratio between the largest and smallest element. This all results in the values given in table 3.4 and 3.5.

Borosilicate glass			
	Symbol	Units	Input values
Young's modulus	E	GPa	70
Poisson's ratio	ν	-	0.2
Density	ρ	Kg/m^3	2500
Initial cooling range	ΔT	°C	530-460 (=70)
Thermal expansion coefficient	α_T	K	3.25×10^{-6}
Thermal conductivity	κ	W/(m*K)	1.15
Specific heat capacity	c_p	J/(Kg*K)	800

Table 3.4: Input values Borosilicate glass. Reformatted; original from (Koniari, 2022)

Borosilicate glass			
	Symbol	Units	Input values
Design tensile strength	$f_{t,des}$	MPa	6.4
Design compressive strength	f_c	MPa	500
Deflection	d	m	l/500
Maximum annealing time	$t_{ann,max}$	s	43200 (5 days)
Minimum element dimension	d_{min}	m	0.06
Ratio of maximum to minimum element dimension	r_{ann}	-	2
Maximum permanent residual stress (after annealing)	$\sigma_{res/max}$	MPa	1

Table 3.5: Hard criteria Borosilicate glass. Reformatted; original from (Koniari, 2022)

4
Structural Optimization

This chapter will explore different structural optimization methodologies. Structural optimization is the process in engineering and design that aims to find the best configuration of structural elements or material within a design domain given an objective and set of constraints. The primary goal of structural optimization is to create efficient designs that achieve an optimum between structural integrity, material usage and design constraints. The development of optimized structures is closely related to advancements in digital manufacturing technologies, which have enabled the realization of complex structures.

The first section will explore different structural optimization methods. The next section will give an in-depth exploration of different topology optimization methods. The following section will explain the finite element method, which is a tool to analyse complex engineering problems, like topology optimization. A set of different case studies and their respective characteristics will be explored. Lastly, different approaches to integrate structural and manufacturing constraints into the optimization process will be discussed.

4.1 Structural Optimization

Within the field of structural mechanics, finding the most optimal material distribution within a domain to transfer forces is a widely researched topic. Structural optimization problems can be categorized into three different types: size, shape and topology optimization.

Size optimization is the form of structural optimization where the problem domain is known and set, but the size of the components is unknown. An example of this is a truss structure in which the layout of the truss is known and the cross-sections of the elements are optimized. As can be seen in figure 4.1a.

Shape optimization is the form of structural optimization where the goal is to determine the optimal form or contour of the domain. An example of this is a plate in which a set amount of holes are present, but the boundary and shape of these holes are optimized. As can be seen in figure 4.1b.

Topology optimization is the broadest form of structural optimization. It involves a problem in which the number, location and shape of holes as well as the connectivity of the domain are determined. For example, a 2-dimensional domain is defined and topological changes are made by allowing the domain to have a thickness of zero at different locations. This gives the freedom to determine the number and shape of the holes. As can be seen in figure 4.1c (Bendsøe & Sigmund, 2004; Querin et al., 2017).

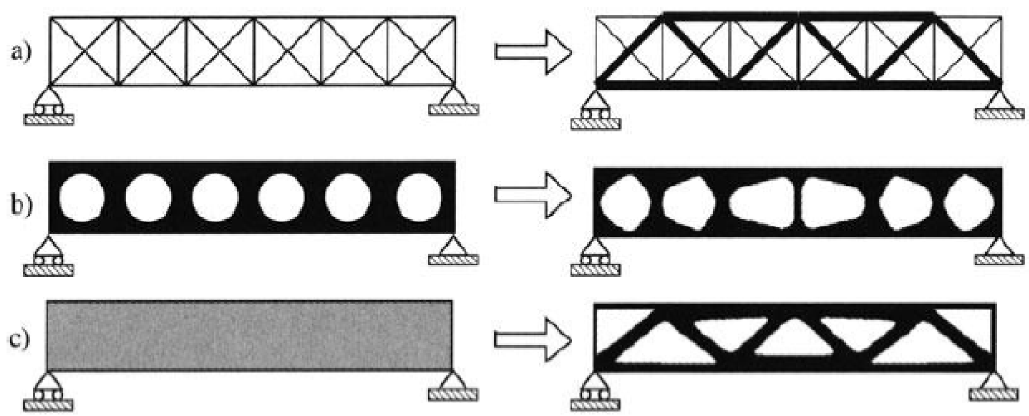


Figure 4.1: Overview types of structural optimization. a) Sizing optimization, b) shape optimization and c) topology optimization.

4.2 Topology Optimization Methods

Topology optimization methods can be divided into two categories; Optimality Criteria and Heuristic methods. Optimality Criteria are often based on the Kuhn-Tucker optimality condition and are more rigorous than heuristic methods. These methods are suitable for TO problems with a large set of design variables and minimal constraints. Heuristic methods are based on intuition or observations of engineering processes. Resulting in designs that are not guaranteed to be the optimal solution. However, they can still result in sufficiently optimal solutions (Querin et al., 2017).

While topology optimization are applied to continuous design domains, the most effective way of evaluating the structural performance of a design is a finite element analysis. Some of the topology optimization methods directly include the discretization of the domain in their formulation, as done in the Solid Isotropic Material with Penalization method (SIMP) method.

4.2.1 Optimality Criteria Methods

Optimality Criteria Methods are a class of optimization algorithms that are based on the idea of iteratively updating a set of variables to approach an optimal state while meeting specific design criteria (Querin et al., 2017). The updating of the variable set is guided by the sensitivity analysis, which quantifies how changes in the variable set affect the objective function. The first developments in the field of TO were based on the use of composite materials as an interpolation of void and full material. One such method is the homogenization approach (Bendsøe & Sigmund, 2004). These developments lay the basis to set up a framework for black-white structures like the Solid Isotropic Material with Penalization method (SIMP) and the level set methods.

Homogenization

The homogenization Method for TO is one of the first optimization solutions proposed by Bendsøe and Kikuchi (1988). The topology problem is converted to a shape optimization problem by dividing the domain into composite finite elements, consisting of void and full material (Xie & Steven, 1993) (image 4.2). Thus creating a porous structure in which the optimization problem becomes a sizing problem, finding the optimum dimensions for the microvoids (Querin et al., 2017).

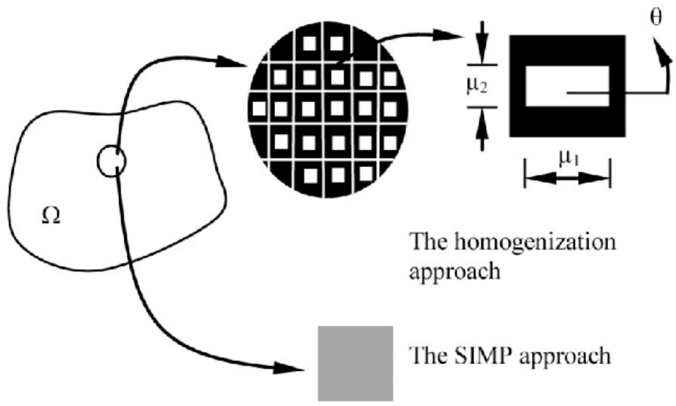


Figure 4.2: Schematic representation of homogenization and SIMP method

Solid Isotropic Material with Penalization (SIMP)

The SIMP Method follows up on the Homogenization method. Instead of having the finite elements being part solid part void, each element is assigned an artificial element density (ρ_e) that ranges between $0 < \rho_{min} \leq \rho_e \leq 1$ (Bendsøe & Sigmund, 2004). Where ρ_{min} is introduced as a very small value to avoid numerical issues.

The goal of the optimization process is to determine where a domain should retain or lose material. In a discrete form, this would end in a black-and-white representation of the geometry. Where the pixels are the finite element discretization. When using an artificial element density the optimization would result in grey areas of intermediate densities. To solve this and steer the calculation to a discrete solution a penalty (p) is introduced. The artificial density and the penalty are introduced into the optimization problem by the artificial elastic modulus (E_e) which is a product of the original material elastic modulus (E_e^0) multiplied by the density on which the penalty is placed. This penalty makes it more "expensive" to have elements of intermediate densities, leading the algorithm to a 0 - 1 solution for the given domain (Ω). A general formulation of the SIMP method is given by Bendsøe and Sigmund (2004) as:

$$E_e(x) = \rho(x)^p E_e^0, \qquad p > 1, \tag{4.1}$$

$$\int_\Omega \rho(x) d\Omega \leq V; \qquad 0 \leq \rho(x) \leq 1, \qquad x \in \Omega, \tag{4.2}$$

With the introduction of an artificial density certain issues arise. For example, how to interpret the intermediate values of the artificial density when the optimization has not fully converged to a 0-1 solution. The properties that these elements have in the model can only be realised with a composite material (Bendsøe & Sigmund, 2004).

A second issue that arises is the dependency of solutions on mesh discretization. The optimization will favour the introduction of a larger number of inner cavities as it results in a more efficient use of material. When the domain is divided into smaller elements it becomes more feasible to add these inner cavities, resulting in a different geometry than for discretization with larger elements (Bendsøe & Sigmund, 2004).

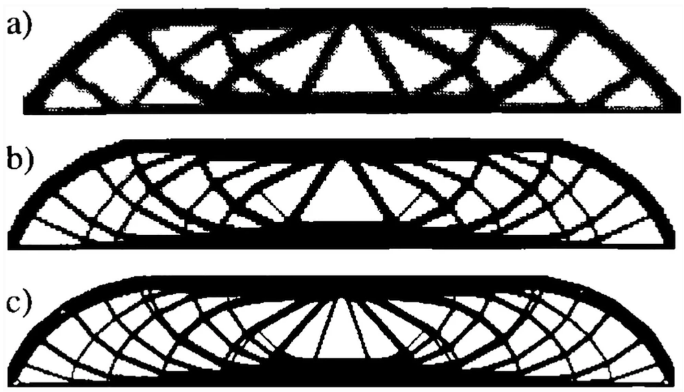


Figure 4.3: Mesh dependence in optimization, shown by optimization of MBB-beam with SIMP. (a) 2700; (b) 4800 and (c) 172000 elements

Furthermore, when using rectangular finite elements geometric singularities tend to emerge. The most important being checkerboards, point flexures and islanding. Checkerboarding is the occurrence of a checkerboard like pattern in the optimized geometry. Due to the nodal connectivity, this pattern falsely gives the impression of structural stiffness (Bendsøe & Sigmund, 2004). Point flexures are similar to checkerboarding, again solid elements are connected to each other in a single point (Yin & Ananthasuresh, 2003). Resulting in a geometry that is not feasible to produce and that has local peak stresses. Lastly, islanding is the occurrence of solid parts in the structure that are not connected to any other part of the structure (Kumar, 2017).

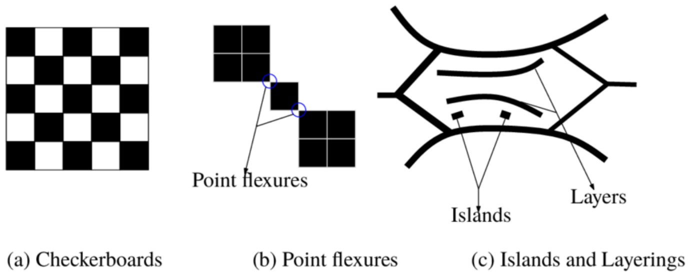


Figure 4.4: Geometric singularities. (a) Checkerboard pattern; (b) Point Flexures; (c) Layering & Islanding.

Finally, geometries obtained from optimization with the SIMP method often don't have a continuous boundary line. The boundary can consist of jagged edges and intermediate elements. As a result, the exact shape of the geometry is unclear (Bendsøe & Sigmund, 2004).

As the SIMP method has been further researched multiple solutions have been formulated for each of the previously explained issues. For example, either filtering can be applied (Bendsøe & Sigmund, 2004), or size constraints can be included (Fernández et al., 2020; J. Guest et al., 2004; J. K. Guest, 2009).

Level Set Method

The level set method was developed by Osher and Sethian (1988) for computing and analyzing the motion of an interface in two or three dimensions. When applying the level set method for topology optimization, the design boundary, between solid and void, is determined by the contour of the level set function($\varphi(x)$) at zero (Osher & Fedkiw, 2001). The level set function φ has the following properties (Osher & Fedkiw, 2001):

$$\begin{aligned}
&\varphi(x,t) > 0 \text{ for } x \in \Omega \\
&\varphi(x,t) < 0 \text{ for } x \notin \Omega \\
&\varphi(x,t) = 0 \text{ for } x \in \partial\Omega = \Gamma(t)
\end{aligned} \tag{4.3}$$

Indicating that any location x where the level set function has a value above 0 is part of the solid domain Ω. The contour Γ of the structure is defined by the locations x for which the level set function is equal to the zero level, and the structure is void wherever φ takes a negative value.

Level set approaches can be subdivided into two categories: the methods where the boundary can change but the number of holes stays constant and the methods in which new holes can be created (Sigmund & Maute, 2013).

For the first category, the most common updating scheme is that of the Hamilton-Jacobi equation. As the boundary of the domain is determined by the intersection between the cut-off level and the LSF graph, the domain can be changed by moving the cut-off level in the normal (vertical) direction. The speed of the vertical translation is a result of the velocity function (Osher & Fedkiw, 2001). Figure 4.5 shows an example of a two-dimensional design before and after a design update. (Here ϕ is used instead of φ)

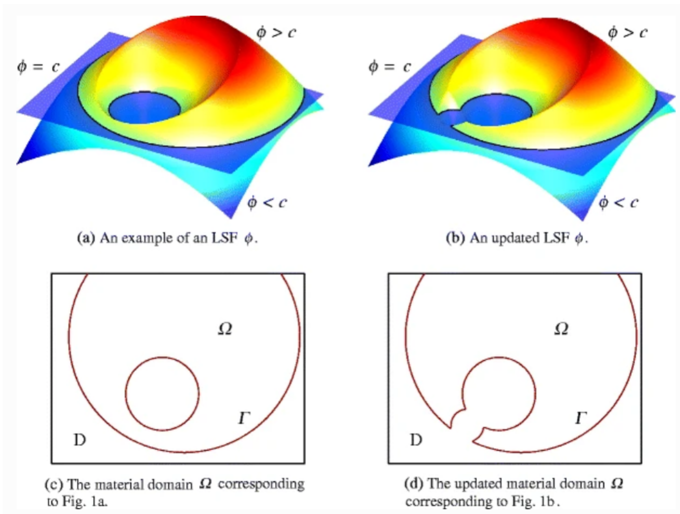


Figure 4.5: An example of the lsf and corresponding domain before and after a design update

4.2.2 Heuristic or Intuitive methods

Heuristic or intuitive methods are a class of optimization techniques that are based on intuition, observations of engineering processes, or observation of biological systems (Querin et al., 2017). These methods cannot always guarantee optimality as their outcome is often influenced by an initial guess. Still, they can provide viable efficient solutions. While there are many heuristic methods, only ESO and BESO will be explained, as they are the methods most commonly used.

ESO

Evolutionary structural optimization (ESO) was the first evolutionary algorithm for structural optimization, introduced by Xie and Steven (1993). The concept behind the ESO method is that by removing inefficient material from a domain, the structure will evolve to an optimum (Sigmund & Maute, 2013). The removal of material relies on a rejection criterion. This rejection criterion is generally based on the local stresses, but can also be defined in terms of displacement, buckling or thermal criteria (Querin et al., 2017). In the case of a stress rejection criterion, the elements that have low stresses are assumed to be inefficient and are thus removed. For this assessment, the stresses of all the individual elements (σ_e) are compared to the maximum stress (σ_{max}) in the whole structure (Querin et al., 2017). The elements that satisfy the following ESO inequality are then removed from the domain (Xie & Steven, 1997).

$$\sigma_e \leq RR_i * \sigma_{max} \tag{4.4}$$

Here RR_i is the rejection ratio of the ith steady state. The elimination of elements is repeated with the same RR_i value until no more elements are being removed from the domain. At this point a steady state is reached the rejection ratio is increased with an evolutionary rate (ER). This loop is repeated until either the maximum rejection ratio or a predetermined volume fraction is obtained (Xie & Steven, 1997).

BESO

Bi-directional evolutionary structural optimization (BESO) is an extension of ESO that works in two directions (Querin et al., 2017). Where in the ESO algorithm material can only be removed, the BESO algorithm allows for material to be placed where stresses are high. Both ESO and BESO have as a benefit that they result in a discrete division of the domain and no 'grey' areas (Sigmund & Maute, 2013). However, the BESO method can lead to mesh-dependent solutions. This means that for a similar design domain with the same loads, choosing a different mesh size can lead to different solutions. It has been observed that a finer finite element mesh will result in a larger amount of voids appearing (Huang & Xie, 2007). Introducing a perimeter control to the BESO method can lead to mesh-independent results. However, setting a correct value for the perimeter length constraint is difficult (Huang & Xie, 2007). A second issue with the BESO method is that it can give a non-convergent solution. Meaning that the solution gets worse in relation to the objective function if the algorithm is left to continue without stop (Sigmund & Maute, 2013).

4.2.3 Comparison Optimization Methods

In general, Optimality Corteria Methods are preferred above Heuristic methods, as they rely on mathematical functions for their updating scheme. This characteristic reduces the dependency on the initial guess, a limitation often encountered with methods such as ESO and BESO. The BESO method is improved compared to ESO as material can be subtracted as well as added to the design. Additionally, the BESO method can be adapted to a soft-kill approach as proposed by (Huang & Xie, 2007). This approach incorporates standard adjoined gradient analysis and filtering techniques similar to those used in the density approach in order to stabilize algorithms and results. Consequently, BESO operates as a discrete update variation of the conventional SIMP scheme (Sigmund & Maute, 2013).

However, as a result of the gradually imposed volume constraint the BESO method tends to result in topologies consisting of a large amount of thin bars. Making the shape and manufacturing more complex (Yago et al., 2022).

Level set methods have the benefit of resulting in a smooth surface, limiting the need for post-processing. However, they still strongly depend on the initial guess (Sigmund & Maute, 2013). Furthermore, they require more computational time and thus it is recommended to use a lower-level language such as C++ (Koniari, 2022).

For this project, the SIMP method is most suitable. While it still has some geometric defects, it leads to fast and robust solutions with good convergence. As it is not dependent on any starting guess, it is possible to result in a global minimum. With the outcome being mostly reliant on the objective function and constraints.

4.3 The Finite Element Method

The finite element method is a numerical method used for a wide range of engineering problems. It is used to analyze the behaviour of components and structures under various loads, boundary conditions and material properties. In topology optimization algorithms the finite element analysis (FEA) gives insight into the structural behaviour of the optimized geometry and is often used as the basis for defining the objective and constraint functions.

In general, a FE analysis starts with discretizing the problem domain into geometric shapes called elements. To solve for the approximate behaviour of the structure, the displacements of the individual elements are assumed in simple forms. A global stiffness matrix $[K]$ can be assembled by combining the equations of every element with their neighbouring elements. This assembly in combination with the boundary conditions makes it possible to solve for all the individual nodal displacements. Using these nodal displacements, the strains and stresses in the individual elements can be determined. (Chandrupatla & Belegundu, 2002; G. R. Liu & Quek, 2014).

The formulation of the element equations relies on the principle of minimum potential energy. In essence, this principle asserts that a body subjected to a force will deform to a stationary point where the total energy in the body is at a minimum. Specifically, the principle of minimum potential energy states that the total potential energy Π of an elastic body is equal to the total strain energy (U) and the work potential (wp). The work potential are any external forces working on the body, which can be traction forces (T), a body force (f) and/or loads acting on a single point (P_i). Thus, the potential energy for a general elastic body is given by:

$$\Pi = \frac{1}{2}\int_V \sigma^T \varepsilon dV - \int_V u^T f dV - \int_s u^T T dS - \sum_i u_i^T P_i \tag{4.5}$$

The stiffness matrix can be derived from the total strain (U). As the relation between stress (σ) and strain (ε) is given by $\sigma = E\varepsilon$, U can be rewritten as $\int_V \varepsilon^T E\varepsilon dV$. Introducing a matrix that translates strain to displacement $(\varepsilon = Bq)$ and for simplicity only considering nodal forces f, the stationary point is given by:

$$\frac{\partial \Pi}{\partial \{q\}} = \int_V [B]^T[E][B]dV\{q\} - \{f\} = 0 \tag{4.6}$$

$$\text{or as:}\quad \{f\} = [K]\{q\} \tag{4.7}$$

$$\text{where:}\quad [K] = \int_V [B]^T[E][B]dV \tag{4.8}$$

For a problem domain with known boundary conditions $\{f\}$, the stiffness matrix $[K]$ can be assembled from the element equations. Solving for the stationary point will result in the nodal displacements $\{q\}$. Important to note is that each finite element type has its respective equations that are dependent on the element's shape functions [N] (Chandrupatla & Belegundu, 2002).

Finite element analysis can be applied to either linear or nonlinear problems. In linear problems, it is assumed that the stresses are within the linear elasticity limit of the material. As a result, the stiffness matrix is constant during the computation. In non-linear problems, there are either large deformations or the material is stretched beyond its yield point resulting in plastic deformation. As a result, the stiffness matrix changes and adaptations need to be made in the calculation (Chandrupatla & Belegundu, 2002).

The division of the domain can be done with a broad range of elements. Figure 4.6 shows an example of different finite elements. For accurate analysis choosing a finite element that can give a valid representation of the structure is important (Chandrupatla & Belegundu, 2002; G. R. Liu & Quek, 2014; Wai et al., 2013). The next sections will cover the three main types of finite elements and their advantages and limitations.

4.3.1 One-dimensional elements

The simplest finite elements are one-dimensional line elements, which can be either straight or curved. One-dimensional elements can be subjected to any loading combination of tension (or compression), torsion and bending. The forces and moments working on any point (x) on a 1D element can be represented as the normal force $N(x)$, a shear force and bending moment around either axis of the cross-section $Q_z(x)$, $M_y b(x)$, $Q_y(x)$ and $M_z b(x)$ respectively, and lastly a torsional moment $M_t(x)$ which works around the axis of the body (Öchhsner & Merkel, 2018).

Stress analysis done with 1D elements will directly result in axial, bending, and combined axial and

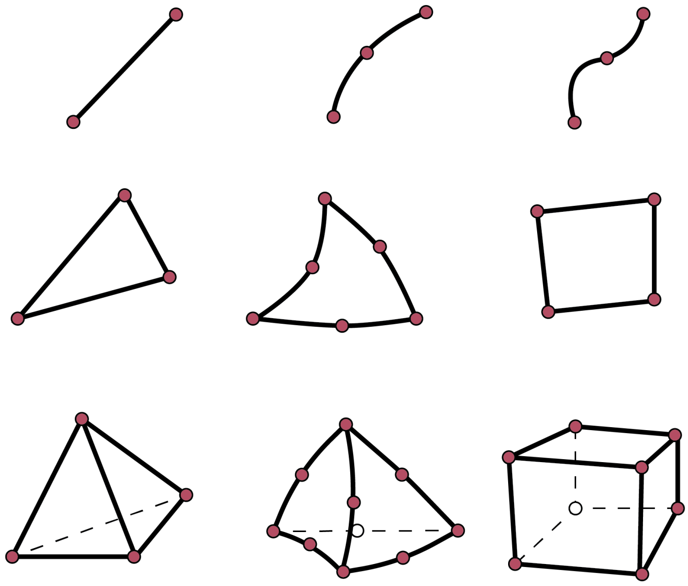


Figure 4.6: Different finite element types. Top row: one-dimensional elements. Second row; two-dimensional elements. Third row; tree-dimensional elements

bending stresses. Making the interpretation of the maximum stresses in the element relatively easy (Öchhsner & Merkel, 2018).

To relate the set of equations for a single bar element to plane or three-dimensional problems global and local coordinates systems need to be defined. The conversion from one system to the other is done with a transformation matrix T. With this conversion, one-dimensional elements become applicable to not only beams but also to (spacial) trusses (Chandrupatla & Belegundu, 2002).

4.3.2 Two-dimensional elements

Two-dimensional elements are surface elements. They usually have a triangular or quadrilateral base shape, with 3 or 4 nodes respectively. Isoparametric versions of these elements can be introduced when greater accuracy is needed. As these elements are planar the displacement vector is given by $\mathbf{u} = [u, v]^T$, where u and v refer to the x and y (Chandrupatla & Belegundu, 2002).
The stresses and strains in a two-dimensional problem are:

$$\sigma = [\sigma_x, \sigma_y, \tau_{xy}]^T \tag{4.9}$$
$$\varepsilon = [\varepsilon_x, \varepsilon_y, \gamma_{xy}]^T \tag{4.10}$$

Triangular elements are also called constant-strain triangles. The shape functions of these elements are linear and thus the strains and stresses are constant throughout the element. The calculated stress vector can be translated to principal stresses with Mohr's circle. In contrast, a quadrilateral element does not have constant stresses. A stress analysis usually results in four sets of stress values that are found at the Gauss points. Depending on the accuracy needed, it is possible to just evaluate the stress at one point. This will reduce the calculation time needed (Chandrupatla & Belegundu, 2002).

Two-dimensional elements are often used for problems that are governed by plane stress and plane strain like plates or shells. One advantage of analysis with 2D elements is that it will give information on the location of stress concentrations (Wai et al., 2013).

4.3.3 Three-dimensional elements

Three-dimensional elements are often times the most accurate representation of engineering problems, as those are usually three-dimensional. However, more accuracy also translates into more calculation time needed. Four-node tetrahedral or eight-node hexahedrons are most commonly used for discretizing a domain in 3D elements. Similar to the 2D elements, the tetrahedral has linear shape functions resulting in a constant strain and stress. As the division of a three-dimensional domain in tetrahedrons can be tedious, it is possible to make use of a master cube. This cube can be divided into five or six tetrahedrons. Similar to the assembly between elements, the element matrices of the hexahedron can be computed from the assembly of the tetrahedron matrices (G. R. Liu & Quek, 2014). As with the 2D elements, it is possible to generate Isoparametric versions of these elements if higher accuracy is needed.

The displacement vector of three-dimensional problems is given by $\mathbf{u} = [u, v, w]^T$, as the nodes are now free to move in any of three directions. Here u, v and w refer to the displacements in x, y and z respectively (Chandrupatla & Belegundu, 2002).

The stresses and strains in a three-dimensional problem are:

$$\sigma = [\sigma_x, \sigma_y, \sigma_x, \tau_{yz}, \tau_{xz}, \tau_{xy}]^T \tag{4.11}$$

$$\varepsilon = [\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{yz}, \gamma_{xz}, \gamma_{xy}]^T \tag{4.12}$$

The principal stresses can be derived from the (3 x 3) stress tensor with the stress invariants. This will be further addressed in subsection 4.7.3.

A Three-dimensional finite element analysis requires more computational time, so it is preferred to simplify structures to either 2D or 1D representations. However, when the analysis of a complex geometry is needed, 3D elements can give accurate deformations as well as insight into the existence of stress concentrations (Wai et al., 2013).

4.4 Formulation of the optimization problem

Boyd and Vandenberghe (2004) states that a generalized optimization problem has the form:

$$\text{minimize} \quad f_o(x) \tag{4.13}$$

$$\text{subject to} \quad f_i(x) \leq b_i, \quad i = 1, \ldots, m. \tag{4.14}$$

The variable that is optimized for is the vector $x = (x_1, \ldots, x_n)$. The objective function $f_0 : R^n \rightarrow R$ is the function that is minimized for and $f_i : R^n \rightarrow R, i = 1, \ldots, m$, are the constraint functions with $b_1, \ldots, b_m$ being the limits for the constraints. These constraints can refer to a wide range of criteria and must be satisfied for a design to be acceptable. The objective function used for structural optimizations often falls under one of the following three categories; compliance-based, stress-based or volume-based [1].

4.4.1 Compliance-based

A minimum compliance design maximized the global stiffness. When adding a volume constraint, the optimization will try to find the best way to distribute the material to get the stiffest structure possible (Bendsøe & Sigmund, 2004). A general form of a minimum compliance objective with a volume constraint is (Sigmund, 2001):

$$\begin{aligned} \min_x : \ & c(x) = U^T K U \\ \text{subject to} : \ & \frac{V(x)}{V_0} = f \\ : \ & KU = F \\ : \ & 0 < x_{min} \leq x \leq 1 \end{aligned} \tag{4.15}$$

[1] The formulations given in the next subsections are based on the SIMP methodology. Here the vector x represents the density of the finite elements. The formulations of the objective function can differ per optimization method while optimizing for the same goal. Specific formulations can be found in papers regarding that optimization method.

Here x represents the design variable. In the case of a SIMP calculation, this refers to the element densities. The value of x is bounded by x_{min} and 1. Where x_{min} is a very small fraction, introduced to avoid singularities in the calculation. Throughout the calculation, the structure must fulfil the static equilibrium given by the global stiffness matrix K, global displacement U, and the global force vector F. Furthermore, $V(x)$ is the current volume fraction, V is the volume of the total domain and f is the prescribed volume fraction.

Compliance-based optimization come with certain drawbacks. Firstly, the prescribed volume fraction has a large effect on the outcome of the structure. As it is set by the designer, an as stiff or stiffer structure with less mass could possibly exist (Shimels et al., 2017). Secondly, the general form of the compliance calculation does not take into account peak stresses, which could lead to the failure of a structure (Collet et al., 2017).

4.4.2 Stress-based

A second way of formulating the optimization problem is with a stressed-based objective. In contrast to the minimum compliance, here the algorithm aims to minimize the stresses in the structure under a volume constraint. The standard formula is (Yang & Chen, 1996):

$$\begin{aligned} \min : G(x) \\ \text{subject to} : \int_{\Omega} x d\Omega \leq M_0 \end{aligned} \tag{4.16}$$

Where:
$G(x)$: Global stress function for von Mises Stresses
M_0 : Volume that is to be maintained

Yang and Chen (1996) conclude that a minimum stress optimization gives a significantly different result than a compliance-based objective. The results demonstrate that it is possible to minimize peak stresses, but this may lead to a decline in structural stiffness. Combining both stress and compliance within a single objective function can prevent this issue, leading to improved results and faster convergence, as affirmed by (Yang & Chen, 1996).

Furthermore, stress-based optimization presents additional challenges beyond the decline in structural stiffness. Firstly, stress is a very local quantity. As a result, every element has to be checked individually to effectively control peak stresses. This significantly increased the computational time required. One approach to addressing this issue is the introduction of a uniform stress constraint that approximates the maximum stress value. However, such a uniform stress constraint can fail in indicating local peak stresses (Le et al., 2010).

A second issue is that elements with a very low density wrongly seem to have non-zero stress values, resulting in the so-called 'singularity' problem. Solutions to this problem include the introduction of an ε relaxation parameter (Cheng & Guo, 1997), or the qp approach as proposed by Bruggi (2008).

4.4.3 Volume-based

The last option for the objective is a volume-based objective. Here the volume is minimized under a constraint for the structural performance. For a volume-based objective, it is usual to both check for compliance as well as evaluating the stresses. Here the compliance constraint ensures a stiff structure, whereas the stress constraint guarantees the feasibility of the final outcome (Bruggi & Duysinx, 2012). A general way of formulating a volume-based optimization is given by (Bruggi & Duysinx, 2012):

$$
\begin{aligned}
\min_x \ &: V(x) = \sum_N x_e v_e \\
\text{subject to} \ &: KU = f \\
&: \frac{c}{c_L} \leq 1 \\
&: x_e^{(p-q)} \frac{\sigma_e}{\sigma_{Lt}} \leq 1, \quad e = 1, \ldots, N \\
&: 0 < x_{min} \leq x \leq 1
\end{aligned} \tag{4.17}
$$

Where v_e is the volume of each element; c is the compliance of the structure and c_L is the minimum compliance. The stresses are again evaluated in each finite element, resulting in a robust but computationally expensive calculation. It is possible to change this constraint to a global constraint. As mentioned before however, that does not assure that all peak stresses will be avoided.

A second option to limit the calculation time is to use a material failure criterion rather than evaluating the absolute values of stresses in each element. For materials that have the same strength in tension and compression, the Von Misses criterion can be used. For materials that have a difference in tensile and compressive strength, the Drucker-Prager failure criterion offers an alternative to the Von Mises criterion (Bruggi & Duysinx, 2012).

4.5 Related projects regarding Topology Optimization for brittle materials

Several projects have researched the possibility of applying TO for structural elements, with a predominant focus on (reinforced) concrete. The liquid phase of concrete makes it easy to form into complex shapes using either additive manufacturing methods or form works (Jewett & Carstensen, 2019). Projects focusing on TO in combination with glass have predominantly been carried out by students at the TU Delft. This section will focus on several of these projects, both in glass and concrete. The aim is to summarize the main conclusions of these projects and identify points that need to be taken into consideration for future research.

4.5.1 Topologically Optimized Concrete Slabs, ETH Zurich

A research group at the ETH in Zurich designed and fabricated two topologically optimized concrete slabs. The two designs were generated with different topology optimization tools. This research is part of a wider research exploring the potential of additive manufacturing within architecture. To create complex geometries a new stay-in-place 3D printed formwork was introduced for the casting of concrete. This form work is produced from 3D printing sandstone. To demonstrate the feasibility of the construction method two large-scale prototypes were fabricated (Meibodi et al., 2017).

Prototype "A" was designed using a Rhinoceros and Grasshopper plugin called Millipede. Millipede is a two-dimensional evolutionary algorithm that produces a grey-scale bitmap representing the material distribution. As the objective of the optimization, the deformation was minimized, while having a volume constraint of 0.2 of the initial volume. The grey-scale bitmap generated by the calculation was converted to a three-dimensional ribbed topology.

Prototype "B" was designed using the SIMP optimization algorithm present in the SIMULIA Abaqus software package. Here the objective function was minimizing the stress in the slab, with a volume constraint set to 0.18 of the initial volume (Jipa et al., 2016).

Seeing as this research mainly aimed to look at the feasibility of a production method catered to concrete, not all the conclusions drawn are applicable to the topological optimization of glass. For example, a stay-in-place formwork would counteract the transparency that is desired when working with glass. However, what can be learned from this research is the difference in shape resulting from different 3D optimization methods. As can be seen from prototype "A", it is possible to create a three-dimensional object from a two-dimensional optimization. This results in a relief-like object with

(a) prototype "A"

(b) prototype "B"

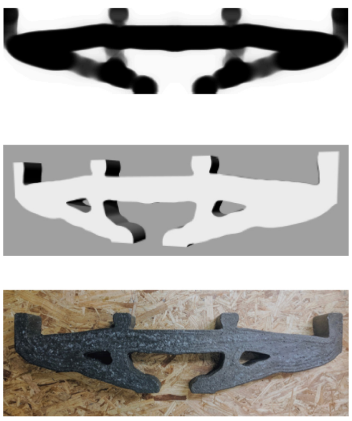
Figure 4.7: Topologically Optimized Concrete slabs designed by ETH Zurich

a change of depth in only one direction. Prototype "B" however has a more truss-like structure. It is not possible here to compare both methodologies in material efficiency and compliance, as the support conditions for which the designs were optimized differ for both slabs.

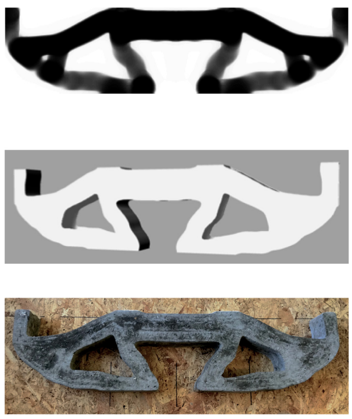
4.5.2 Topologically Optimized Plain Concrete Beams, Jewett

An example of a structure purely made with concrete, without any reinforcement, are the topologically optimized concrete beams by Jewett and Carstensen (2019). Here the focus lay on designing for the concrete phase. Three design cases are compared: Compliance, High tension and Low Tension. In the compliance design, the structural stiffness is maximized with a volume constrained and no limits on the stress fields. For the High and Low Tension design, the volume is minimized with a constraint on the stresses. To differentiate between the compressive and tensile stress of the concrete, the Drucker-Prager stress condition was used. After testing of the designs, it could be concluded that the specimens appeared to behave elastically at the design load. This gives reason to believe that this way of topologically designed concrete results in models that behave similarly as in reality. Additionally, the research shows that the beam designed for compliance has a higher structural stiffness than the High tension. Seeing as the stress-based cases have no incentive to create a stiff design, this is as expected. Furthermore, it was concluded that the need for significant heuristic post-processing had a negative effect on the performance of the design. Resulting in a recommendation to implement manufacturing and construction constraints into consideration in the topology optimization.

(a) Minimum compliance design



(b) High Tension design



(c) Low Tension design

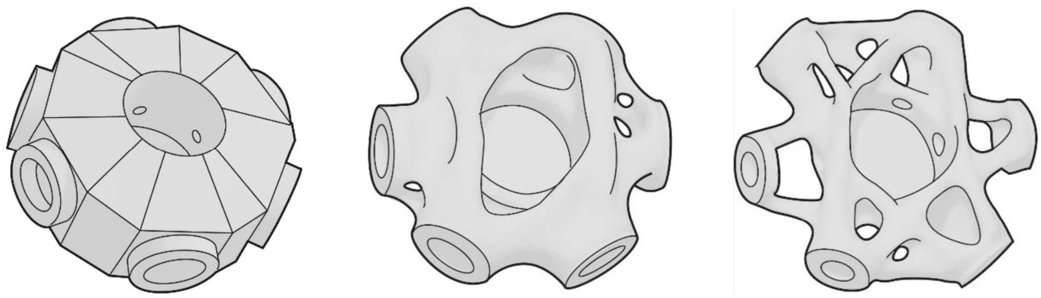
Figure 4.8: Topologically Optimized Concrete beams showing the topology optimized design (a), as built (b) and post processed(c)

4.5.3 Topologically Optimized Cast Glass Grid Shell Nodes, Damen

The thesis of Damen (2019) focuses on the topological optimization of cast glass grid shell nodes. The design of the nodes followed after the overall shape of the shell was created with the structural plug-in Karamba. From here the un-optimised base geometry is generated, as well as the loads acting on each node. A selected node was then optimized with a compliance-based TO algorithm, using the SIMP method, in the software ANSYS. As this software does not allow for the brittle nature of glass, it leaves the compressive strength of glass underutilized while requiring a post-analysis to check that none of the stresses exceed the allowable limits.

This thesis pointed out the design criteria that need to be taken into account for the TO of cast glass. These design criteria are subdivided into criteria for fabrication and structural performance. For fabrication, the focus lies mainly on designing for annealing. Where the mass should be minimised, and element thickness, as well as mass distribution, need to be uniform throughout the element. This results in more homogeneous cooling and thus limits the stresses in the component. The thesis concluded that topology optimization can lower the annealing time significantly.

One of the issues pointed out by this thesis is that topology optimization results in a highly efficient shape for the given load case. However, this also means if a structure is subjected to multiple changing load cases, optimizing it for just one will cause it to fail if it is subjected to other load cases. This is a crucial issue with TO as within the built environment it is reasonable to assume that loads can change from the design load. To tackle this issue it is suggested to add dummy loads to the optimization process. However, the thesis also points out that when multiple different load cases are applied, it is possible that the optimization will prioritize the heavier load case, leaving the parts carrying the lighter load under-dimensioned. Concluding in the recommendation to use TO for design challenges with a limited amount of load cases.



(a) Unoptimised node **(b)** Optimization TO3 **(c)** Optimization TO2

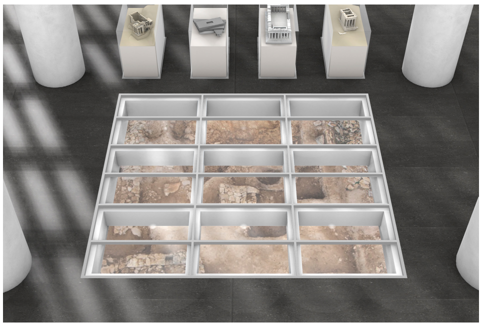
Figure 4.9: The full domain of the node, with optimized examples for two load case scenarios

For the prototype of the node, two different types of moulds were used. One was made with the lost-wax investment technique. Because of the complicated optimized shape the sacrificial wax element was made through Fused Deposition Modelling. The second technique tested was a 3D-printed sand mould. The sand-printed mould has benefits over the geometry generated with the FDM process. The FDM process results in an element with a rough, layered texture while sand moulds result in a smoother surface. A second benefit is that sand moulds require fewer production steps, as it allows for direct printing of the mould. However, more research into sand moulds is needed to determine the correct binders as well as the possible use of coatings to further improve the surface quality.

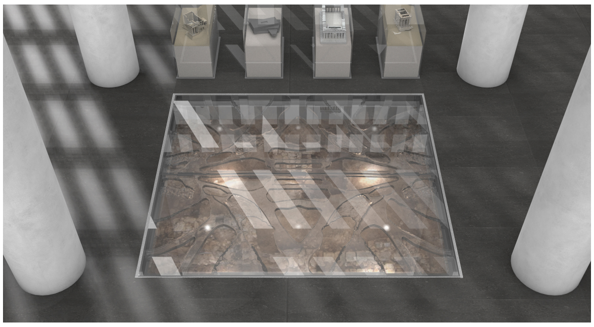
4.5.4 Mass-Optimized Massive Cast Glass Slab, Stefanaki

The thesis ofStefanaki (2020) focuses on the topological optimization of a cast glass slab. A similar approach was taken as in Damen (2019). The slab was optimized in ANSYS with the SIMP methodology. The compliance was minimized with constraints for the manufacturing, transportation and annealing as mentioned by Damen (2019). In addition to the established design principles, this case study required a flat surface which was also integrated into the optimization.

In line with the previous research, it is again stated that optimization with the principal stresses would



(a) Render Current Glass Floor



(b) Redesigned Glass Floor by Stefanaki (2020)

Figure 4.10: Glass floor in the Acropolis Museum

have given a better result. However, here the same issue arises as mentioned before that the constantly changing reference plane of the elements would make this very complex.

4.5.5 Just Glass TO Pedestrian Bridge, Koniari

The thesis of Koniari (2022) is the most recent thesis done at the TU Delft regarding the design of glass giants. This thesis tackled the limitations on TO glass structures set by the use of commercial software. To do so, a customized optimization tool was created. Matlab was used to create an algorithm that can take the specific structural constraints of glass into consideration and additionally incorporates the limitations set by the manufacturing process. For the optimization process, a SIMP updating scheme was chosen. The domain was discretized with 4-node two-dimensional quadrilateral finite elements. Two different problem formulations were tested out, one with a compliance objective and the second with a volume objective. This was done to see if the volume objective optimization could be a good alternative to the classical compliance approach.

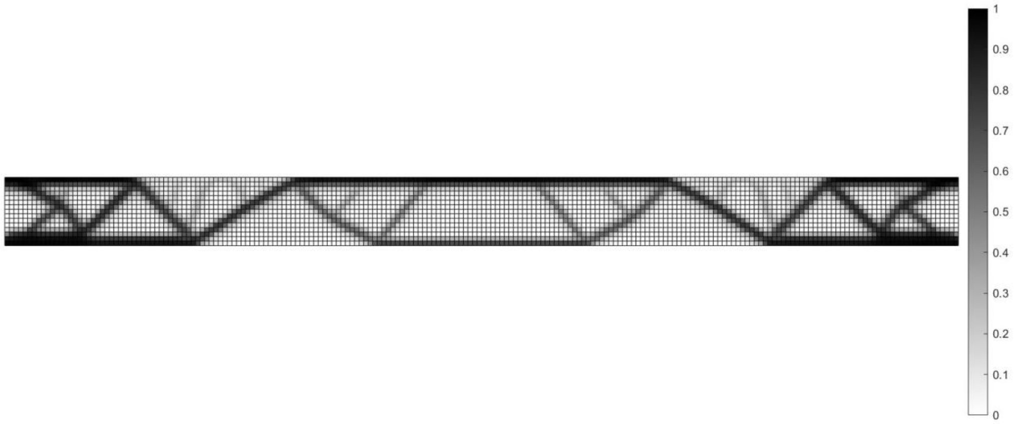


Figure 4.11: Resulting shape for Volume Objective & compliance, deflection, annealing and Drucker-Prager constraint (soda lime glass)

To test the performance of the algorithm, each objective and constraint is individually applied to a smaller-scale benchmark problem. Afterwards, both problem formulations were tested and applied to the case study of a small pedestrian bridge in the British Museum.

It was concluded that the volume-based problem formulation offers a robust solution which is similar to the result obtained with the compliance-based formulation. For the final design, the result of the volume-based optimization was implemented, as it had a clearer and sharper boundary.

When compared to commercial software, the algorithm has the main benefit that it directly gives a feasible geometry in terms of peak stresses. Furthermore, it allows for more specific and efficient outcomes due to the incorporation of the glass properties. However, a post-processing step is still necessary to convert the range of densities and jagged boundaries into a shape that can be fabricated.

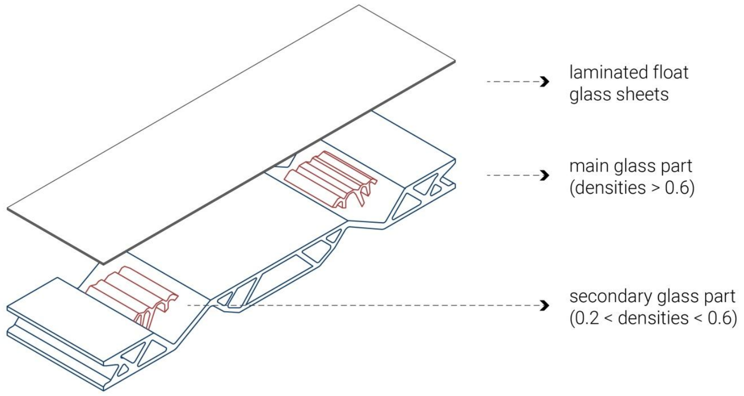


Figure 4.12: Diagram of final TO geometry

4.6 Topology Optimization Algorithms Matlab

Matlab is a high-level programming language that is primarily used for numerical computation, data analysis and visualization. There are numerous algorithms that provide insight into topology optimization methods and finite element analysis using Matlab. The following sections will elaborate on the relevant algorithms for the purpose of this thesis.

4.6.1 99-line for two-dimensional topology optimization

The 99-line topology optimization code written in Matlab is created by Sigmund (2001). It is a two-dimensional optimization algorithm for compliance minimization of statically loaded structures. The optimization uses the SIMP methodology and the Optimality Criteria (OC) method for solving the optimization problem. A mesh-independency filter is implemented to ensure checker boarding does not occur. Furthermore, the code introduces a symmetric boundary condition, to just calculate half of the structure. The program uses nested loops to assemble the stiffness matrix and prepare the filtering. As a result, this code is readable but slow when increasing the problem size (K. Liu & Tovar, 2014).

4.6.2 88-line for two-dimensional topology optimization

The 88-line topology optimization code has been developed by Andreassen et al. (2011) using the 99-line code of Sigmund (2001) as a basis. The program is coded for a compliance minimization of a statically loaded two-dimensional structure. A considerable improvement in calculation time has been made by more optimally utilizing Matlab. This is done by loop vectorization and memory preallocation. Vectorization is a technique to optimize the execution of operations on arrays or collections of data. Rather than processing each element individually, operations are executed on entire arrays or vectors simultaneously. This avoids the use of costly for and while loops. By implementing memory preallocation MATLAB reserves memory for a data structure, before it populates it with data. This avoids the performance issues caused by dynamic resizing of arrays and repeatedly reallocating memory. Compared to the 99-line code the 88-line program is restructured, parts of the code are moved out of the optimization loop to only be calculated once. Due to these improvements, the 88-line algorithm has a substantial increase in efficiency. For a benchmark problem of 7500 elements, the computational time has been reduced by a factor of 100.

4.6.3 Two-dimensional topology optimization algorithm for cast glass structures

The two-dimensional topology optimization algorithm for cast glass structures has been developed by Koniari (2022). It employs the SIMP methodology for both a compliance and volume minimization with specific constraints for the design of cast glass components. The algorithm utilizes the Fmincon function included in the optimization toolbox in Matlab. This is a function for finding the minimum of a constrained nonlinear multi-variable function. Multiple algorithms can be chosen for solving the optimization problem, here the standard interior-point algorithm is used. The algorithm includes glass-specific constraints as a principal stress constraint and a maximum member size constraint to ensure the annealing time is within a given limit. Similar to the 88-line algorithm, the structure of the code follows the logic of pre-assembling data to avoid recalculation in the optimization loop. However, the assembly of the stiffness matrix and calculation of the principal stresses are done using nested loops. Which as explained in the previous section, is linked to an increase in computational time.

4.6.4 169-line 3D topology optimization algorithm (top3d)

The 169-line three-dimensional topology optimization algorithm has been developed by K. Liu and Tovar (2014) and includes a finite element analysis, sensitivity analysis, density filter and plotting function for displaying the results. The algorithm solves minimum compliance problems for rectangular domains using a density-based approach with a modified SIMP interpolation. The gradient-based optimality criterion method is implemented as the default optimization algorithm. For the problem formulation, the partial derivatives of the compliance objective and volume constraints are included in the algorithm. Additionally, the algorithm can be altered to implement non-linear programming strategies such as SQP and MMA. The finite element discretization makes use of uniformly sized eight-node hexahedral elements. While the code uses a density filter, options are given to include a sensitivity filter and grey scale filter. The algorithm incorporates the same principles as 88-line for the efficient calculation of the optimization problem. In addition, an iterative solver is provided to handle large models. For solving large-scale models, the iterative solver is shown to be about 30% faster than a traditional direct solver.

4.7 Constraints

The manufacturing process of an object can give certain limitations on the size and shape of the element. One important criterion given by the manufacturing process is a constraint on the size of members and gaps between the members. A second is the avoidance of complete internal voids. Different approaches to integrate these limitations in the optimization process have been created and will be explained in the following two sections.

4.7.1 Filters

As mentioned in previous sections, filters are implemented to avoid numerical instabilities like checkerboard patterns, point flexures and islanding. Furthermore, filters can reduce mesh-dependent solutions. The application of a filter can be done at either the sensitivities, densities or both. The following sections will shortly explain a range of the most common filtering methods applied in topology optimization algorithms. An extensive description of a broader range of filters is given by Sigmund (2007).

Sensitivity Filter

The sensitivity filter works by calculating the sensitivities in the standard way, and modifying them as weighted averages of the sensitivities within a given neighbourhood (Sigmund, 2001; Sigmund, 1994). This is done using the following equation:

$$\frac{\partial c(x)}{\partial x_i} = \frac{1}{\max(\gamma, x_i)\sum_{j\in N_i} H_{ij}} \sum_{j\in N_i} H_{ij} x_j \frac{\partial c(x)}{\partial x_j} \tag{4.18}$$

Here $\gamma(=10^{-3})$ to avoid division by zero, H_{ij} is a weight factor given by $H_{ij} = r_{min} - \text{dist}(i,j)$. The distance is measured between the centre of element i and the centre of element j. The weight factor is zero outside of the area determined by r_{min}. The filter is implemented by using the filtered sensitivities for the Optimality Criteria update.

Density Filter

With a density filter, each element's density is redefined as a weighted average of the densities in a given neighbourhood N_i of the element x_i (Bruns & Tortorelli, 2001). The filtered density is given by:

$$\tilde{x_e} = \frac{\sum_{j\in N_e} w(x_e) v_j x_j}{\sum_{j\in N_e} w(x_e) v_j} \tag{4.19}$$

Here $w(x_e)$ is a linear weight factor given by $w(x_e) = R - \text{dist}(e, j)$. The operator dist$(e, j)$ gives the distance between the centre of element e and the centre of element j, and R is the radius size of the neighbourhood. This weight factor decreases linearly over the distance of the centroid to r_{min}. The original densities x_i are replaced by the filtered densities $\tilde{x_e}$, also referred to as the physical densities (Sigmund, 2007).

Density filtering with a Heaviside step function

The density filter can be modified with a Heaviside function in order to further increase the convergence to a 0-1 design. Introducing a Heaviside function will ensure that the density elements with a filtered density higher than 0 will become 1 and only if the filtered density is 0 the physical density will be 0 (J. Guest et al., 2004).

$$\overline{x_e} = 1 - e^{-\beta \tilde{x_e}} + \tilde{x_e} e^{-\beta} \tag{4.20}$$

For β set to zero, this filter is exactly the same as the original density filter. When β approaches infinity, the density value of the centre pixel is set to one if any of the pixels within the neighbourhood are larger than zero. For stable convergence, the filter has to be employed by gradually increasing the value of β from 1 to 500 (Sigmund, 2007).

4.7.2 Manufacturing Constraints

Manufacturing criteria can be integrated into the constraint function of a topology optimization to ensure a feasible design. As explained in chapter .. when designing cast glass structures it is important to avoid any large changes in cross-section size. Furthermore, the maximum cross-section size should be limited to ensure a feasible annealing time. For the structural integrity of the mould, the thickness of the mould needs to have a minimum size. And lastly, complete internal voids need to be avoided. The following sections will give mathematical implementations of these different constraints.

Maximum member size

A maximum member size constraint ensures that the diameter of all structural members is less than d_{max}. There are two approaches to limiting the maximum member size. The first is described by J. K. Guest (2009) and has a similar basis as the density filter. A circular domain (or spherical in the case of 3D) Ω_r is passed over the entire design domain and it is checked that this region is never completely filled with solid material.

$$\int_{\Omega_r(y)} (\tilde{x}) d\Omega < \int_{\Omega_r(y)} d\Omega \quad \forall y \in \Omega \tag{4.21}$$

This method can be implemented in continuous topology optimization problems, therefore the constraint should be able to penalizing intermediate densities, to prevent the constraint from counteracting existing penalization techniques employed in the SIMP method. For easier implementation of penalizing the intermediate volume fraction J. K. Guest (2009) proposes reformulating the constraint to require a minimum volume of voids to be present in the region Ω_r.

$$V_v^e(\rho_e) \geq V_{min}^e \tag{4.22}$$

Here V_{min}^e is the minimum required volume of voids, and $V_v^e(\rho_e)$ is the amount of void present in Ω_r computed by:

$$V_v^e(\rho_e) = \sum_{i \in R^e} \nu^i (1 - p^i + p_{min}^e)^n \tag{4.23}$$

Minimum Gap size

For the fabrication of the mould it is important that a minimum gap size is ensured between structural members to ensure enough wall thickness. For the implementation of a minimum gap size constraint Fernández et al. (2020) proposes to define a new domain (Ψ), that is slightly bigger than Ω_r and introducing a void fraction (ε) for a local region in Ψ. By finding the geometric relationship between the minimal distance (h), ε and Ψ the volume fraction can be used to impose a minimal h (image 4.13). Adding this minimum gap constraint to the optimization results indirectly in the limitation of the number of cavities and increases the size of the gaps remaining. Making an easier to produce geometry. In 3D, the following equation can be used to solve for ε Fernández et al. (2020):

$$\varepsilon 4r_o^3 = 3r_o(h + rr_{min)^2} - (h + 2r_{min})^3 - 12hr_oR_{min}^3 + 8r_m^3in \tag{4.24}$$

Where r_o can be found with:

$$r_o = r_{min} + r_{max} + h/2 \tag{4.25}$$

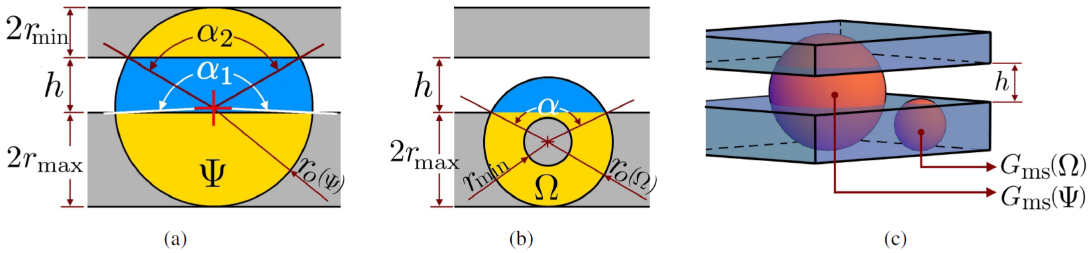


Figure 4.13: (a) The test region for the gap distance; (b) the maximum radius r_o depending on r_{max}; (c) region in 3D.

Internal voids

Printed sand moulds seem the most suitable type of fabrication for complex glass geometries. As these mould are disposable, breaking the mould to release the glass is not a direct issue. However, to make assembly of the mould possible, every piece needs to be reachable. As a result, it is not possible to create a geometry with complete internal voids. There have been multiple attempts to integrate moulding constraints into the TO calculations. One such way with density-based optimization schemes is to restrict the densities of elements along the parting direction to just decreasing.

Another way is the virtual (VT) temperature method introduced by S. Liu et al. (2015). With this method, it is assumed that voids in the structure consist of a high-heat conductive material with spontaneous heating. The solid areas consist of a virtual thermal insulator. When a fully enclosed void occurs in the structure, the heat generated in that void can not lose this heat. Resulting in a high maximum temperature in the structure. However, when a void is connected to the boundaries of the structure, the heat generated can leave thus resulting in a very low maximum temperature value.

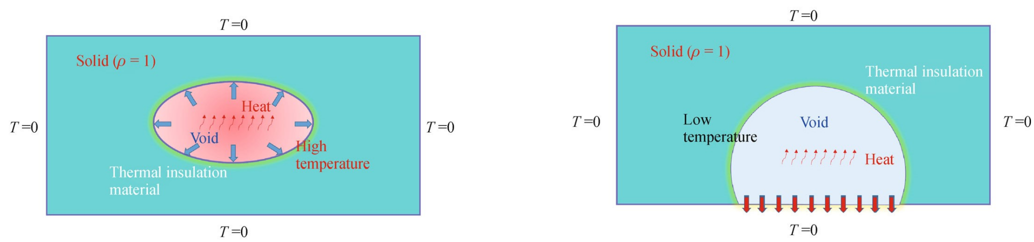
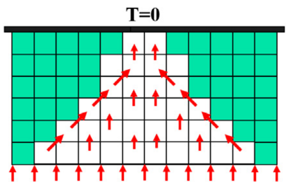


Figure 4.14: Virtual temperature method for two different structures. (a) with a fully enclosed void; (b) without a fully enclosed void

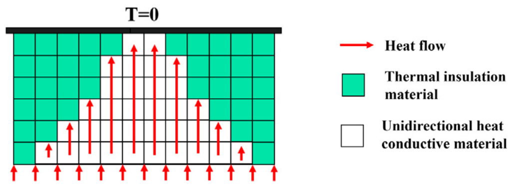
To model this a steady temperature model needs to be established. With this model, the criteria for a simply connected structure without any enclosed voids can be formulated as:

$$\overline{T}_{max} \leq \overline{T} \tag{4.26}$$

Here $\overline{T}_{max}$ is the maximum temperature value in the steady temperature field. The maximum allowable virtual temperature is given by the constant $\overline{T}$. Setting a tighter constraint for the $\overline{T}$ results in a structure in which the support structure (or mould) is easier to remove (S. Liu et al., 2015). Depending on the manufacturing process this method can be adapted. However, it is also mentioned by Li et al. (2018) that applying the VT method to a FE model might give false results, as the heat is transferred by the nodes, making heat flows as seen in figure 4.15 possible. This image also shows the same example when a Finite Volume method is applied instead.



(a) Heat flow modelled by FEM with concave boundary



(b) Heat flow modeled by FVM with concave boundary

Figure 4.15: Illustration of difference heat transmittance FEM and FVM

4.7.3 Structural Constraints

There are multiple ways of implementing a stress constraint in an optimization problem. The first is to evaluate the principal stresses for every individual element. However, this requires a lot of computational power. For this reason, the von Mises yield criterion is often adopted to calculate and avoid peak stresses (Bruggi & Duysinx, 2012). The von Mises criterion gives accurate results for materials that display similar behaviour in tension and compression. However, this gives problems when working with a brittle material like glass as the allowable stress in tension and compression is different (Damen et al., 2022). Alternatively, the Drucker-Prager failure criterion can be implemented for brittle materials like glass (Bruggi & Duysinx, 2012).

Principal Stresses

The principal stresses of a deformed three-dimensional hexahedral element can be derived from the stress-strain relationship. The principal stresses in 3D elements can be derived from the stress tensor in combination with the stress invariants. These invariants are given by (Chandrupatla & Belegundu, 2002):

$$I_1 = \sigma_x + \sigma_y + \sigma_z \tag{4.27}$$

$$I_2 = \sigma_x\sigma_y + \sigma_y\sigma_z + \sigma_x\sigma_z + \tau_{yz}^2 + \tau_{xz}^2 + \tau_{xy}^2 \tag{4.28}$$

$$I_3 = \sigma_x\sigma_y\sigma_z + 2\tau_{yz}\tau_{xz}\tau_{xy} - \sigma_x\tau_{yz}^2 - \sigma_y\tau_{xz}^2 - \sigma_z\tau_{xy}^2 \tag{4.29}$$

Drucker-Prager Stress Constraint

For the Drucker-Prager failure criterion, the feasible domain for the material is given by the equivalent stress measure σ^{eq} can be written as:

$$\sigma^{eq} = \frac{s+1}{2S}\sqrt{3J_{2D}} + \frac{S-1}{2S}J_1 \leq \sigma_{Lt} \tag{4.30}$$

Here s is the ratio between compression σ_{Lc} and tension σ_{Lc} strength of the material ($s = \sigma_{Lc}/\sigma_{Lc}$). The first stress invariant is here denoted by J_1 but is the same as I_1 defined in 4.27. J_{2D} Refers to the second deviatoric stress invariant, which is defined as (Liolios, 2020):

$$J_2 = \frac{1}{3}I_1^2 - I_2 \tag{4.31}$$

$$= \frac{1}{6}[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2] + \tau_{yz}^2 + \tau_{xz}^2 + \tau_{xy}^2 \tag{4.32}$$

Image 4.16 shows the difference in admissible range between a stress constraint set up with the principal stresses and Drucker Prager criterion.

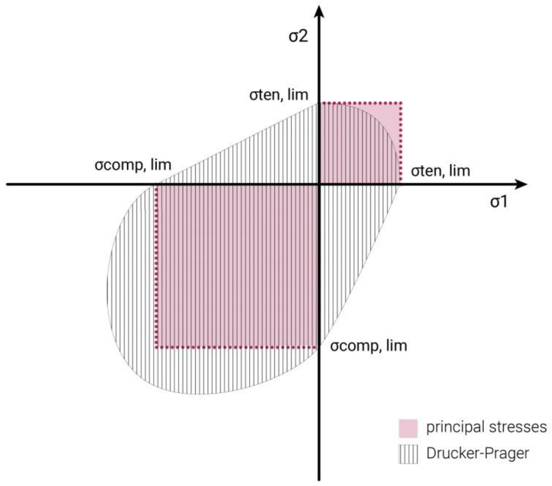


Figure 4.16: Qualitative diagram of difference in admissible range between principal stresses and Drucker-Prager criterion.

4.8 Discussion

For accurate analysis and topology optimization, it is essential to choose the right combination of finite element, optimization solver, objective and constraints. This thesis aims to create an algorithm for the topology optimization of a three-dimensional structure. While it is possible to achieve a three-dimensional structure with 1D and 2D elements, the 3D element is likely to be most suitable for this purpose. The use of 1D elements would result in a spatial truss. This would require smoothing out of the connections, as Damen (2019) stated, harsh corners will result in local peak stresses and must be avoided. This would possibly require a lot of post-processing. Furthermore, the structure would be limited to bars of a continuous size, which may not be the most optimal material distribution.

The bridge design of Koniari (2022) and prototype "A" of the topologically optimized concrete slabs made at the ETH in Zurich (Jipa et al., 2016) show that is it possible to create a 3D geometry with two-dimensional elements. In the case of the concrete slab, a bitmap was extruded to a 3D geometry, limiting the freedom of shape. Similarly, Koniari (2022) extruded the map created from the 2D elements, factoring in the final density of the elements.

Taking these points into consideration, the use of 3D elements seems the most promising. It is possible to use either a tetrahedral-based discretization of the domain or a hexahedron. The tetrahedral has the benefit that it has constant strain and thus constant stress in the element, making the calculation easier. While for the hexahedron multiple points have to be checked for the stress values, resulting in a more accurate stress interpolation but a longer calculation time.

As explained in section 4.2.3 the most suitable topology optimization method is the SIMP method. This preference follows out of the clear formulation of the method, the robust solutions it gives and the good convergence. A level-set method would give clearer boundaries, however, as it would require knowledge of a lower-level language like c++, it is not suitable for this thesis.

For the formulation of the optimization problem, three possible objectives are given. While compliance-based optimizations are known for their robust solutions, the final result is dependent on the volume fraction set by the designer. Making it possible that a stiff or stiffer structure could exist with less mass. Furthermore, Koniari (2022) showed that volume-based optimizations lead to just as good solutions with a clearer boundary for the structure. Stress-based optimizations for brittle materials are possible as has been shown by Jewett and Carstensen (2019). However, if compliance is not well integrated into the objective, it can lead to the deterioration of the structural stiffness. Thus, this objective will not be tested in this thesis.

For the constraints, Koniari (2022) has set up a clear overview of the constraints used in the 2D optimization. Some of these constraints can be adapted to a 3D case study, as many of the papers regarding these constraints also give formulations for 3D optimizations. However, the third dimension adds an extra constraint on internal voids. There are multiple approaches given in the literature to overcome this issue. Some do employ a finite volume method instead of a finite element. Creating possible difficulties when applied to an optimization using finite elements.

There are two possible ways to integrate the stress constraint into the algorithm, either by checking the principal stresses or using the Drucker-Prager criterion. While Koniari (2022) used both methods, no definite conclusion was made on the difference in the performance of both. Testing both methods on a smaller problem might be needed to determine the best approach.

Lastly, it needs to be mentioned that just optimizing for one possible situation might lead to very bad structural performance if loads change (Damen et al., 2022). Taking this into consideration, the performance of the final generated geometry needs instigated under different load cases.

5 Algorithm Development

This chapter will focus on the development of the algorithm. The algorithmis based on previous work from Koniari (2022), Sigmund (2001) and K. Liu and Tovar (2014). Sigmunds code was used to understand the fundamentals of a topology optimization code. Koniaris algorithm gave insight in the setup of a code with multiple functions as well as glass specific constraints. Lastly, the algorithm from Liu helped to understand how to efficiently set up a code for a 3D optimization.

The algorithm is written in Matlab and initially run on a HP laptop with an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz processor and 8 GB installed RAM. Vectorizing and optimizing the code made it possible to run the benchmark problem within a reasonable time on this laptop. However, for larger domains more computational capacity is needed for which the DelftBlue supercomputer, provided by the Delft High Performance Computing Centre was used.

5.1 Overview

Several versions of the algorithm were developed. The first is a Finite Element analysis of the algorithm. This is a stripped-down version of the algorithm without the optimization loop. It is used to validate the structural analysis with results from commercial software like ANSYS. The primary version of the algorithm follows the logic of the diagram shown in image 5.1, this is the version that this chapter will focus on.

This version of the algorithm can be challenging to work with if the user has no prior knowledge of the algorithm or Matlab. In order to facilitate easier usage a version of the algorithm has been developed that has a integrated workflow with Grasshopper. Section 5.2.2 will further expand upon the changes made to the code to facilitate this workflow.

The algorithm can be separated into different parts. The first part 'Initialization and Boundary Conditions Setup' encompasses entering the input values, like the dimensions of the domain as well as the material properties and certain values needed for the optimization like the penalization factor. These input values are stored in a separate function, called at the beginning of the optimization. The second part of the Initialization and Setup phase is defining the boundary conditions. This consists of defining the position of the load as well as defining the supports.

In the second part of the algorithm 'Matrix Initialization and Preprocessing' different functions are called to create vectors and matrices containing data that is repeatedly needed during the optimization loop. Calling these before entering the loop and storing them saves time during the optimization loop as the functions do not need to be run again. Important during this step are the two filtering functions as well as the creation of the connectivity matrix.

The third part of the algorithm consists of the optimization loop. In this loop, the optimization solver (Fmincon)iteretively minimizes the objective function while complying with the equality and inequality constraints posed by the constraint function. Both the objective and constraint function start with filtering the design variable x.

Depending on the preference of the user, the outcome of each iteration can be plotted to get visual output on the behaviour of the optimization. Furthermore, the outcome of the density variable is stored in an Excel file to be able to transfer the optimized geometry to grasshopper, where further post-processing of the geometry can happen.

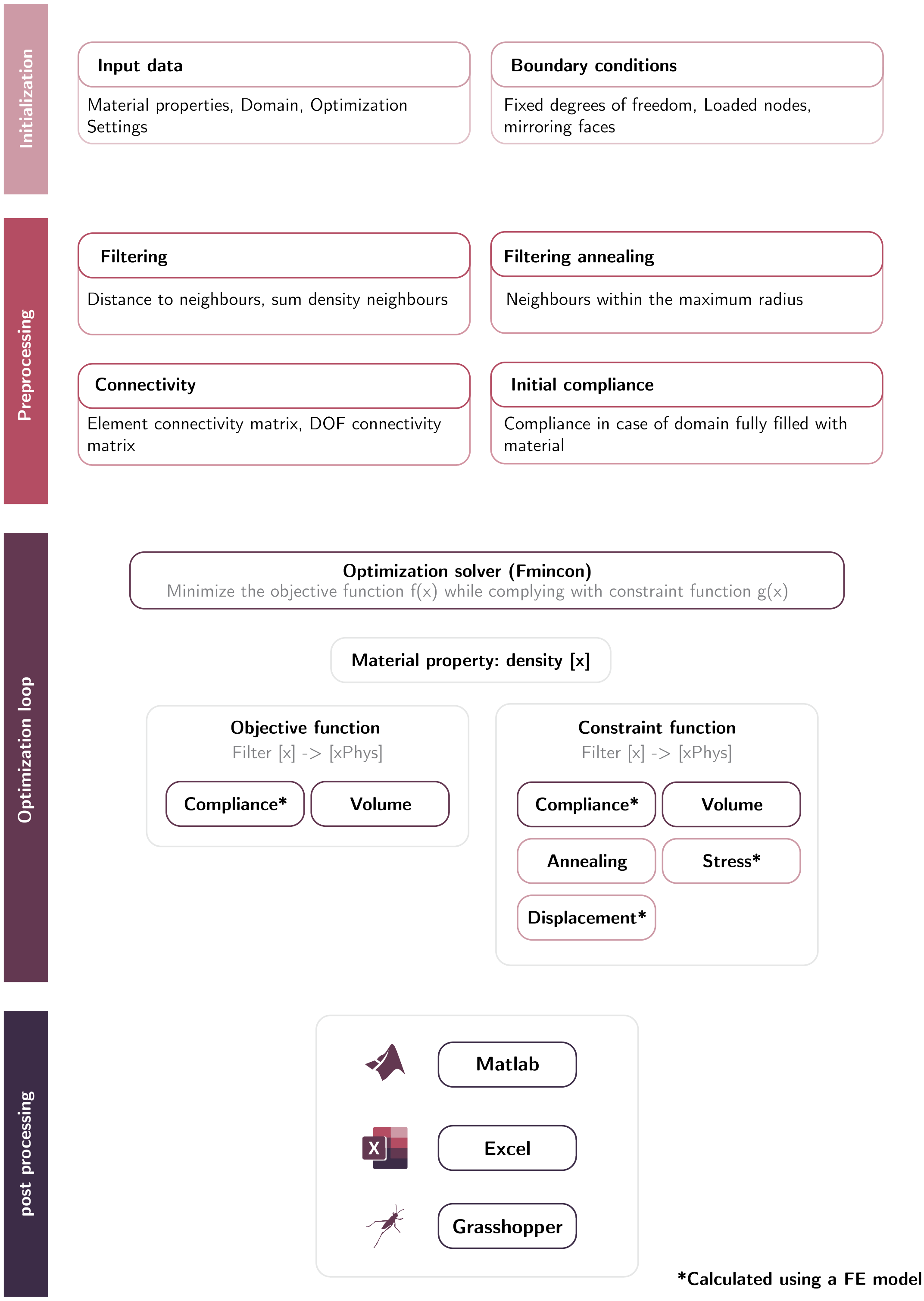


Figure 5.1: Overview algorithm

5.2 Initialization and Boundary Conditions Setup

The following section will introduce the steps taken during the initialization of the algorithm. During this phase the input values for the optimization are provided, as well as information on the domain dimensions, material properties and boundary conditions.

5.2.1 Input Values

The input values can be set by the user, for this thesis the material properties of glass are used. To facilitate different needs during the steps of the algorithm development, the algorithm was tested with multiple domain sizes. To limit the computational time, the width of both the benchmark and case study domain were set to the width of one element during the initial testing and structural verification of the algorithm. By setting the width to 0.02 the algorithm essentially runs a 2D optimization, similar to the optimizations done by Koniari (2022), using 3D elements. Making it so that the performance of both algorithms can be compared.

The mesh partitioning is also defined with the input values. As introduced in chapter 4.3 for structural calculations, the domain gets divided into smaller finite elements. Depending on the purpose of the calculation, different elements can be chosen. As the goal of this algorithm is a three dimensional optimization, 3D elements are used. The algorithm is based on 8-node hexahedron elements, with every node having three degrees of freedom. The size of the elements has a direct influence on the computational time of the algorithm. While a smaller element size gives a 'sharper' result, it can cause a significant increase in computational time. For a good compromise between resolution and computational time a element size of 0.02*0.02*0.02 m was used.

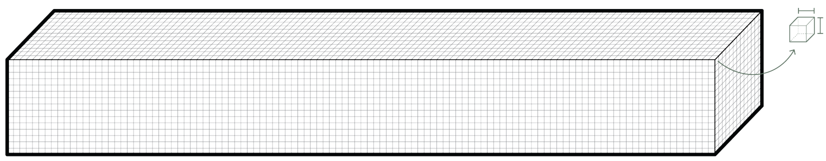
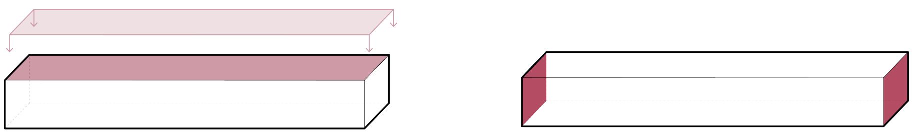


Figure 5.2: Example Mesh Partitioning

5.2.2 Boundary Conditions

The boundary conditions encompass the selection of supports and the distribution of loads. In the case study, which exhibits symmetry in both its length and width, the introduction of symmetric boundary conditions offers a substantial reduction in the number of elements required. These symmetric boundaries are established by effectively halving the calculated domain, either in terms of length or width, and setting the degrees of freedom on the mirrored faces to zero in the direction corresponding to the mirroring process.



(a) Distribution Loads Case Study **(b)** Support Conditions Case Study

Figure 5.3: Boundary Conditions Case Study

Boundary Conditions Grasshopper

As mentioned in the introduction, defining appropriate boundary conditions in MATLAB can present a significant challenge. Unlike some software with a user-friendly virtual interface, MATLAB requires the manual assignment of boundary conditions based on nodal numbers and their corresponding degrees of freedom. In cases where a design involves straightforward support and loading conditions, as seen in the case study, it is feasible to establish rules for identifying the nodes on the outer faces. However, for

(a) Calculation size after mirroring is introduced

(b) Support conditions reduced domain

Figure 5.4: Reduced domain after introducing the symmetric boundary conditions

more intricate designs or for individuals less familiar with coding languages like MATLAB, the task of formulating rules to identify the correct nodes can become quite challenging

To facilitate more complex optimizations tasks, a grasshopper script has been developed. Within this script, parameters such as the width, height and length of the domain, combined with the desired finite element size, are employed to generate all the necessary nodes. Subsequently, a selection process can be executed via a set of sliders. For each selection, the coordinates of these chosen nodes are recorded in an Excel sheet, which can be accessed by the Grasshopper version of the algorithm. The Matlab algorithm will translate the nodal coordinates to the nodal numbers and corresponding degrees of freedom. Appendix A.1 gives a more in depth explanation on how to use the grasshopper implementation of the Matlab code.

5.3 Matrix Initialization and Preprocessing

The next section will explain the functions that are part of the 'Matrix Initialization and Preprocessing' phase of the algorithm. It's essential to execute these steps before entering the optimization loop since the data compiled in this stage remains constant throughout the iterative optimization process. The following functions will be further elaborated: filtering functions, connectivity matrix and the element stiffness matrix.

5.3.1 Filtering Functions

Filters are used in topology optimizations to avoid numerical instabilities and additionally can be used to ensure minimum and maximum member sizes. There are two filtering functions included in the preprocessing phase of the algorithm. Both functions create arrays that map which elements fall within the neighbourhood defined by the filter size. Additionally, the minimum filter also maps the distance to these neighbouring elements to calculate the weight factor. A slight adaptation is made to the formulation of the density filter equation given in section 4.7.1. [1] The altered formulation is given by:

$$\tilde{x_i} = \frac{\sum_{j \in N_i} H_{ij} v_j x_j}{\sum_{j \in N_i} H_{ij} v_j} \tag{5.1}$$

The weight factor w_x is replaced by the weight factor H_{ij} that is defined by $H_{ij} = r_{min} - \text{dist}(i, j)$ (K. Liu & Tovar, 2014). The filtering for the maximum cross-section is part of the annealing constraint that will be discussed in section 5.4.3.

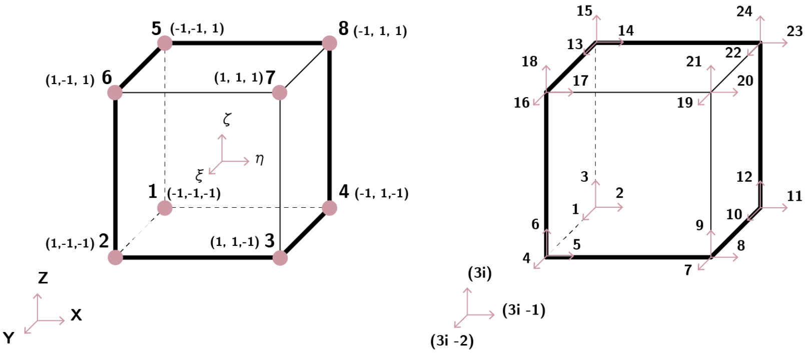
5.3.2 Element Stiffness Matrix

To solve the potential energy equation of the structure the global stiffness matrix needs to be derived. In chapter 4.3 the general formula for the stationary point is given by $f = Kq$ Where the stiffness matrix $[K]$ is given by $\int_V [B]^T [D][B] dV$ and depends on the element's shape functions.

While the global stiffness matrix changes during every iteration of the optimization loop, the elemental stiffness matrix (24x24) can be calculated during the preprocessing phase of the algorithm to save time during the optimization.

For a 8-nodal hexahedral element the stress strain relationship matrix $[D]$ comes from the generalized Hooke's law. This is a symmetric (6 x6) matrix that integrates the Young's modulus E and Poisson's ratio ν.

[1] K. Liu and Tovar (2014) makes a slight change in the syntax of the formulation, replacing the weight factor w_x by a weight factor denoted by H_{ij}. As the gradient functions provided by K. Liu and Tovar (2014) are used later in this chapter, the new formulation is given for consistency.



h

Figure 5.5: Master Cube

$$D = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5-\nu & \nu & \nu \\ 0 & 0 & 0 & \nu & 0.5-\nu & \nu \\ 0 & 0 & 0 & \nu & \nu & 0.5-\nu \end{bmatrix}$$

The B matrix that is introduced translates strain to displacement ($\varepsilon = Bq$) can be derived with the Lagrange shape functions (equation 5.2) that follow out of the master element shown in figure: 5.5.

$$N_i = \frac{1}{8}(1+\xi_i\xi)(1+\eta_i\eta)(1+\zeta_i\zeta) \tag{5.2}$$

These shape functions are used to define the displacements at any point in the element. As well as explain the x y z functions. Following the steps described in appendix A.2 the B matrix can be derived. After which the elemental stiffness matrix is given by equation 5.3 The integration of this function is performed numerically using the Gauss points. [2]

$$k^e = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} B^T D B |\text{det} J| d\varepsilon d\eta d\zeta \tag{5.3}$$

5.3.3 Connectivity Matrix

In the previous section a master cube is introduced. To connect the information from the elemental stiffnes matrix to the global structure information on the global numbering is needed. The connectivity matrix establishes the,is the local-global correspondence.

Image 5.6 shows an example of a small domain discretized by eight-noded cubic elements. Globally the nodes are identified with a number or node ID (NID_i) ordered row wise, left to right, top to bottom and front to back. Every node has three degrees of freedom (DOFs) that correspond to linear displacements in x-y-z directions. The node numbering of the mastercube follows a slightly different counting system, starting with the node at the left back bottom. For the numbering of the elements a column-wise ordering was followed, going top to bottom, left to right, front to back. [3] Table 5.1 shows an example of the connectivity matrix.

[2] For this integration the algorithm uses a code provided by Petraroia (2023), as this code was comprehensive and fast.

[3] First the element were numbered similar to the nodes;left to right, top to bottom and front to back. However, the standard reordering of MATLAB operates column and then pages wise. The difference could be solved using the permute function. For the readability of the algorithm, the numbering of the elements was changed to fit the operation of MATLAB. The counting of the nodes was not changed, as this did not influence any other function and the translation from the global counting to the local nodal numbers was already set up.

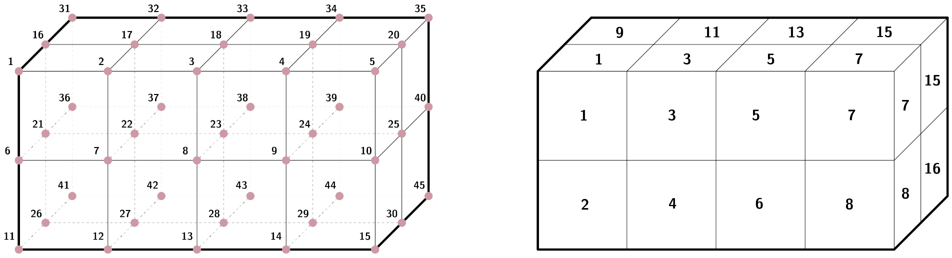


Figure 5.6: Global numbering nodes (left) and elements (right)

	Elements	Nodes							
Local		1	2	3	4	5	6	7	8
Global	1	21	6	7	22	16	1	2	17
	2	26	11	12	27	21	6	7	22
	16	44	29	30	45	39	24	25	40

Table 5.1: Example of the connectivity matrix

5.4 Optimization Loop

The next section will explain the steps included in the Optimization Loop. This phases serves as the core of the algorithm, where the code iteratively improves the geometry based on defined criteria. As discussed in the chapter 4 both a minimum compliance and a minimum volume optimization are considered and compared. First, the solver choice will be elaborated on. Then both the objectives and their mathematical formulation will be explained. Lastly, the different constraints that are included will be discussed.

5.4.1 Solver Choice

The optimization is done in Matlab v.2022. The Fmincon solver from the Optimization Toolbox is used. This is a solver for finding the minimum of a constrained nonlinear multivariable function. It offers multiple different algorithms for both large-scale and small-scale problems. The standard 'interior-point' algorithm suits best for this optimization, as it can solve large-scale problems, it will satisfy the bounds at every iteration and it can solve issues arising from Nan of Inf results. Furthermore, while the Fmincon algorithm has the option to use numerical gradients that are calculated by finite difference approximation, it also allows the user to add analytical gradients and a hessian function. With the increased computational need for a 3D algorithm, it became clear after testing that it was necessary to include functions to compute the partial derivatives analytically. Including this information proved to speed up the optimization by a vast amount and will be elaborated on in further sections. Lastly, the feasibility mode was used to ensure that the algorithm works to a geometry that confirms the constraints given.

5.4.2 Objectives

The objective function defines what is to be achieved during the optimization process. Both a minimum compliance and a minimum volume optimization are made to compare the results. To increase the performance of the algorithm the gradient for both objectives are defined analytically.

Compliance Objective

As described before, with a minimum compliance problem the objective is to find the best distribution of material that minimizes the structure's deformation. This is defined as:

$$\begin{aligned}
\text{find:} \quad & x = [x_1, x_2, x_3, ..., x_n]^t \\
\text{minimize:} \quad & c(\tilde{x}) = F^T U(\tilde{x}) \\
\text{subject to:} \quad & \frac{V(\tilde{x})}{V_0} = f \\
& U(\tilde{x}) K(\tilde{x}) = F \\
& 0 < x_{min} \leq x \leq 1
\end{aligned}$$

The derivative of the compliance is worked out by K. Liu and Tovar (2014), and given as:

$$\frac{\partial c(\tilde{x})}{\partial x_e} = \sum_{i \in N_e} \frac{\partial c(\tilde{x})}{\partial \tilde{x}_i} \frac{\partial \tilde{x}_i}{\partial x_e}$$

Where the partial derivative of the filtered density is given by:

$$\frac{\partial \tilde{x}_i}{\partial x_e} = \frac{H_{ie} v_e}{\sum_{j \in Ni} H_{ij} v_j}$$

The derivative of the compliance is given by:

$$\frac{\partial c(\tilde{x})}{\partial \tilde{x}_i} = -u_i(\tilde{x})^T [p \tilde{x}_i^{p-1} (E_o - E_{min}) k_i^0] u_i(\tilde{x})$$

Volume Objective

The volume objective minimizes the total volume of the structure. This is calculated by the sum of all the densities multiplied by the volume of each element (v_e).

$$\begin{aligned}
\text{find:} \quad & x = [x_1, x_2, x_3, ..., x_n]^t \\
\text{minimize:} \quad & V(\tilde{x}) = \sum_{e=1}^{N} \tilde{x}_e v_e \\
\text{subject to:} \quad & \frac{c}{c_L} \leq 1 \\
& U(\tilde{x}) K(\tilde{x}) = F \\
& 0 < x_{min} \leq x \leq 1
\end{aligned}$$

The derivative is given by:

$$\frac{\partial V(\tilde{x})}{\partial x_e} = \sum_{i \in N_e} \frac{\partial V(\tilde{x})}{\partial \tilde{x}_i} \frac{\partial \tilde{x}_i}{\partial x_e} \tag{5.4}$$

Where

$$\frac{\partial V(\tilde{x})}{\partial \tilde{x}_i} = v_i$$

5.4.3 Constraints

The fmincon algorithm can take one or multiple constraints. These constraints can be equality (C_{eq}) and inequality (C) constraints. Both can be given a additional gradient matrix to improve the efficiency of the code. Important to note is that it is not possible to just define the gradient for one constraint, but the gradient constraint has to match the size of the corresponding equality or inequality array.

$$\begin{bmatrix}
\frac{\partial C_1}{\partial x_1} & \frac{\partial C_2}{\partial x_1} & \cdots & \frac{\partial C_n}{\partial x_1} \\
\frac{\partial C_1}{\partial x_2} & \frac{\partial C_2}{\partial x_2} & \cdots & \frac{\partial C_n}{\partial x_2} \\
\cdots & \cdots & \cdots & \cdots \\
\frac{\partial C_1}{\partial x_n} & \frac{\partial C_2}{\partial x_n} & \cdots & \frac{\partial C_n}{\partial x_n}
\end{bmatrix}$$

Compliance Constraint

The compliance constraint ensures that the compliance of the structure is under a certain threshold that can be set by the user. The mathematical formulation of the compliance constraint is given by equation 5.5.

$$C_{cmpl} = \frac{c(\tilde{x})}{c_0 a_c} - 1 \leq 0 \tag{5.5}$$

The c_0 refers to the initial compliance of the domain with full material. A factor for the allowable compliance percentage a_c is introduced. A lower a_c value results in a faster convergence. However, setting the value to low will limit the feasible area of the solution.

The derivative is similar as that of the compliance objective with the initial compliance and factor for the allowable compliance percentage added.

$$\frac{\partial C_{cmpl}}{\partial \tilde{x}_i} = \frac{-u_i(\tilde{x})^T [p\tilde{x}_i^{p-1}(E_o - E_{min})k_i^0] u_i(\tilde{x})}{c_0 a_c} \tag{5.6}$$

Volume Constraint

The volume constraint ensures that the total volume of the structure is under a desired volume fraction v_{frac} specified by the user. The mathematical formulation of the volume constraint is given by the following equation:

$$C_{vol} = \frac{V(\tilde{x})}{V_0} - 1 \leq 0 \tag{5.7a}$$

$$\text{where:} \quad V(\tilde{x}) = \sum_{e=1}^{N} \tilde{x}_e v_e \tag{5.7b}$$

$$V_0 = v_{frac} * \sum_{e=1}^{N} v_e \tag{5.7c}$$

The gradient of this constraint is equal to that of the gradient of the volume objective given by equation 5.4.

Stress Constraint

The stress constraint is introduced to ensure that the maximum value for tensile and compressive stress are not exceeded. To achieve the most accurate stress analysis, the principal stresses are calculated at the 8 Gauss points of every element. For this calculation $\sigma = DBq$ is used to calculate the stress tensor at the Gauss points. The deformations (q) follow out solving the equilibrium equation $F = Kq$, and the B matrix is the same as that of the stiffness matrix. Taking the eigenvector of the stress tensor results in the 3 principal stresses. Of these 3 principal stresses the highest is taken as the maximum tensile stress at this point and the lowest value is taken as the compressive stress at this point in the element. [4] Per element both the maximum $\sigma_{\text{tens,e}}$ and the minimum $\sigma_{\text{comp,e}}$ of the 8 values are evaluated using the formulas 5.8. These formulas are derived from altered qp approached proposed by Bruggi and Duysinx (2012).

$$C_{comp} = \tilde{x}_e^{(p-q)} \left(\frac{\sigma_{\text{comp,e}}}{\sigma_{\text{comp,lm}}} \right) \leq 1 \tag{5.8a}$$

$$C_{tens} = \tilde{x}_e^{(p-q)} \left(\frac{\sigma_{\text{tens,e}}}{\sigma_{\text{tens,lm}}} \right) \leq 1 \tag{5.8b}$$

Stress Gradient

To decrease the computational time of the algorithm a gradient for the stress constraint is provided.

[4] Important to note is that if a element is purely under compression (or tension) all values can be negative (or positive).

This gradient is derived from equations 5.8. Technically σ_e is a function of $\tilde{x}$ resulting in the following derivative:

$$\frac{\partial C_{stress}}{\partial \tilde{x_i}} = \frac{(p-q)\tilde{x}^{(p-q-1)}\sigma_e(\tilde{x}) + \tilde{x}^{(p-q)} * (\frac{\partial}{\partial \tilde{x}}\sigma_e(\tilde{x}))}{\sigma_{lm}} \tag{5.9}$$

However, within the scope of this thesis this is simplified to:

$$\frac{\partial C_{stress}}{\partial \tilde{x_i}} = (p-q)\tilde{x}^{(p-q-1)}\left(\frac{\sigma_e(\tilde{x})}{\sigma_{lm}}\right) \tag{5.10}$$

Resulting in a gradient matrix that takes the following shape:

$$\nabla C_{stress} = \begin{bmatrix} \frac{\partial C_{stress\ x_1}}{\partial x_1} & 0 & \dots & 0 \\ 0 & \frac{\partial C_{stress\ x2}}{\partial x_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{\partial C_{stress\ xn}}{\partial x_n} \end{bmatrix} \tag{5.11}$$

Deflection Constraint

The deflection constraint is introduced to ensure that the permissible deflection for a glass structure is not exceeded. The deflection (U) of every node (k) in all three directions follows out of the finite element calculation. Depending on the optimization problem all vertical displacements U(k*3) can be included in the constraint or just a set of critical nodes can be tested. The formulation of the constraint is given by:

$$C_{\text{def}} = \frac{U_{\text{crit dof}}}{U_{\text{max}}} - 1 < 0 \tag{5.12a}$$

$$\text{where:}\quad U_{\text{max}} = \frac{\text{lenght}}{500} \tag{5.12b}$$

Deflection gradient
The deflection gradient is derived from equations 5.12 and given by:

$$\nabla C_{\text{def}} = \frac{\frac{\partial}{\partial \tilde{x}} U_{\text{crit dof}}}{U_{\text{max}}}$$

Here the critical deflection follows out of the equilibrium equation:

$$U(\tilde{x})K(\tilde{x}) = F$$

Of which the derivative with respect to $\tilde{x}_i$ is given by:

$$\frac{\partial K(\tilde{x})}{\partial \tilde{x}_i}U(\tilde{x}) + \frac{\partial U(\tilde{x})}{\partial \tilde{x}_i}K(\tilde{x}) = 0 \tag{5.13}$$

Using

$$\frac{\partial K(\tilde{x})}{\partial \tilde{x}_i} = p\tilde{x}_i^p - 1(E_0 - E_{min})K_i^0 \tag{5.14}$$

Combines to:

$$\frac{\partial U(\tilde{x})}{\partial \tilde{x}_i} = -U(\tilde{x})[p\tilde{x}_i^{p-1}(E_0 - E_{min})K_i^0]K(\tilde{x})^{-1} \tag{5.15}$$

Taking the partial derivative for a certain element will only result in a non zero value for the deflection of the degrees of freedom corresponding to that element. This makes it possible to simplify and speed op the calculation by calculating only the vector for specific elements and filling it into a larger gradient matrix.

$$\nabla C_{\text{def }(\text{dofs}_{x_i})} = \frac{-U(\text{dofs}_{x_i})[p\tilde{x}_i^{p-1}(E_0 - E_{min})k_i^0]K(\text{dofs}_{x_i}, \text{dofs}_{x_i})^{-1}}{U_{\text{max}}} \tag{5.16}$$

Annealing Constraint

The annealing constraint is introduced to keep control over the max cross section size in the optimized geometry. The maximum allowable cross section size is determined by the following formula from Koniari (2022):

$$d_{max} = \sqrt{\frac{\sigma_{res} T_{annmax} (1-\mu)\lambda}{\Delta T E a_{ex} \rho c_p b}} \tag{5.17}$$

To ensure that there is homogeneous shrinkage a large difference in cross section size must be avoided. So if d_{max} is larger then $2 * d_{min}$, $d_{max} = 3 * d_{min}$.

To calculate if parts of the structure don't exceed the maximum cross section size, the sum of densities of each elements neighbours is divided by the maximum possible volume if all neighbours were filled. A parameter a_v is introduced to ensure that the volume within the neighbouring elements stays under the threshold.

$$C_{ann}(x_i) = \frac{\sum_{j \in N_i} v_j x_j}{\sum_{j \in N_i} v_j * a_v} - 1 < 0 \tag{5.18}$$

$$\tag{5.19}$$

Gradient Annealing

The annealing constraint takes the shape of a vector with the size: (number of elements, 1). The gradient is created by taking the partial derivative of $\tilde{x}$ of each element x_e for every constraint, resulting in a matrix of size: (number of elements, number of elements). If x_e is not a part of N_i $\partial C_{ann}(x_i) \setminus \partial x_e = 0$, otherwise the gradient is determined by:

$$\begin{aligned} \nabla C_{ann}(x_1, x_2, x_3, \ldots, x_n) &= \frac{\partial}{\partial x_e} \frac{\sum_{j \in N_i} v_j x_j}{\sum_{j \in N_i} v_j * a_v} \\ &= \frac{v_e}{\sum_{j \in N_i} v_j * a_v} \end{aligned}$$

5.5 Post-Processing

The last part of the algorithm consists out of writing the data from the converged geometry to a excel file and if desired plotting the geometry in Matlab. The computations executed on the supercomputer do not generate graphical output. Therefore, a separate script has been developed to read the Excel output and generate geometry plots, as well as provide values for peak stresses and maximum deformation. Additionally, the Excel file can be read by a Grasshopper script for transferring the geometry to Rhino. This opens up further possibilities for post-processing, such as smoothing out the geometry and resolving any geometric errors. Section 8.1 will provide insight into the specific steps taken to post-process the geometry.

6 Algorithm Performance

This chapter will elaborate on the testing procedures conducted to asses the algorithm's performance. The result of the optimization problem depends on multiple variables. The right configuration of the problem setup is needed to ensure convergence towards a viable solution. The varying effects of the different parameters were explored to understand why in some instances the algorithm fails to converge.

The first section of this chapter will elaborate on the structural benchmarking tests. These tests serve to ensure the FE analysis is working properly. Next, both the performance of the volume optimization algorithm and the compliance optimization algorithm will be discussed. This assessment was conducted using both a two-dimensional and a small three-dimensional domain.

6.1 Domains, Properties and Boundary Conditions

As stated in the introduction different domains and boundary conditions were used to assess the performance of the algorithm over a range of settings. Three domain sizes will be explored in the following tests. Table 6.1 gives an overview of the dimensions of these domains. Both the 2D cantilever (figure: 6.1) and the 2D case study (figure: 6.2) are used for the verification of the structural model. For analysing the performance of the compliance and volume optimization algorithms all three domains were initially explored, however, the following sections will only discuss the results from the 2D case study and 3D beam (figure 6.3).

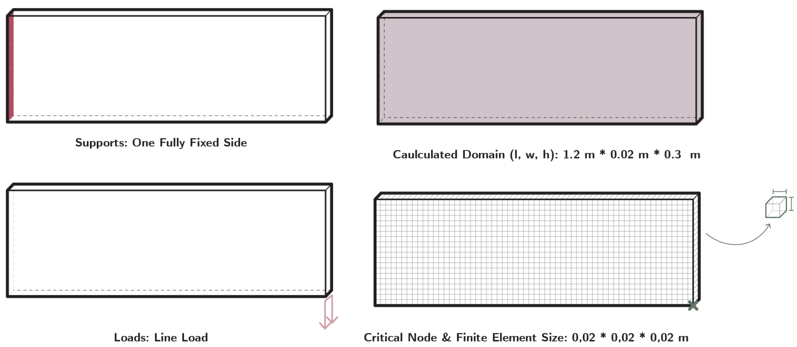


Figure 6.1: Boundary conditions and size 2D Cantilever

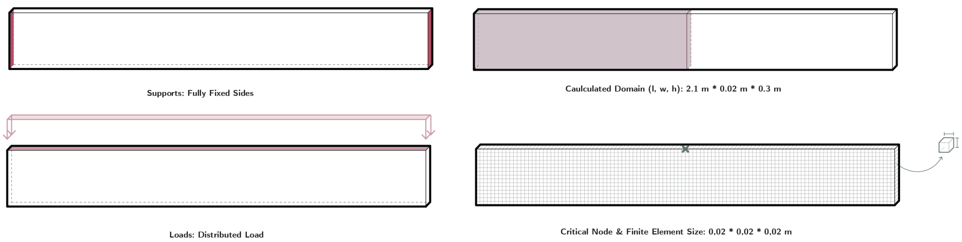


Figure 6.2: Boundary conditions and size 2D Case Study

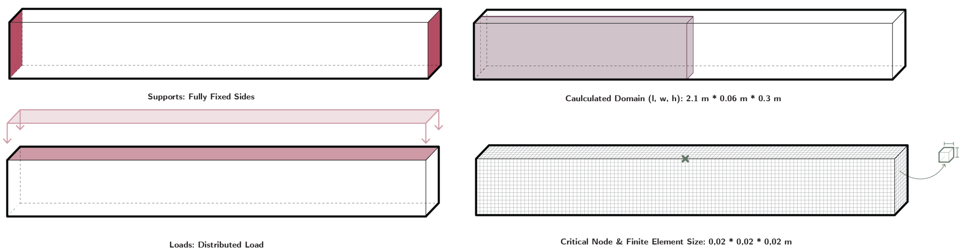


Figure 6.3: Boundary conditions and size 3D Beam

	Length	Width	Height	Size FE	Mirror length	Mirror width	Mirror height
2D Cantilever 2D	1.2	0.02	0.4	0.002	no	no	no
2D Case study	2.1	0.02	0.3	0.002	yes	no	no
3D Beam	2.1	0.12	0.3	0.002	yes	yes	no

Table 6.1: Domain sizes

6.2 Verification of the Structural Model

To validate the structural model a comparison was made between the results obtained from Matlab and the commercial software ANSYS. In both software, the calculations were conducted using the same domain and boundary conditions. Tables 6.2 and 6.3 give an overview of the obtained results.

The deformation in both models differs by an insignificant amount. There is however a slightly larger difference in the stresses calculated. While the difference between the stresses is very small for the case study, ANSYS gives a significantly higher peak tensile stress for the benchmark problem. ANSYS does not provide any insight into their method to calculate the peak stresses, but it can be assumed that a more precise interpolation of the principal stresses is done. For the benchmark problem, the load is also very localized, which might be the reason why the Matlab code has more issues calculating this load accurately.

As the stress calculation is already computationally expensive and gets very comparable results in the case of a distributed load, it was decided to not finetune the calculation any further to include more points. However, because of this difference, it does become important to do a final check of the optimized geometry in ANSYS to make sure none of the stress limits are exceeded. The diagrams of the results can be found in appendix A.3.

	Maximum displacement	Maximum principal stress	Minimum principal stress
	mm	Kn / mm2	Kn / mm2
ANSYS	-0.94366	0.002419	-0.002099
Matlab	-0.94430	0.001993	-0.001976

Table 6.2: Results 2D Case study (distributed load)

	Maximum displacement	Maximum principal stress	Minimum principal stress
	mm	Kn / mm2	Kn / mm2
ANSYS	-0.4309	0.003254	-0.001380
Matlab	-0.4148	0.002349	-0.001321

Table 6.3: Results 2D Cantilever (line load)

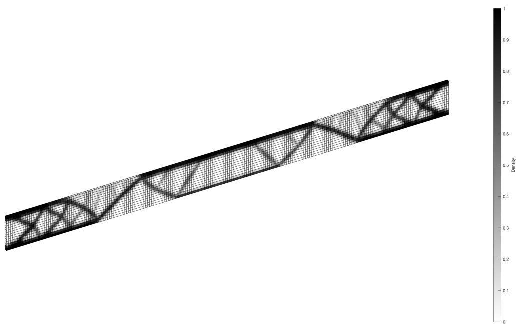
6.3 Compliance Objective Algorithm 2D

As mentioned in chapter 4.4.1 a minimum compliance optimization maximizes the global stiffness of the structure. The mathematical formulation introduced in chapter 5.4.2 is applied in the algorithm. The following sections focus on the results of testing the individual constraints on the case study domain. The main purpose of these tests is to determine if the constraint is working as expected. Additionally, it gives insight into the influence of each parameter used to define the constraints.

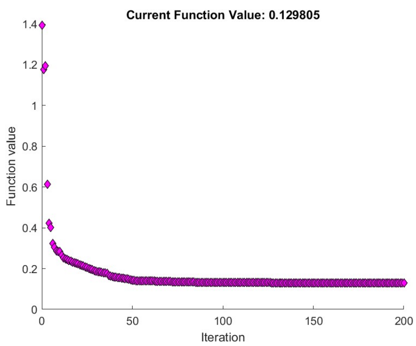
6.3.1 Volume Constraint

The volume constraint ensures a maximum usage of material. The compliance algorithm with volume constraint was tested on both the benchmark cantilever problem and the case study domain. Two parameters were found to be essential in the convergence of the algorithm. These are the values for the volume constraint v_{frac} and the initial volume $v_{initial}$.

The compliance optimization for the case study domain with a volume constraint of 30% gave the results seen in figure 6.4. The calculation starts with an initial volume of 30% and the compliance decreases to 0.129. The calculation is cut off by the iteration limit of 200 before it is able to converge (time: 1.8 minutes). At this iteration, the volume constraint is met. [1]



(a) Result geometry



(b) Plot of function value per iteration

Figure 6.4: Results 2D case study domain optimized for compliance with volume constraint of 30%

6.3.2 Principal Stress Constraint

In the algorithm, the principal stress constraint is divided into a constraint for the maximal principal stress (tensile stress) and the minimum principal stress (compressive stress). For every element it is checked that neither of these constraints is exceeded. As it was chosen to take a very simplified version of the stress gradient, testing the algorithm for the principal stress constraint was deemed essential.

A compliance optimization with a stress constraint results in a full domain.[2] To test the influence of the different parameters included in the stress constraint, it has to be combined with a volume constraint. Two parameters were found to be essential for the convergence of the algorithm. These are the values for the initial volume $v_{initial}$ and the relaxation parameter q that is included in the stress constraint.

Firstly, the constraint is tested with an initial volume of 30%, a volume constraint of the same amount and a relaxation parameter of 2.8. The calculation is cut off after 59 iterations as the step size is reached and the constraints are within the given limits, at this point the compliance decreased to 1.16. However, the resulting geometry is that of a full grey domain with all elements having a density between 0.25 and 0.4 [3].

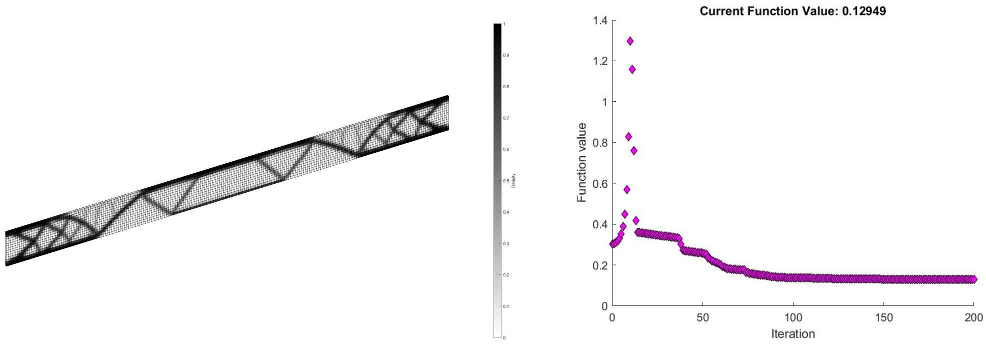
Increasing the initial volume to 50% while keeping q as 2.8 and the volume constraint as 30% results in a similar geometry as obtained during the compliance optimization with only a volume constraint

[1]The iteration limit was set to 200 after testing showed that the algorithm would continue into the thousand iterations without reaching the stepsize for certain calculations. After iteration 200 the function value does not get significantly lower. Making the difference in geometry between iterations 200 and 1000 negligible.

[2]All tests of the individual constraints can be found in appendix A.4.1

[3]The geometry of this optimization can be found under figure A.15 in appendix A.4.2

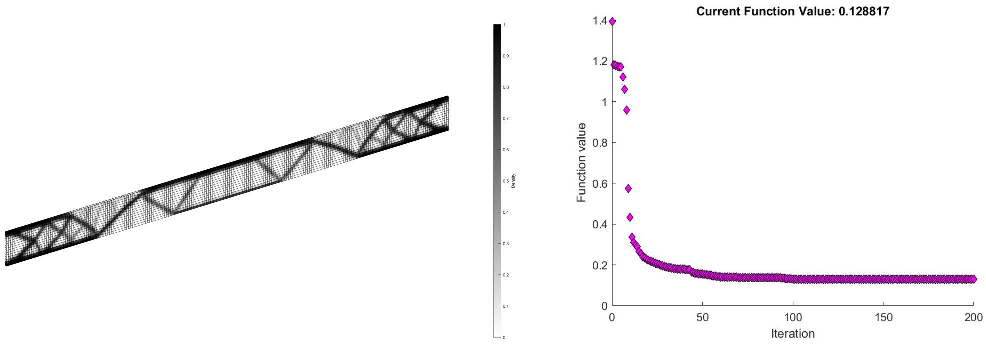
(figure 6.5). In the function value graph, it can be seen that during this optimization the compliance first goes up, this is because the algorithm first wants to satisfy the volume constraint.



(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.5: Results 2D case study domain optimized for compliance with a stress and volume constraint ($q = 2.8$ $v = 0.3$ $v_{initial} = 0.5$)

Decreasing the q value to 2.5 results in a similar geometry as increasing the initial volume. Figure 6.6 shows the results of a compliance optimization with a volume constraint of 30%, an initial volume of 30% and a q value of 2.5. This shows that for combining multiple constraints, sometimes it might be needed to make slight changes to the parameters to get the algorithm to converge.

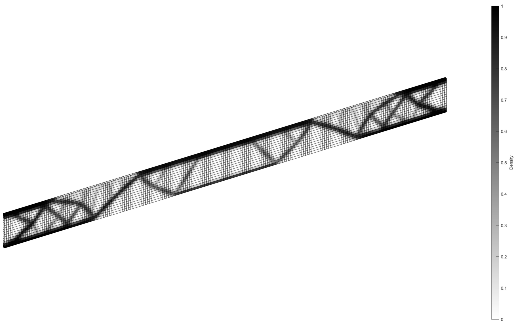


(a) Result Geometry **(b)** Plot of function value per iteration

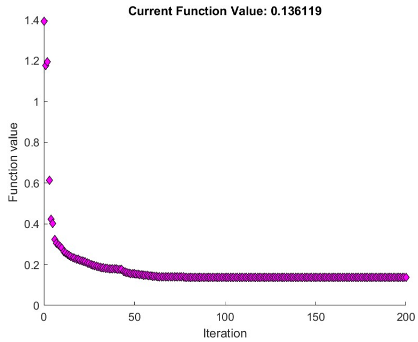
Figure 6.6: Results 2D case study domain optimized for compliance with a stress and volume constraint ($q = 2.5$ $v = 0.3$ $v_{initial} = 0.3$)

6.3.3 Deflection Constraint

Similar to the stress constraint, a compliance optimization with only a deflection constraint will result in a completely filled domain. To test the effect of the deflection constraint on the optimization, it can be combined with a volume constraint. Image 6.7 shows the results of an optimization including both a volume and deflection constraint. The calculation starts with an initial volume of 30% and the compliance decreases to 0.136 and is cut off by the iteration limit of 200 before it is able to converge. While the compliance optimization with just the volume constraint did not exceed the deflection constraint. Optimizing with both constraints results in a slightly different geometry and a higher function value.



(a) Result Geometry

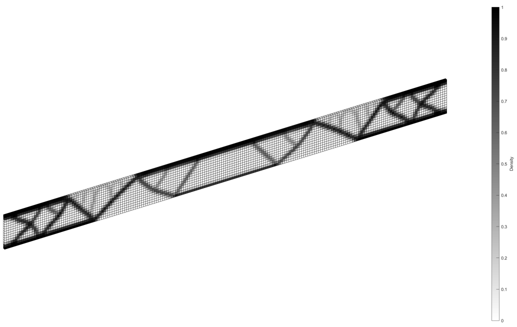


(b) Plot of function value per iteration

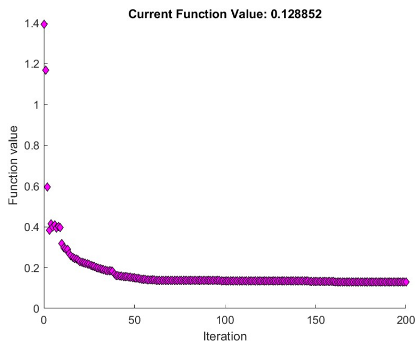
Figure 6.7: Results 2D case study domain optimized for compliance with a deflection constraint

6.3.4 Annealing Constraint

The annealing constraint involves the parameter a_v. This variable represents the percentage of the volume within a radius of r_max that is allowed to be filled. Testing multiple values a_v values showed that if set to low (strict) the algorithm produces geometries where the 'nodes' do not connect. If set too high the structure might still exceed the maximum cross-section area. Two different values were tested for the a_v factor: 0.75 and 0.9. Both optimizations had a volume constraint of 30% and were stopped at the iteration limit of 200. The main difference between both optimizations is the size and number of bars near the supports. With the result of $a_v = 0.75$ having slightly less but thicker bars than the optimization with $a_v = 0.9$.

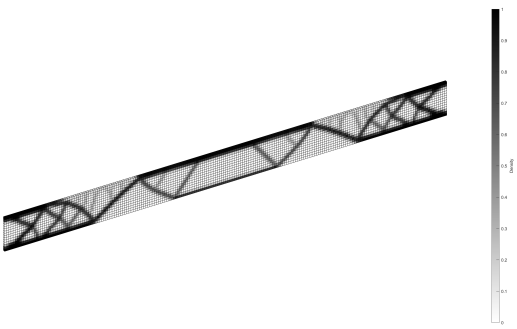


(a) Resulting Geometry

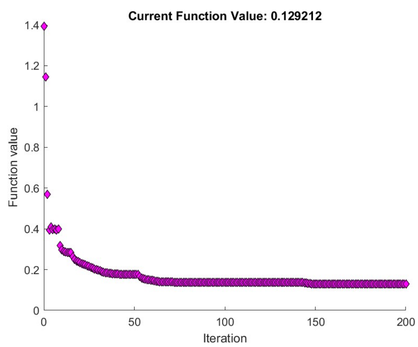


(b) Plot of function value per iteration

Figure 6.8: Results 2D case study domain optimized for compliance with an annealing constraint ($a_v = 0.75$)



(a) Result Geometry



(b) Plot of function value per iteration

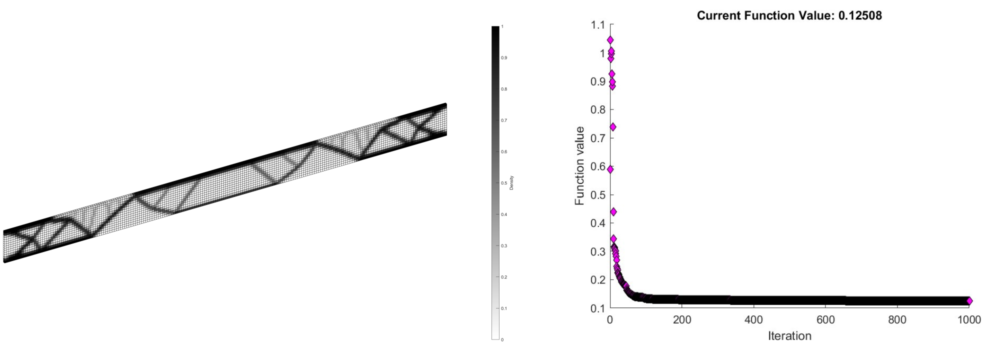
Figure 6.9: Results 2D case study domain optimized for compliance with an annealing constraint ($a_v = 0.9$)

6.3.5 Combining Constraints

As mentioned in the previous sections, certain constraints conflict with each other. For example, the tensile constraint might want to increase the densities around a certain element to avoid exceeding the tensile stress limit. However, this might mean there is too much material for the annealing constraint to be met. If the parameters for the constraints are not adjusted to each other it is possible that the algorithm does not converge to a suitable geometry. One successful example is the setup with the previously discussed parameters set as shown in table 6.4. The result reached is similar to that of Koniari (2022). With a geometry that represents two cantilevers supporting a simply supported beam.

setup	Constraints	Initial volume	Volume constraint	Stress relaxation	a_v	F value
1	ALL	0.4	0.3	2.8	0.9	0.125

Table 6.4: Setup Parameters



(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.10: Results 2D case study domain optimized for compliance with all constraints ($v_{initial} = 0.4$)

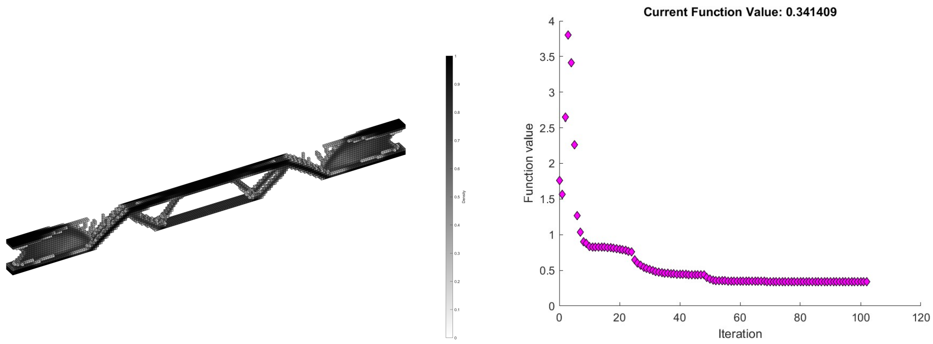
The geometry reached is similar to that of Koniari (2022). With a geometry that represents two cantilevers supporting a simply supported beam.

6.4 Compliance Optimization 3D Beams

Certain issues arose when directly translating the previous findings to the three-dimensional case study (see appendix A.4.3). As the calculation time of this domain was too long to efficiently check why certain abnormalities were occurring a domain size representing a beam was introduced (figure 6.3). These beams were optimized using a combination of constraints and values to see which changes occur with the introduction of more elements.

6.4.1 Volume Constraint

Optimizing the domain for compliance with a volume constraint of 30% and an initial volume of 40% results in the geometry depicted in figure 6.11. A similar contour as the 2D optimization is observed. However, a noticeable difference can be seen at the sides of the beam. Here the structure assumes an 'I' shaped configuration with flanges and a web as opposed to the truss-like structure observed in the two-dimensional optimization. The middle segment of the beam does retain the truss-like topology. In the 2D optimization, grey values are observed in the region where the topology changes from a cantilever to a simply supported beam. Koniari (2022) observed these values and named them 'nerves'. In the 3D optimizations, these nerves start getting the shape of thinner members that extend up to form thin supports in the voids.

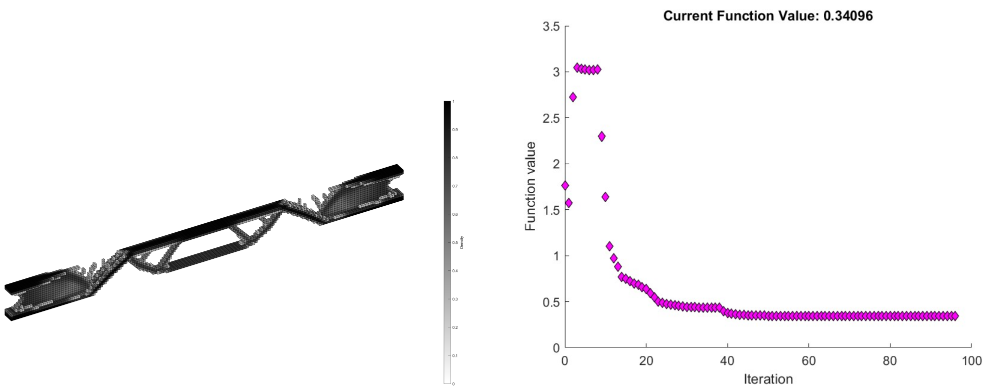


(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.11: Results 3D Beam compliance optimization with volume constraint (initial volume 0.4)

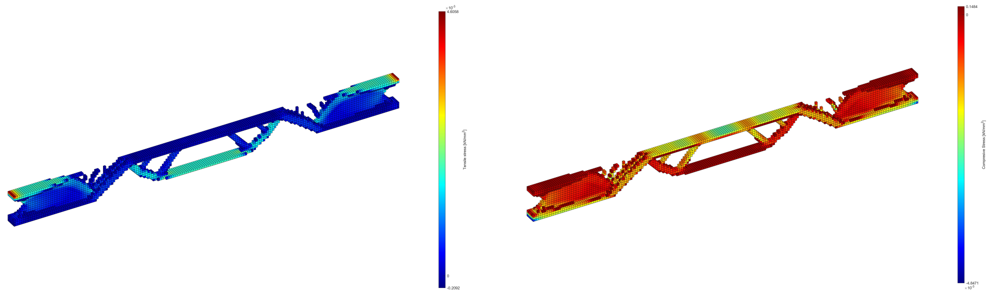
6.4.2 Principal Stress Constraint

Optimizing the domain with a volume constraint and the addition of the principal stress constraint results in a very similar geometry (figure 6.12). However, instead of having parallel beams, one solid beam connects the top and bottom flange at the centre of the design. Pointing out that in the previous design (figure 6.11), the tensile stress might have been exceeded at those locations. Again the thinner members can be seen to support the void at the top.



(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.12: Results 3D Beam compliance optimization with volume and stress constraint (initial volume 0.4)

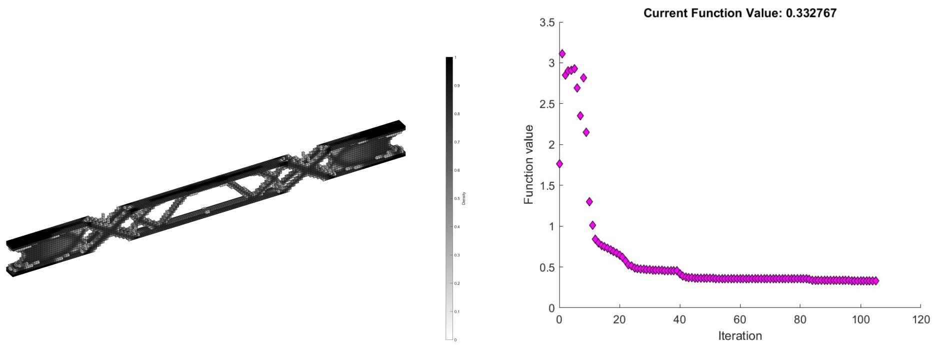


(a) Tensile Stress Graph (b) Compression Stress Graph

Figure 6.13: Stress graphs result 3D Beam compliance optimization with volume and stress constraint

6.4.3 Annealing Constraint

Optimizing the domain for compliance with a volume and annealing constraint results in the geometry as seen in figure 6.14. This geometry significantly differs from the previous results. It is almost symmetric in its height and the side and middle section are connected by a x shape. This shape is also observed in early iterations of the volume/stress optimizations. However, in these optimizations, the connection slowly converges to a v-shape that is observed in the final geometries.



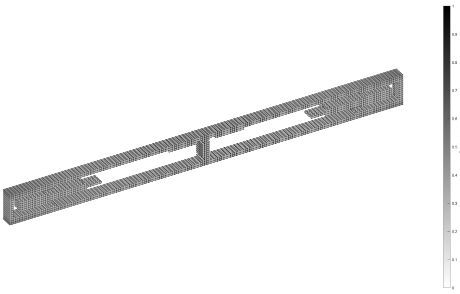
(a) Result Geometry (b) Plot of function value per iteration

Figure 6.14: Results 3D Beam compliance optimization with volume and annealing constraint (initial volume 0.4)

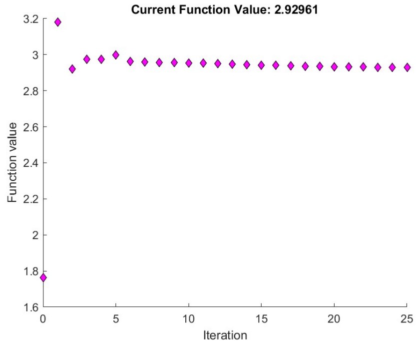
6.4.4 Combining Constraints

Combining all constraints with an initial volume of 0.4; volume constraint 0.3; q factor 2.8; a_v value 0.9 results in a grey block (figure 6.15). Lowering the initial volume to 0.3 and keeping the other parameters unchanged enables the algorithm to find a discreet geometry. Lowering the q value while keeping the initial volume at 0.4 also results in a discreet geometry.[4] This shows a similar trend as the 2D calculations where either the q factor needs to be lowered or the initial volume needs to be raised. A final test was done with an initial volume of 0.3; volume constraint 0.3; q factor 2.5 and $_v$ value of 0.9. The function value plot of this optimization is continuously decreasing and this optimization is taken as most potential for the large-scale calculation.

[4]The geometries of these calculations and the plots can be found under figure A.16 and figure A.17 in appendix A.4.2.

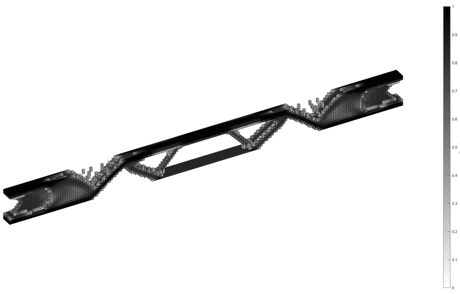


(a) Result Geometry

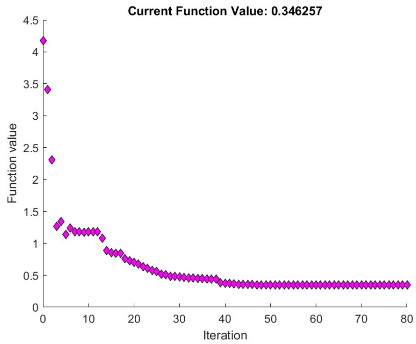


(b) Plot of function value per iteration

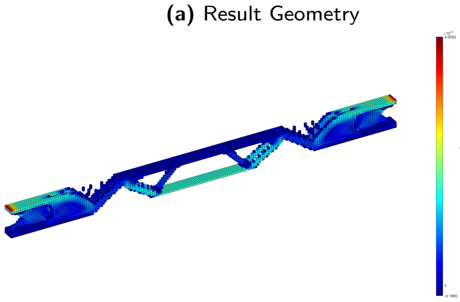
Figure 6.15: Results 3D Beam compliance optimization with All constraints (vi 0.4; q 2.8; av 0.9)



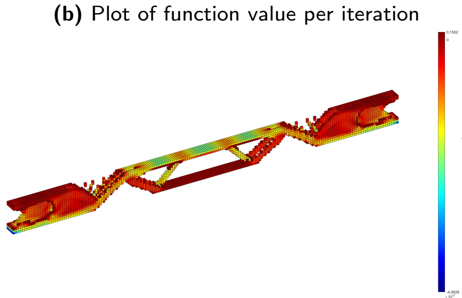
(a) Result Geometry



(b) Plot of function value per iteration



(c) Tensile Stress Graph



(d) Compression Stress Graph

Figure 6.16: Results 3D compliance optimization setup Beams, All constraints (vi 0.3; q 2.5; av 0.9)

6.5 Volume Objective Algorithm 2D

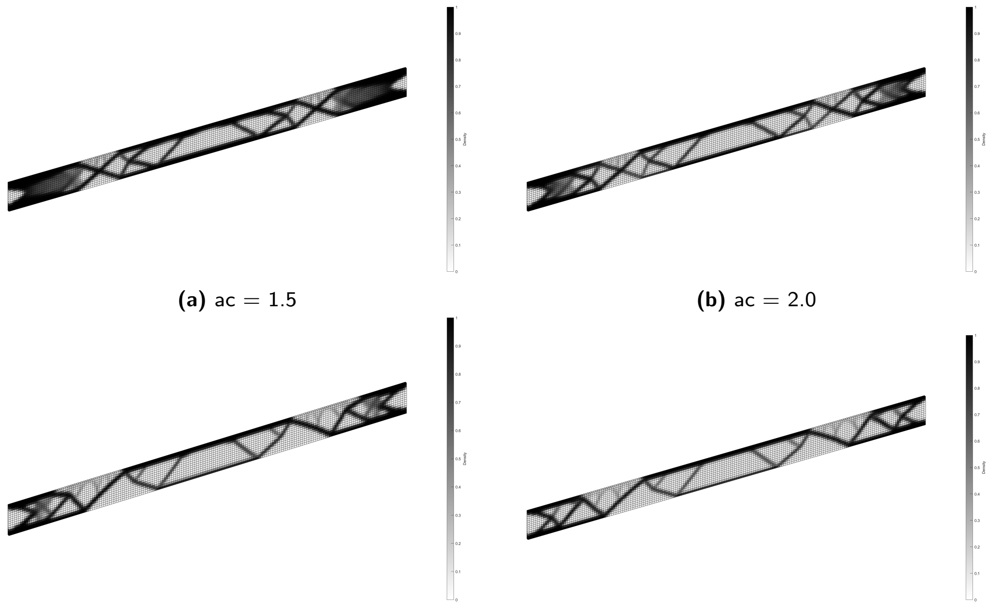
In this section results from the volume optimization optimization algorithm with the different constraints are discussed. The algorithm was tested on both the benchmark cantilever problem and the case study domain.

6.5.1 Compliance Constraint

The compliance constraint is calculated with the formula 5.5 described in chapter 5.4.3. The initial compliance is calculated at the initiation phase of the algorithm and is the compliance for a filled domain. The a_c factor is introduced to relax this constraint and to allow the algorithm to find a structure with less material.

Compliance refers to the general stiffness of the structure. Compared to the other constraints a designer has more freedom in choosing a value for how to relax this constraint, as it guides the algorithm to a stiff structure. As long as the stress constraint and or deformation constraint are not exceeded a structure with a high compliance can be as sufficient as that of a structure with a lower compliance. This becomes especially important for trying to minimize the mass of a structure, as a strict compliance constraint might result in a structure that can be further minimized within the other constraints.

Image 6.17 shows the difference in results for different a_c values. It can be seen that when a stricter a_c value is chosen the structure is more robust with thicker sections, a higher volume and more grey areas. Increasing the a_c factor results in thinner sections and a more discreet structure.



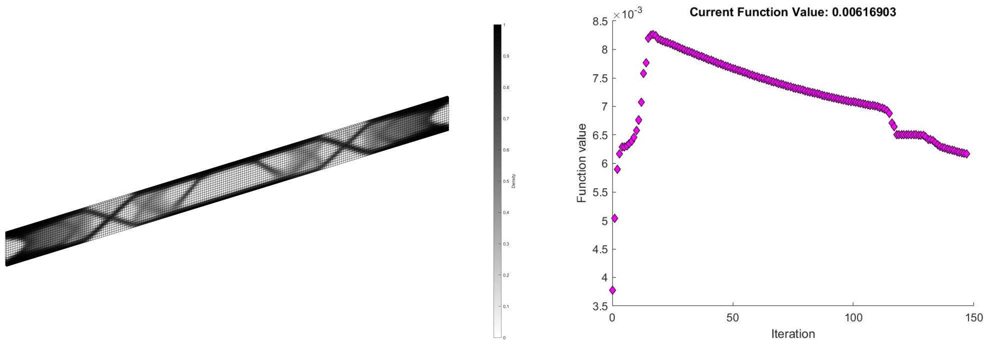
(a) ac = 1.5 **(b)** ac = 2.0

(c) ac = 2.5 **(d)** ac = 3.0

Figure 6.17: Results volume optimization with compliance constraint

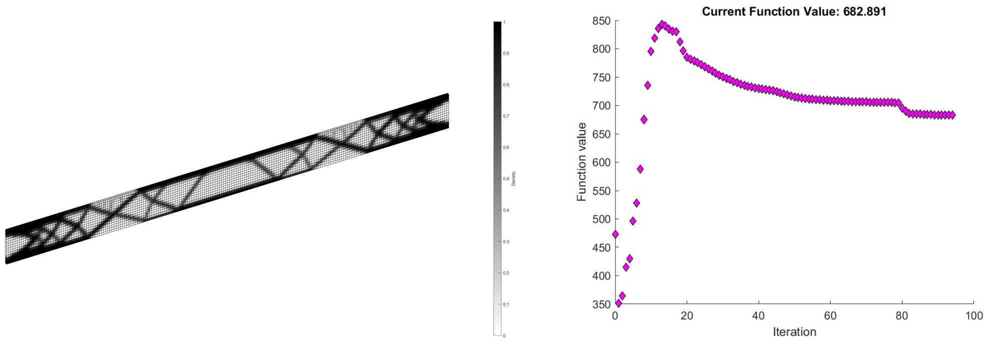
6.5.2 Principal Stress Constraint

As mentioned in section 5.4.3, every element is checked for both the maximum and minimum principal stress value. Optimizing the benchmark domain for volume with only a stress constraint will result in the algorithm removing all material. As a material can not be stressed if it is not there. Therefore, to get insight into how the q parameter influences the result and convergence of the algorithm, the calculation needs to be combined with compliance constraint. Optimizing the domain for volume with a compliance constraint (a_c factor of 2) and a constraint for both the maximum (tensile) and minimum (compressive) principal stress with a q factor of 2.5 results in a structure with a lot of grey area. With the stress relaxed this far, the intermediate densities are not penalized enough which results in this non-discrete geometry with a lot of grey values. Increasing the q value to 2.8 causes the optimization to converge in fewer iterations and results in a more discrete geometry.



(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.18: Results volume optimization with compliance and stress constraint constraint ($a_c = 2; q = 2.5$)



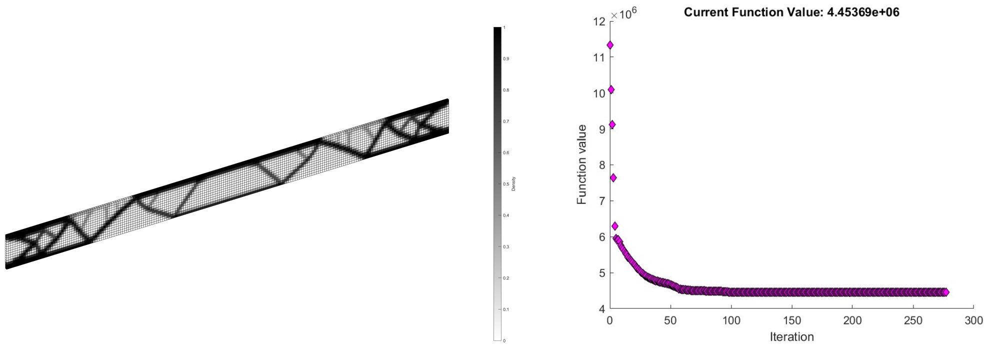
(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.19: Results volume optimization with compliance and stress constraint constraint ($a_c = 2; q = 2.8$)

6.5.3 Deflection Constraint

Optimizing the benchmark domain with a volume objective and only a deflection constraint results in the removal of all material. This is different from the results obtained by Koniari (2022) and can be explained by the gradient of the constraint. The analytical gradient of the displacement constraint is mainly filled with zeros, except for the positions in the vector that correspond to the partial derivative over the density of the elements that connect to the critical point. This results in all the material moving to zero density, even though the displacement constraint is not met in the last iterations. When the gradient is calculated numerically a similar result as Koniari (2022) is obtained, further confirming that the change is caused by the gradient of the constraint.

The fmincon function only allows for either including all gradients or all gradients being computed numerically. It is not possible to optimize with a mix of the two. When the gradients are calculated numerically the computational time needed for a 3D domain becomes infeasible. While it is possible to check the maximum deflection after the geometry is computed, this would add an extra step to the process. Furthermore, this could potentially result in geometries that are insufficient in their structural behaviour. As a result, it is preferred to include the deflection constraint, even if it might not be as accurate as a numerical gradient. Optimizing the case study domain with a volume objective, deflection constraint and compliance constraint with analytical gradients for all results in the geometry as seen in figure 6.20. The geometry is similar to that obtained from an optimization with just the compliance constraint. From this, it is concluded that the gradient of the deflection does not negatively impact the calculation when it is combined with other constraints.

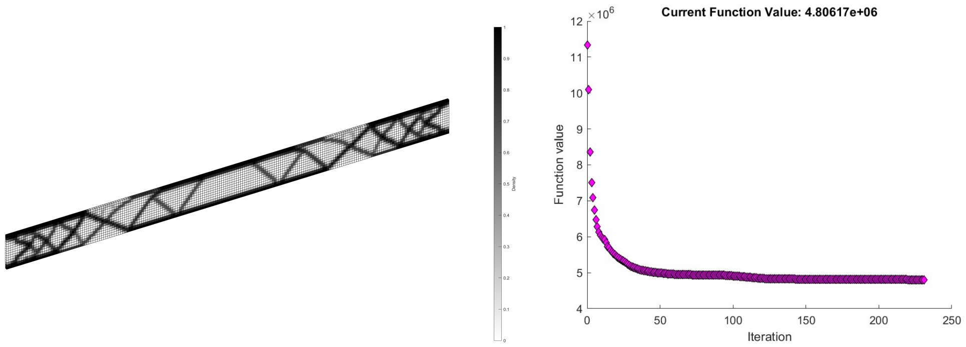


(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.20: Results volume optimization with compliance and displacement constraint ($a_c = 2.5$)

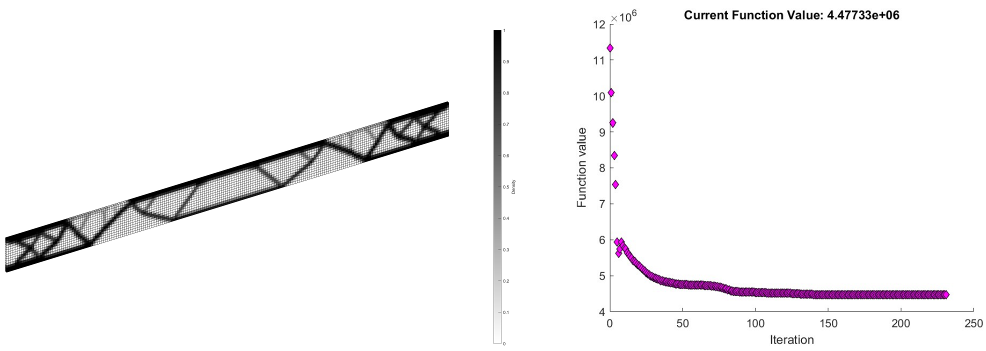
6.5.4 Annealing Constraint

Optimizing the case study domain for volume with a compliance and annealing constraint with a a_v factor of 0.75 results in the geometry as can be seen in figure 6.21. The geometry consists of a lot of thin sections and grey areas. Increasing the a_v factor to 0.9 results in the geometry as seen in figure 6.22. This geometry has less but thicker members.



(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.21: Results volume optimization with compliance and annealing constraint ($a_c = 2.5$ $a_v = 0.75$)

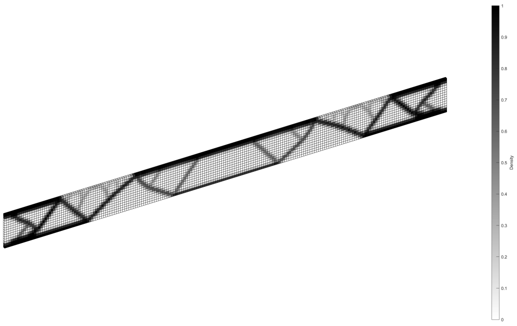


(a) Result Geometry **(b)** Plot of function value per iteration

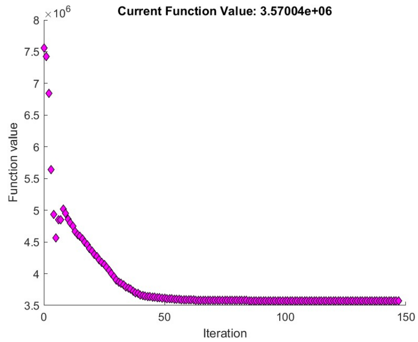
Figure 6.22: Results volume optimization with compliance and annealing constraint ($a_c = 2.5$ $a_v = 0.9$)

6.5.5 Combining Constraints

Combining all constraints with an initial volume of 0.6, a_c factor of 3.0, q factor of 2.8 and av of 0.9 results in a geometry that converges in 180 iterations. Reaching a final volume percentage of 28.39%. The deformation at the critical node is -0.2457 millimetres. The maximum tensile stress in the structure is 0.0048 kN/mm^2 and the maximum compression stress in the structure is -0.0048 kN/mm^2. The geometry is similar to that obtained by Koniari (2022), with slightly lower tensile and compression values. [5]



(a) Result Geometry



(b) Plot of function value per iteration

Figure 6.23: Results volume optimization with all constraints

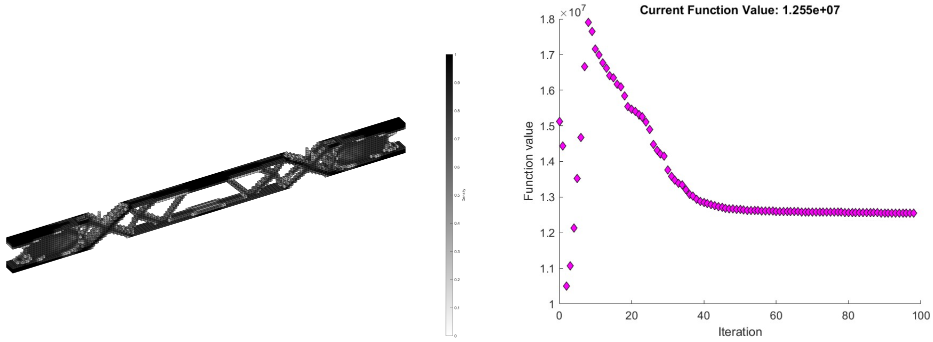
[5] The difference in deformation and stress is caused by Koniari (2022) using a a_c value of 4.0 instead of 3.0

6.6 Volume optimization 3D Beams

Directly scaling the optimization to the three-dimensional case study, gave similar issues as noted for the compliance optimization. Thus, again the algorithm was tested on the smaller three-dimensional case study of the beam.

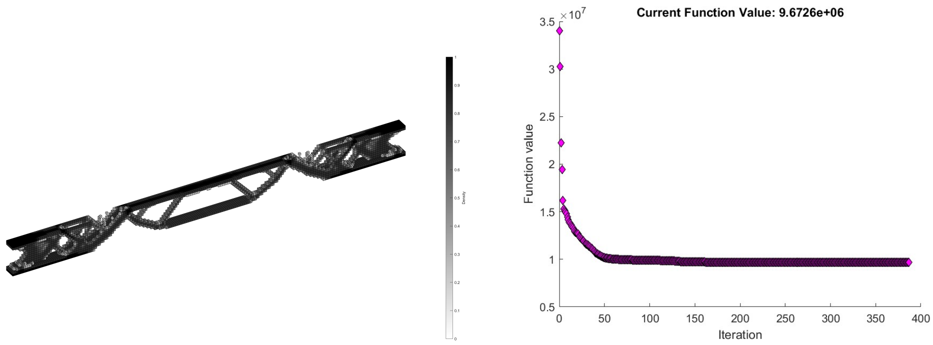
6.6.1 Compliance constraint

The algorithm has been tested for both a a_c of 2.5 (figure 6.24) and 3.5 (figure 6.25). For the optimization with a compliance constraint of 2.5 the geometry assumes an I-shaped configuration with flanges and a web, as opposed to the truss-like structure observed in the two-dimensional optimization. The optimization with a compliance constraint of 3.5 results in a geometry that resembles a truss near the supports of the structure, similar to the geometry seen in the two-dimensional optimization.



(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.24: Results 3D Beam volume optimization with compliance constraint (a_c =2.5)

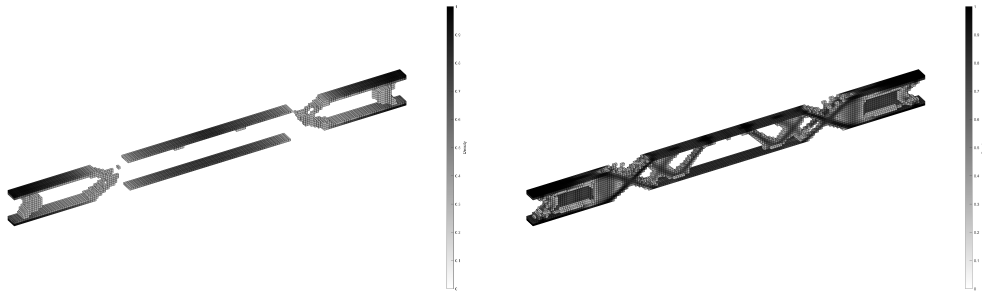


(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.25: Results 3D Beam volume optimization with compliance constraint (ac =3.5)

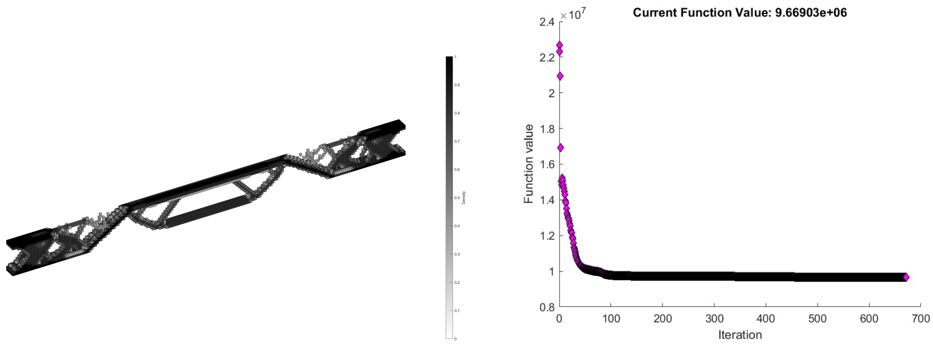
6.6.2 Principal Stress Constraint

Optimizing the beam domain with a volume objective, a compliance constraint ($a_c = 3.5$), an initial volume of 30% and a q value of 2.5 and 2.8 does not converge to discreet geometry (figure 6.26. Increasing the initial volume does give converged geometries, as can be seen in figure 6.26. These images show the result of the volume optimization with a compliance constraint ($a_c = 3.5$), an initial volume of 60% and a q value of 2.8.

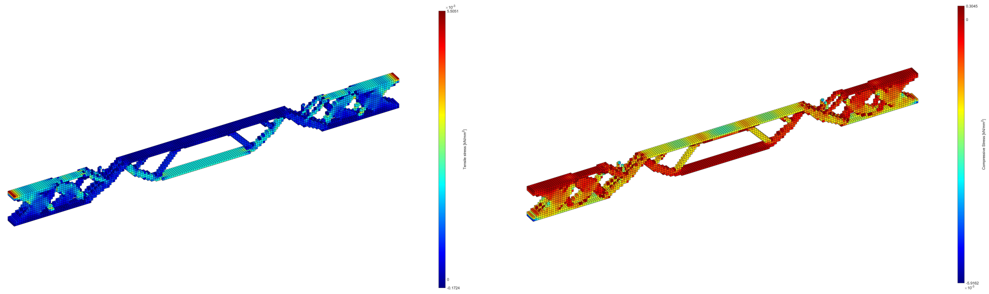


(a) Result Geometry $q = 2.5$ **(b)** Result Geometry $q = 2.8$

Figure 6.26: Results 3D Beam volume optimization with stress and compliance constraint $(ac = 3.5)$



(a) Result Geometry **(b)** Plot of function value per iteration

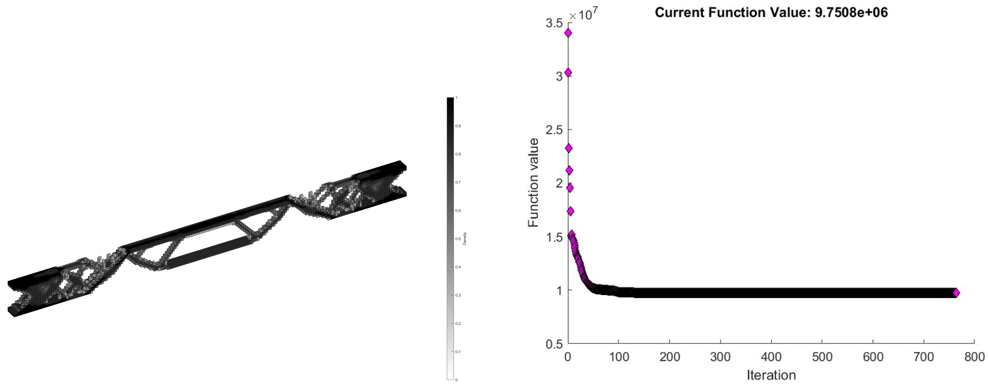


(c) Tensile Stress Graph **(d)** Compression Stress Graph

Figure 6.27: Results 3D Beams volume optimization with stress and compliance constraint $(q = 2.8$ $ac = 3.5$ $v_{initial=0.6})$

6.6.3 Annealing Constraint

Optimizing the beam domain with a volume objective, a compliance constraint $(a_c = 3.5)$, an initial volume of 30% and a a_v value of 0.9 results in the geometry as shown in figure 6.28. The obtained result is very similar to both the volume optimization with a compliance and stress constraint.

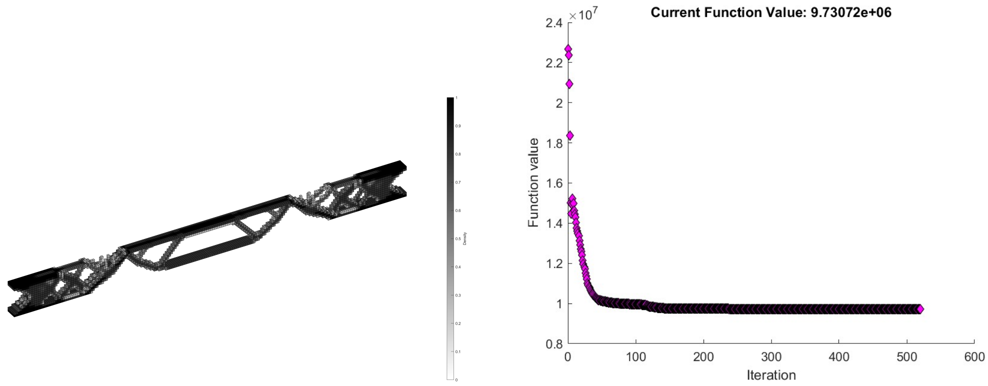


(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.28: Results 3D Beam volume optimization with compliance constraint and annealing constraint $(ac = 3.5\ av = 0.9)$

6.6.4 Combining Constraints

Combining all constraints with an initial volume of 0.6; ac value of 3.5; q factor 2.8; and av value of 0.9 results in the geometry as shown in figure 6.29.



(a) Result Geometry **(b)** Plot of function value per iteration

Figure 6.29: Results 3D Beam volume optimization with all constraints

6.7 Conclusion

Analysing the results of the tests conducted reveals some key findings. Firstly, they demonstrate that the analytical gradients integrated into the constraint function yield comparable results as those obtained using numerical gradients.

Secondly, the results revealed that certain constraint conflicts when incorporated simultaneously into the optimization process. An instance of this is the conflict between the annealing and stress constraint. Which seems to be the cause of instances in which the optimization converges to an infeasible point. This interaction between constraints demonstrates the need for careful constraint definition and selection.

Furthermore, the a_c value for the compliance constraint has a critical role in the volume optimization. When set correctly, this parameter ensures that the optimization process does not overly constrain the design. Allowing the algorithm to utilize the available material fully.

Lastly, comparing the two and three-dimensional optimization, the first signs arise that the truss typology seen in the 2D optimization might not be the most optimal material distribution for a three-dimensional design.

7
Design Exploration Case Study

This chapter focuses on the design of the case study. The first section will focus on the analysis of different optimized geometries. The next section will discuss the similarities and differences observed between the various calculations. Lastly, the results will be discussed, suggestions for further design exploration will be given and a choice will be made on the geometry for the case study application.

7.1 Analysis of the Optimized Geometries

While the full width of the bridge is 2.4 meters, two different domains are considered for the final design. One cuts the design into 3 sections to reduce the computational time and increase the residual safety of the structure (figure 7.1). The other design option is to optimize the full length, width and height of the bridge as one component (figure 7.2).

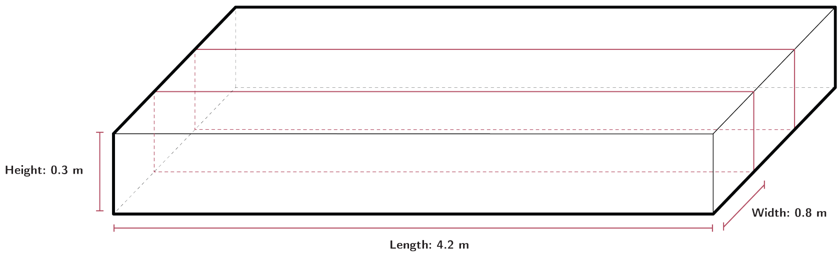


Figure 7.1: Dimensions Split Domain

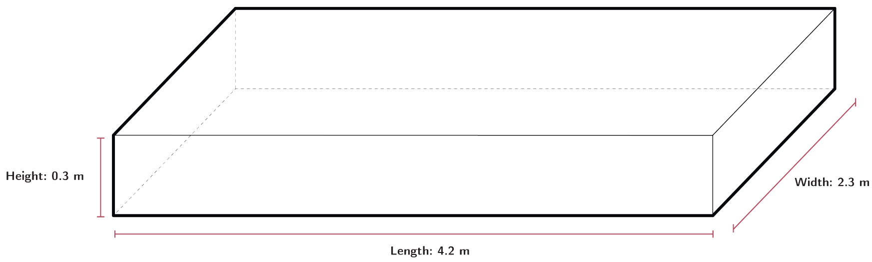


Figure 7.2: Dimensions Complete Domain

For the case study, four different setups were considered (table 7.1). These design options consist of both a volume and a compliance optimization of the design split into three components. Additionally, both a volume and compliance optimization of the full domain width were calculated. The next paragraphs analyse the results of each of these setups and compare them to see which one gives the desired result for the case study design.

In addition to the optimizations discussed in this chapter, extra experiments were conducted. These experiments encompassed optimizing the domain using soda lime glass instead of borosilicate, as well as implementing a change in boundary conditions. Further details and results of these alternative design options can be found in Appendix A.5.

Design Option	Objective	Constraints	FE size	$v_{initial}$	v_{frac}	a_c	q	a_v
Split	Volume	ALL	20 mm	0.6	-	3.5	2.8	0.9
Split	Compliance	ALL	20 mm	0.3	0.3	-	2.5	0.9
Complete	Volume	ALL	40 mm	0.6	-	3.5	2.8	0.9
Complete	Compliance	ALL	40 mm	0.3	0.3	3-	2.5	0.9

Table 7.1: Different setups for following calculations

7.1.1 Split Domain Optimized with a Volume Objective

The generation of the split domain, optimized with a volume objective, was conducted using a finite element size of 20mm. The calculation was executed on the DHPC and, unfortunately, it timed out after 24 hours before convergence was reached. During this time, the algorithm iterated 274 times, with an average iteration time of 4 minutes and 30 seconds. It reached a stable f value of 57.976 00^6 mm^3. Indicating that the volume was reduced to 23% of the full volume. Image 7.3 shows the geometry reached at this iteration. The deflection of the structure at the critical node is 0.2370 mm. The maximum tensile stress within the structure registered at 0.0047 Kn / mm2, while the maximum compressive stress is -0.0047 kN/ mm2.[1]

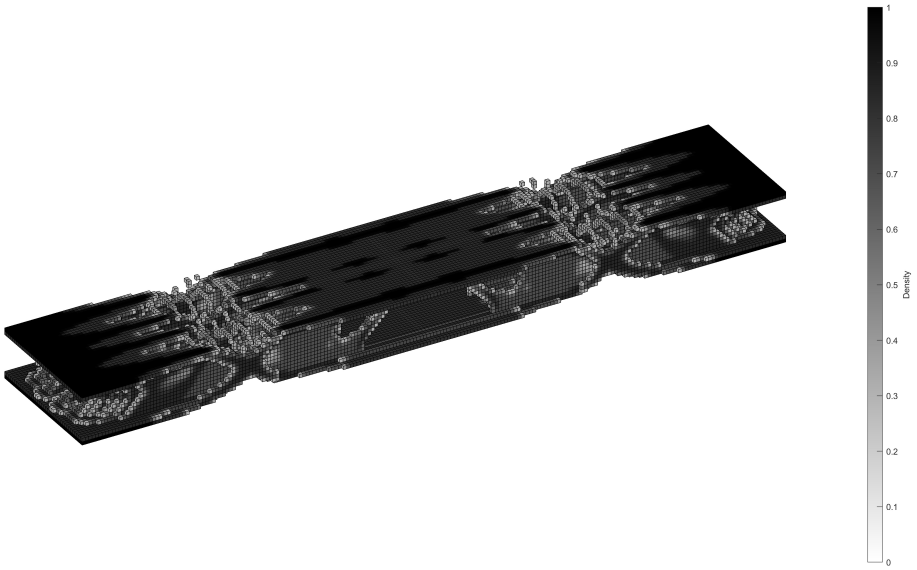


Figure 7.3: Result Volume Optimization, third of domain

The final geometry is symmetric in the length and width. The structure can be simplified as four I beams that are repeated in the structure's width. Image 7.4 shows a diagrammatic analysis of the structure. The structure can be subdivided in three parts: the side sections, the middle section and the connection between those two sections. Closest to the supports, the structure follows the topology of an I beam with a top and bottom flange. The outer two webs do not run parallel to the structure length but diverge slightly to the outside. The flanges thin out at around a quarter of the length. Here the web changes into a thicker X-shaped connection to the middle section. At the middle section, the web gets further reduced to just a diagonal bar supporting the top flange. Small arches are formed, running parallel to the width of the structure, connecting the I sections with each other.

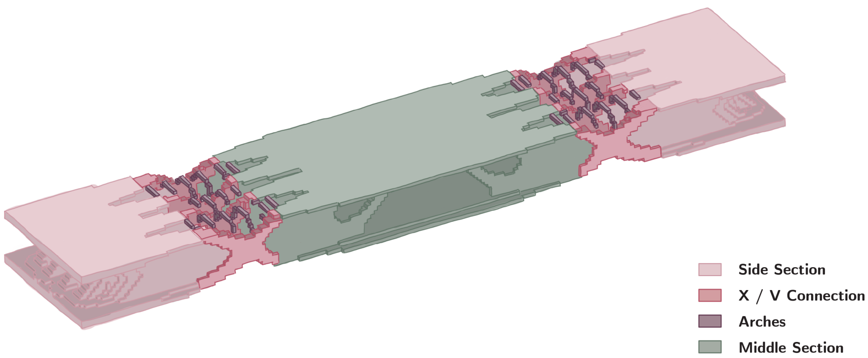


Figure 7.4: Part of the optimized geometry, simplified to show the different typologies

[1]The structural results from the design exploration can be found in Appendix A.6

7.1.2 Split Domain Optimized with a Compliance Objective

The generation of the split domain, optimized with a volume objective, was conducted using a finite element size of 20mm. The calculation was executed on the DHPC and, unfortunately, it timed out after 8 hours before convergence was reached. During this time, the algorithm iterated 157 times with an average iteration time of 3 minutes and 4 seconds. At this iteration, an f value of 1.7209 is reached. The initial compliance of the same structure with full material is 0.7504. An increase in the compliance value signifies a decrease in the structural stiffness. The compliance reached is 2.3 higher than the initial compliance. This signifies that with 30% of the volume a structure can be reached that has around 42% of the structural stiffness of the full domain. Image 7.5 shows the geometry reached at this iteration. The deflection of the structure at the critical node is 0.2196 mm. The maximum tensile stress within the structure registered at 0.0047 Kn / mm2 while the maximum compressive stress is -0.0047 kN/ mm2.

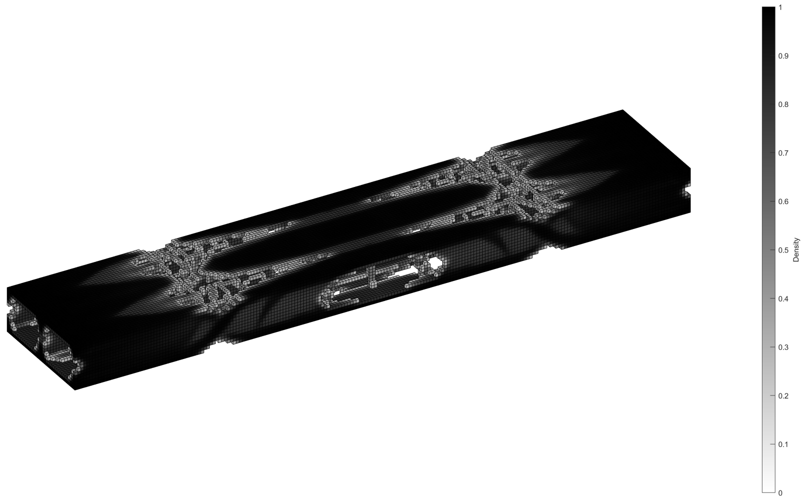


Figure 7.5: Result Compliance Optimization, split domain

The final geometry is symmetric in the length and width. The structure can be simplified as two box sections. Image 7.6 shows a diagrammatic analysis of the structure. The structure can again be subdivided into three parts: the side sections, the middle section and the connection between those two sections. Closest to the supports, most of the material is placed at the domain's edges, forming a box section. The middle section follows a similar typology but with more material reduction at the top and bottom of the structure. The side and middle are connected by an X-shaped connection. Of this connection, the member that is under compression is notably thicker than the tension member. Similar to the volume optimization, there are arches that span in the void at the top. These arches branch and form an organic shape.

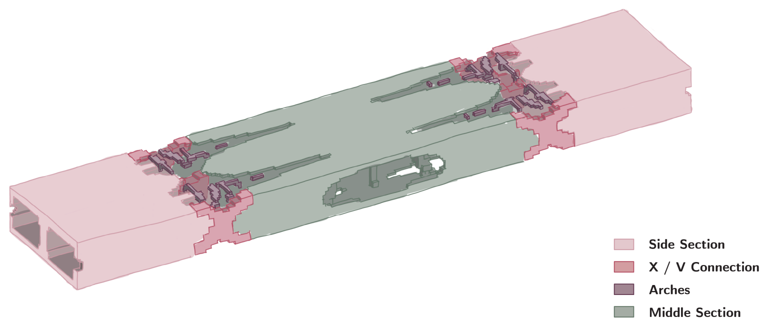


Figure 7.6: Part of the optimized geometry, simplified to show the different typologies

7.1.3 Complete Domain Optimized with a Volume Objective

The generation of the complete domain, optimized with a volume objective, was conducted using a finite element size of 40 mm. The algorithm converged after 138 iterations with an average iteration time of 30 seconds. The converged geometry is shown in image 7.7 image has a volume percentage of 28%. Important to note is that for this calculation the minimum cross section size was increased to 120. Consequently, this also resulted in a higher maximum cross-section as these values can not be too close together. [2] The deflection of the structure at the critical node is 0.2358 mm. The maximum tensile stress within the structure registered a 0.0035 Kn / mm2, while the maximum compressive stress is -0.0036 kN/ mm2.

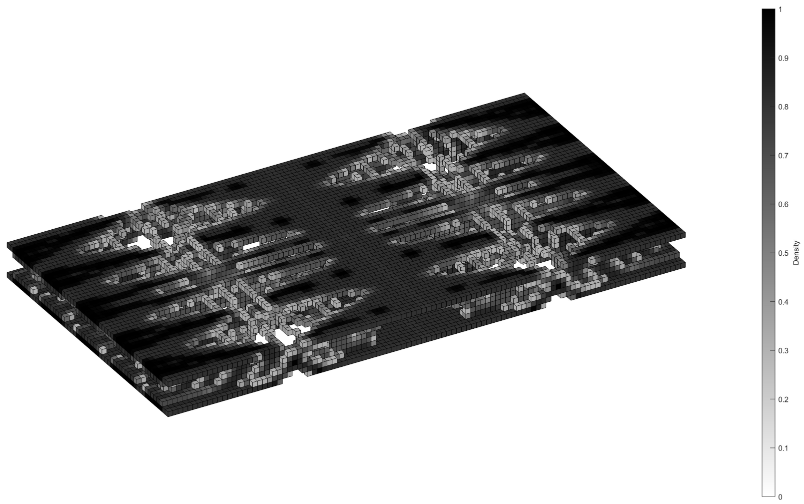


Figure 7.7: Result Volume Optimization, Large Domain

The optimized geometry is symmetric on two axes and can be further subdivided into sections as shown in Image 7.8. Closest to the supports the optimization forms something akin to a row of I sections with a top and bottom flange connected by a vertical web. The flanges thin out at around a quarter of the length of the structure. Here the structure changes to two different connection typologies. At the most outer of the I beams, the side and middle sections are connected by a X-shaped connection. The middle of the I beams is connected to the middle by a V connection. Here only the compression member of the X shape remains. The middle section consists again of a top and bottom flange, but instead of being fully connected by a web, there are diagonal supports. In the voids at the top of the structure, the algorithm forms arches that connect the I sections.

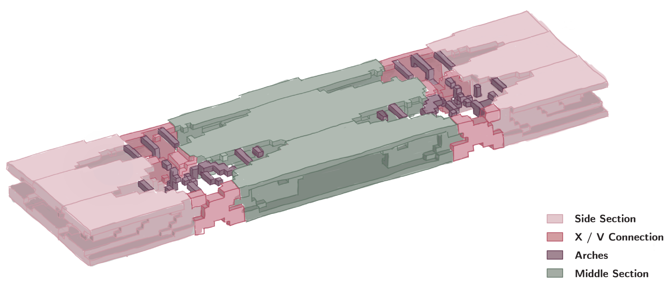


Figure 7.8: Part of the optimized geometry, simplified to show the different typologies

[2] Initially, an attempt was made to perform volume optimization using a 20-millimeter mesh size. However, it was observed that each iteration with this finer mesh required 30 minutes for calculation. Consequently, a decision was made to adopt a coarser mesh to facilitate more calculations.

7.1.4 Complete Domain Optimized with a Compliance Objective

The generation of the complete domain, optimized with a compliance objective, was conducted using a finite element size of 40 mm. The algorithm converged after 310 iterations with an average time per iteration of 1 minute and 10 seconds. The converged geometry is shown in image 7.9 and has a compliance of 5.5620. The full domain would have a compliance of 2.2211. As described in the previous sections, an increase of the compliance value signifies a decrease in the structural stiffness. The reached compliance is 2.5 times higher than the initial compliance. This signifies that with 30% of the volume, a structure is reached that has around 40% of the structural stiffness compared to the usage of 100% of the volume. [3] The deflection of the structure at the critical node is 0.2505 mm. The maximum tensile stress within the structure registered a 0.0037 Kn / mm2, while the maximum compressive stress is -0.0039 kN/ mm2.

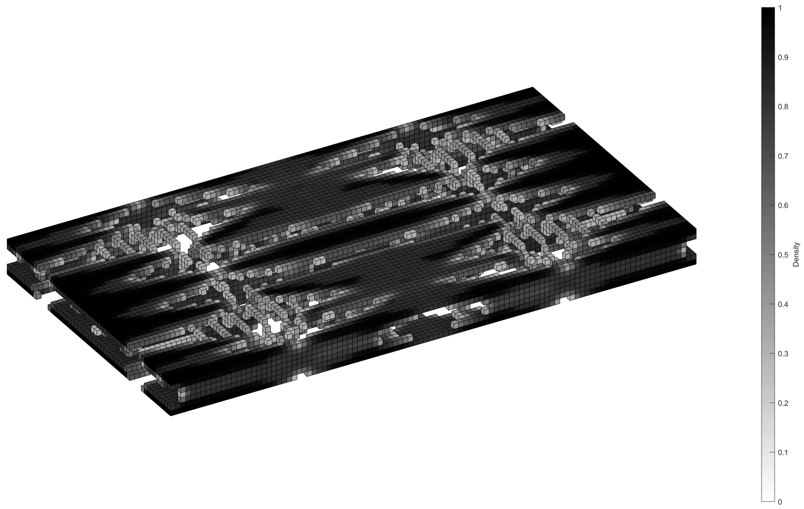


Figure 7.9: Result Volume Optimization, Large Domain

The optimized geometry is symmetric in its length and width, however, each quadrant is isometric and can be further subdivided in sections as shown in image 7.10. Similar to the compliance optimization of the split domain, material is placed on the long sides of each quadrant. However, the middle section is more organic in shape. Which is particularly clear to see in the top view (figure: 7.12. this organic shape follows from the webs of the middle I sections that cross diagonally through the width of the structure. At the long sides, the connection between the middle and side section does not form an X or V-shaped connection, but most of the domain is filled. The connections in the middle of the structure do form the V-shaped connection as seen in the 2D optimizations. In the voids at the top of the structure, the optimization forms arches that connect the I section. These arches have a consistence spacing of 0.12 meters.

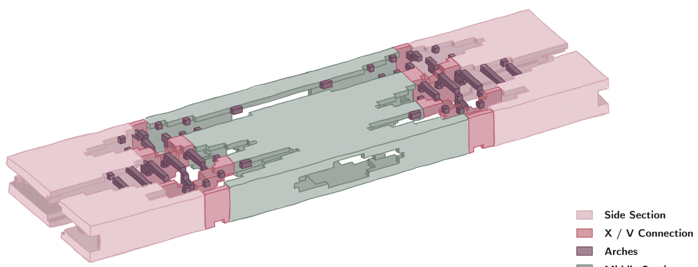


Figure 7.10: Part of the optimized geometry, simplified to show the different typologies

[3] One iteration of the full domain takes approximately 15 minutes, meaning that the calculation can take anywhere from a day to three days to complete, this should be attempted for the final report.

7.2 Comparison Different Design Options

Table 7.2 shows an overview of the most important results gathered from the optimizations.

Domain	Objective	Final volume	Initial compliance	Final compliance	Deformation	Maximum Tensile Stress	Maximum compressive stress	Time iteration
Split	Volume	23%	0.7456	2.0182	-0.2370	0.0047	-0.0047	4 min 30
Split	Compliance	30%	0.7456	1.7209	-0.2194	0.0047	-0.0047	3 min 4
Complete	Volume	28%	2.2211	7.2425	-0.2358	0.0035	-0.0036	0 min 30
Complete	Compliance	30%	2.2211	5.5620	-0.2505	0.0037	-0.0039	1 min 10

Table 7.2: Results setups calculations

7.2.1 Comparison Complete and Split Design

Several differences can be observed between the outcomes of the varying domain sizes. Certain geometric features are seen in both the smaller and larger domains. For example, all designs contain arches parallel to the structure's width. In both cases, the material is predominantly distributed at the top and bottom of the design domain, creating a structure that can be simplified to flanges supported by webs oriented in the structure's length.

Furthermore, in both compliance optimizations, the algorithm develops a box section, concentrating the material to the outer boundaries of the domain. Both domain sizes exhibit asymmetry in the calculated quarter. The larger domain exhibits this asymmetry more pronounced, as the webs curve diagonally through the width of the structure. The geometry of the smaller compliance domain more clearly shows the x-connection on these outer sides, with more material concentrated around the compression member of this connection. Both optimizations reach an approximate 2.30-2.50% increase of the compliance value compared to that of a fully filled domain, with a 70% decrease in material. The smaller domain performed marginally better on the final compliance.

Both volume optimizations show a preference for placing the material more toward the middle of the structure and connecting the side and middle section with an x-connection. This connection is slightly thicker than the rest of the web to which it is connected. The volume optimization of the smaller domain attains a material reduction of 77%, outperforming the larger domain where just a 72% reduction is reached. Additionally, the larger domain exhibits lower peak stresses than its smaller counterpart, although it experiences a greater maximum deflection.

7.2.2 Comparison Two-Dimensional Optimization and Three-Dimensional Optimization

A comparative analysis of the 2D geometries obtained in chapter 6.3.5 and 6.5.5 and the previously analysed 3D structures reveals several notable distinctions (figure 7.11). Firstly, the three-dimensional optimization shows a clear preference for a web and flanges geometry, with some truss-like elements linking sections together. The two-dimensional optimization, on the other hand, fully resembles an optimized spacial truss.

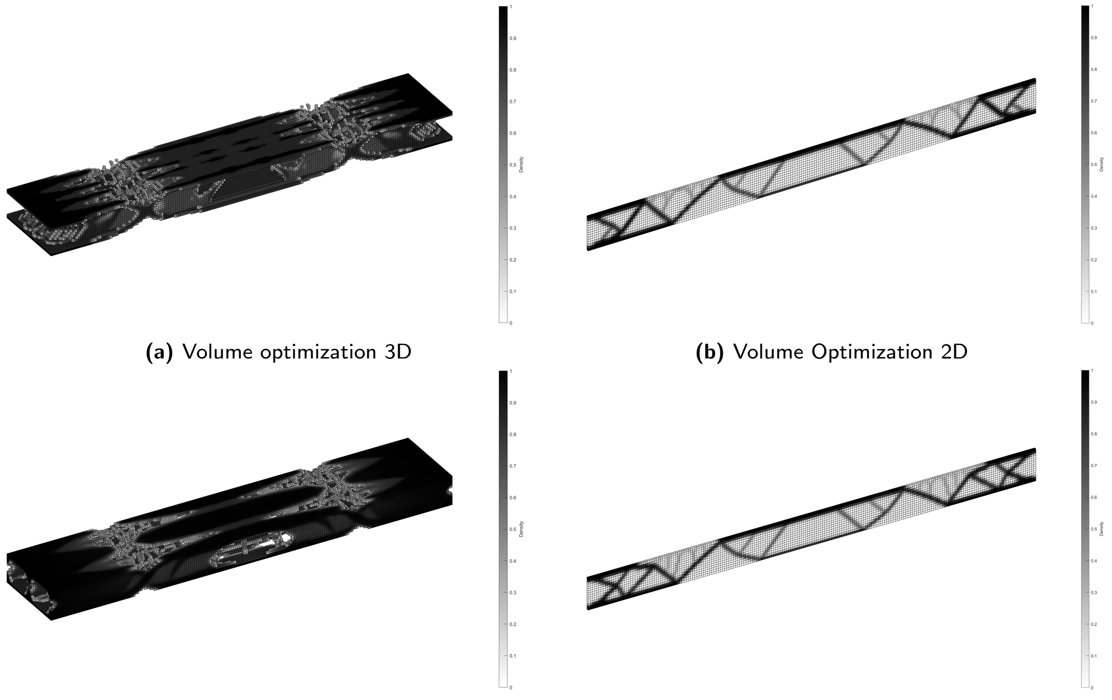
Interestingly, the 2D optimizations can clearly be divided into three parts, two cantilevers on the side and a simply supported beam in the middle. This distinction is less clear for the 3D optimization. It has an x-shaped connection instead of the V shape seen in the 2D optimizations.

Furthermore, the 2D optimizations show what Koniari (2022) called 'nerves', which are the light grey voxels (density 0.3-0.6) supporting the top loads where most of the material is reduced. In the 3D optimization, the algorithm forms small arches at these locations. Demonstrating the importance of including the width in the optimization.

Table 7.3 gives an overview of the results of the two-dimensional calculation. The structural performance of the different optimizations further indicates that the 3D optimization is able to find a more efficient material utilization than the 2D geometry.

The smaller of the two three-dimensional volume optimizations yields a 5% greater reduction in material usage compared to its two-dimensional counterpart. While the stress distribution and maximum deflection remain similar for both domains. Furthermore, comparing the compliance reached for both design domains shows that the 3D geometries have a compliance of 2.3-2.5 times the initial compliance, while the 2D structure has that of 3.0-3.3 times the initial compliance. Indicating that the 3D optimization is capable of globally reaching a stiffer structure with the same amount of material while keeping similar peak stresses and deformation.

In conclusion, only one of the volumes derived from 3D optimization is significantly lower than its 2D counterpart. However, all 3D geometries perform better structural stiffness. In addition, the 3D optimization offers an improved insight in the spatial distribution of material. This does however come at the price of a more complex geometry, and thus more complex manufacturing process.



(a) Volume optimization 3D

(b) Volume Optimization 2D

(c) Compliance Optimization 3D

(d) Compliance Optimization 2D

Figure 7.11: Overview geometries obtained from the two-dimensional and three-dimensional optimizations

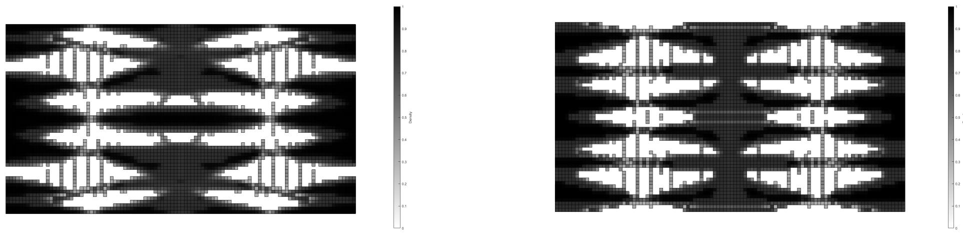
Domain	Objective	Final volume	Initial compliance	Final compliance	Deformation	Maximum Tensile Stress	Maximum compressive stress
2D case study	Volume	28.39%	0.0384	0.1282	-0.2457	0.0048	-0.0048
2D case study	Compliance	30%	0.0384	0.1151	-0.2334	0.0042	-0.0042

Table 7.3: Results 2D calculations

7.3 Comparison Objectives

Several differences were observed when comparing the outcomes of the compliance and volume optimizations conducted for the same domain size. Firstly, the volume-optimized geometry exhibits a larger degree of symmetry. Whereas, the compliance-optimized geometry is more organic in its shape (figure 7.12). Furthermore, the volume optimization introduces a slight offset away from the domains edge in the placement of material, creating a profile that is similar to that of an I-beam. While the compliance optimization prefers to allocate the material close to the domain's edge, essentially creating a U-section.

Moreover, the compliance optimization adheres to the prescribed volume constraint of 0.3 whereas the volume optimization is able to reduce the final volume by 5% more. The peak stresses in both optimizations are similar, the maximum deflection in the structure is slightly lower in the compliance optimizations. These differences underline the influence of the choice in objective on the resulting geometry.



(a) Compliance Optimization

(b) Volume Optimization

Figure 7.12: Topview volume and compliance optimization for full domain

7.4 Discussion

In all cases, the algorithm creates a geometry with the material predominantly distributed at the top and bottom of the design. Creating structures that can be simplified to flanges supported by webs oriented in the structure's length, with arches spanning in the structure's width.

Comparing the complete and split design optimized for volume, a notable difference is seen in the volume reduction. One possible explanation for this is the coarser mesh size chosen for the larger calculation. To be able to effectively filter for the minimum cross section size the minimum diameter needs to be three times the element size. As the element size of the larger calculation was set to 40 millimeters this resulted in a minimum cross section size of 120 millimeter. In comparison, for the finer mesh size a minimum filter size of 60 millimeter was applied, and the diameter for the annealing time was taken as 120 mm. Essentially, this result in the coarser mesh using a minimum cross-section size that is considered to be the maximum cross-section for the calculation with the finer mesh. For the compliance optimizations, there is no significant difference in performance between the split and complete design.

Comparing the outcome of the different objectives shows mainly a difference in geometry. Architecturally, the complete domain optimized for compliance has the most outspoken and unique geometrical shape. The compliance optimization exhibits a greater degree of asymmetry. This could potentially be explained by the algorithm utilizing the mirrored face in the width direction. The nodes at this face are set to be non-moving in the direction of the direction of the width of the structure. However, the compliance optimization adheres to the prescribed volume constraint. The volume optimization shows that it is possible to create a geometry that complies with all the constraints with less material usage.

Comparing the 2D optimization and 3D optimization shows that, while only one of the volumes derived from the 3D optimization is significantly lower than its 2D counterpart, all geometries perform better on structural stiffness. Indicating that there is a potential to find geometries with even lower volume by fine-tuning the volume constraint for compliance optimization. Furthermore, the 3D optimization offers an improved insight into the spatial material distribution.

Since all the results are within the allowable limits regarding structural performance and annealing time, the decision for the final shape of the slab is mainly based on the volume reduction of the results. While architecturally the compliance optimization for the complete domain might have the most spatial quality, the volume optimization for the split design performs significantly better on volume reduction. Resulting in the lightest structure of all optimizations. Furthermore, subdividing the supports of the bridge into three separate components ensures residual safety, and makes the components lighter for transportation and easier to install. Thus the final design of the slab will consist of three volume-optimized components.

8 Final Design

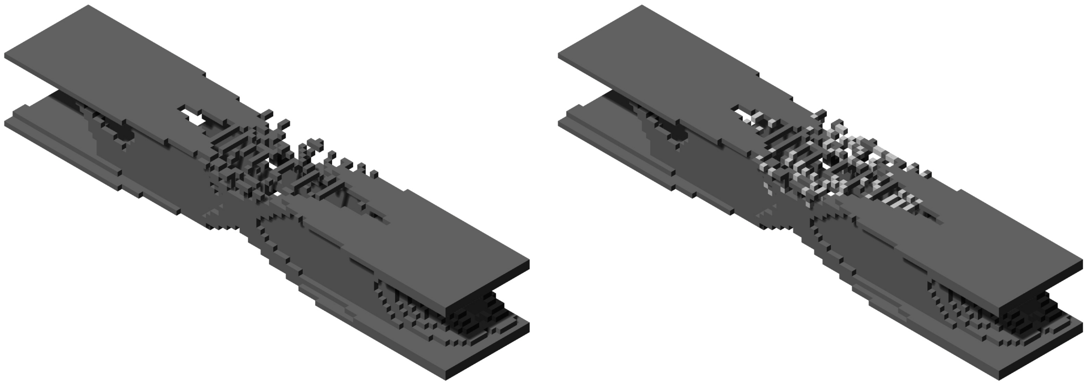
This chapter will discuss the final case study design. It will cover the steps taken for the post-processing, the structural verification of the final geometry, the strategy regarding the fabrication of the components and the building integration. Lastly, the final design will be presented in the context.

8.1 Post-Processing

As described in the previous chapter, the final design of the slab will consist out of three volume optimized components. To post-process the density map resulting from this optimization, different approaches were conducted. Firstly, experiments were conducted to find a methodology for post-process the geometry in grasshopper. However, as this was not giving the desired results, similar steps as proposed in Damen (2019) were followed.[1]

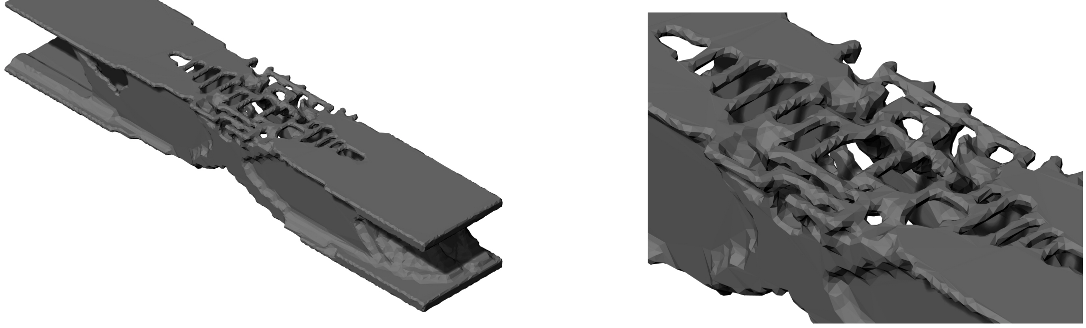
The initial step in post-processing involves the transfer of the density map to Rhino. This task is accomplished through a Grasshopper script that reads the Excel file and generates the corresponding geometry. Only the elements with a density higher then 0.3 are drawn in rhino. Image 8.1a presents the geometry as directly derived from the density map. Not all elements are connected to each other. To ensure proper connectivity, additional elements are manually added to bridge any gaps, as illustrated in mage 8.1b. The combined geometry is transfered to ANSYS spaceclaim for further processing.

"In SpaceClaim, the initial post-processing step involves applying a 'shrinkwrap' operation to the geometry, as depicted in figure 8.1. This process serves to fill any gaps in the geometry and smooth out the sharp edges of the cubes. However, it's important to note that this step can inadvertently smooth out the faces where the geometry is intended to be mirrored .To maintain flat mirroring faces, additional blocks are imported from Rhino. The geometry after this step, is the result of merging the 'shrinkwrapped' geometry with these newly introduced blocks as seen in image 8.3.



(a) Density map transferred to volumes **(b)** Geometry after filling in gaps

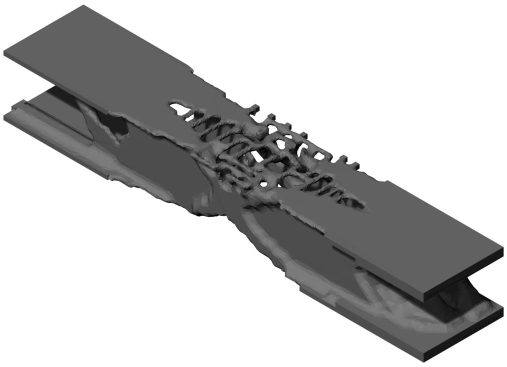
Figure 8.1: First step of the post processing, transferring the density map via grasshopper to rhino.



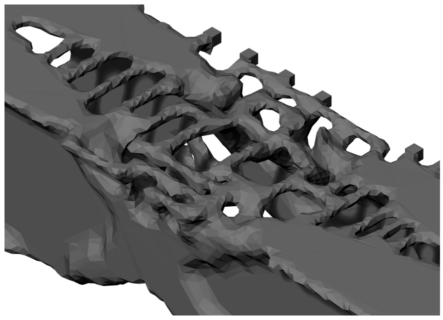
(a) Geometry **(b)** Close up

Figure 8.2: Geometry after shrinkwrapping, the faces are reduced by 80% in spaceclaim in order to create a geometry light enough to transfer to other software.

[1] To ensure that the final geometry is symmetric, only a quarter of the component is post processed. When a smooth geometry is reached, this can be mirrored to create the full component.

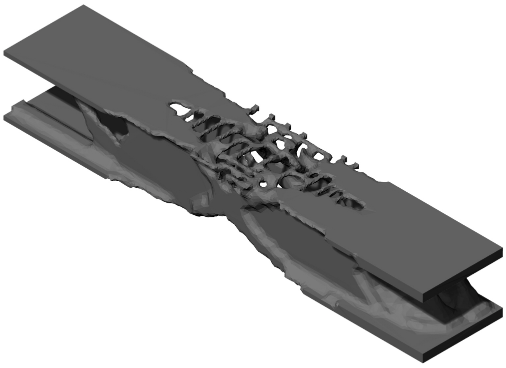


(a) Geometry

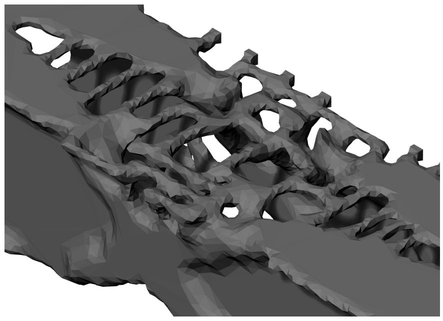


(b) close up

Figure 8.3: Geometry after merging it with volumes to redefine the connection edges, the faces are reduced by 80% in spaceclaim in order to create a geometry light enough to transfer to other software.



(a) Geometry



(b) Close up

Figure 8.4: Geometry after manually smoothing

The last phase of post-processing involves the utilization of the 'smooth' tool to address any remaining sharp corners (figure 8.4). Subsequently, the final geometry is mirrored to generate the complete component (figure 8.5) This component can be used for structural verification.

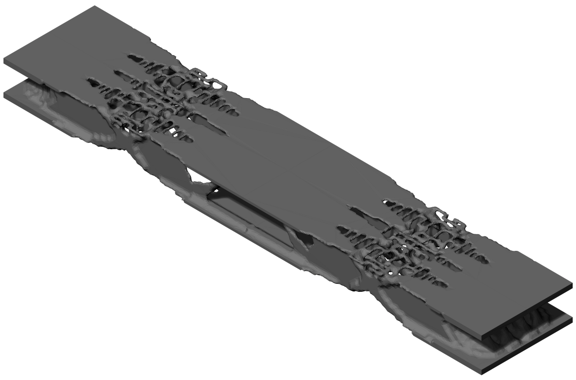


Figure 8.5: Full geometry after mirroring

8.1.1 Structural Verification

The structural performance of the final shape was verified in ANSYS. For this model one component with a additional layer of float glass is calculated. The results are summarized in table 8.1 and showcase that the deformation and principal stress values lay inside the constraints imposed on the structure. The full structural analysis can be found in Appendix A.7, this includes a check for buckling. The differences in the results of the algorithm and ANSYS can be explained by the changes in the geometry that resulted from post-processing.

	Maximum displacement mm	Maximum principal stress Kn / mm2	Minimum principal stress Kn / mm2
Matlab	-0.2370	0.0047	-0.0047
Ansys	-0.1869	0.0054	-0.0049

Table 8.1: Results structural verification

8.1.2 Discussion Post-Processing

To facilitate the transfer of the post-processed geometry to a different software application than SpaceClaim, a substantial reduction in the level of detail was necessary. This reduction results in a visibly faceted surface. Although this level of detail suffices for structural verification and accuracy, it may not be ideal for designing the mold, where a completely smooth surface is preferred.

The Grasshopper plug-in Weaverbird appears to be a promising solution for achieving a smooth geometry. It has the capability to create smooth surfaces without the triangulation being visibly apparent. The geometry generated in spaceclaim was used as a basis to explore if the component could be further post processed and resulted in the shape depicted in image 8.6. However, this plugin has a similar problem as the shrink-wrap option of spaceclaim where it is hard to isolate corners of the shape that should not be reduced.

Figure 8.6: Geometry after additional step of post-processing with Weaverbird

8.2 Fabrication

The fabrication of the glass components can be split up in two phases; the fabrication of the moulds and the fabrication of the glass component with casting. As discussed in chapter 3 the most suitable mould type for this type of component is a 3D-printed sand mould. It combines low manufacturing cost and accurate but easy fabrication of high-accuracy moulds. For the design of the mould, a few key aspects have to be taken into consideration. The first is the casting direction, it is important that the flow of the glass is such that inclusions as bubbles and cords are avoided. Furthermore, it has to be considered that the surface exposed to the air will have the largest deformations in the surface. As the top surface of the component has to be adhered to additional layers of laminated float glass, it is desired that this surface has limited deformations. Thus, the most suitable casting direction is to cast the component upside down.[2]

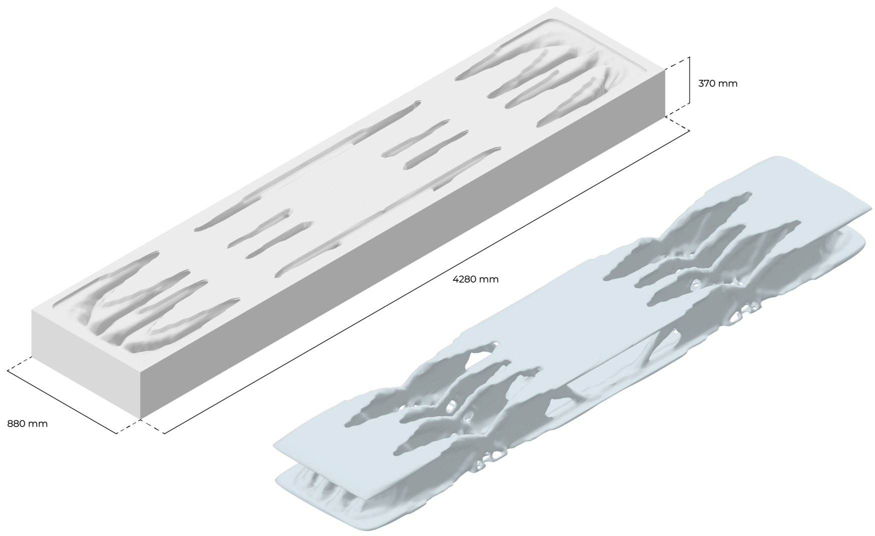


Figure 8.7: Overview casting direction and mould size

A second key aspect of 3D printing large-scale products is the treatment of the solid areas. As the component has a volume reduction of approximately 77% this means the negative of this would be 1.08 cubic meters.[3] Printing the mould with 100% density is highly time-consuming, increases the price of printing and will result in a heavy mould. Several solutions for reducing the area of solids are proposed in Stefanaki (2020). One such option is to just print a shell of the component and fill the unused areas of the mould with sand. A second approach is to replace the solid areas with an infill structure like a triangular or honeycomb grid. This way, the solid areas can be reduced to 20-25% density. A last approach is the addition of buttresses to the shell and filling in the rest of the unused areas with sand. Due to the shape of this component, the mould needs to be able to carry the weight of the top flanges. Thus, having a honeycomb infill can be the solution to reduce material usage and keep the structural integrity of the mould.

The thickness of the shell can be estimated by calculating the hydro-static pressure that is generated during the casting. From the respective equations described in Stefanaki (2020), a rough estimate for the minimum wall thickness should be 0.7cm (appendix A.8). However, this is increased to 4cm to accommodate for the large simplification that is done in this calculation. Furthermore, the bottom surface of the mould is increased to 5cm and the height is increased by 2cm to prevent any glass overflow. This results in a mould size of 4.28*0.88*0.37 meters.

Lastly, with the addition of the shell the mould size is at least 4.28*0.88*0.37 meters. The maximum dimensions that can be printed with sand is 4*2*1m. Consequently, the mould must be divided into multiple parts to fit within these printing constraints. To prevent any potential leakage at the seams of the mould pieces, an interlocking connection system is needed with extensions and undercuts (Koniari, 2022; Stefanaki, 2020).

[2]Additional air vents might need to be included in the mould design to further eliminate the chance of air bubbles.
[3]This includes the addition of the wall thickness

8.3 Building Integration

The location of the bridge is between the reading room and the exhibition spaces in the court of the British Museum. The bridge has to be installed at approximately 11 meters high. The connection detail is depicted in image 8.8. First, the metal brackets will be installed in the neighbouring walls. The glass elements can then be lifted into place with a crane. The T bracket will be brought into place to fix the components into place. After, the laminated float glass sheets are attached to the monolithic components. Lastly, the railing will be attached to the metal frames in the neighbouring walls

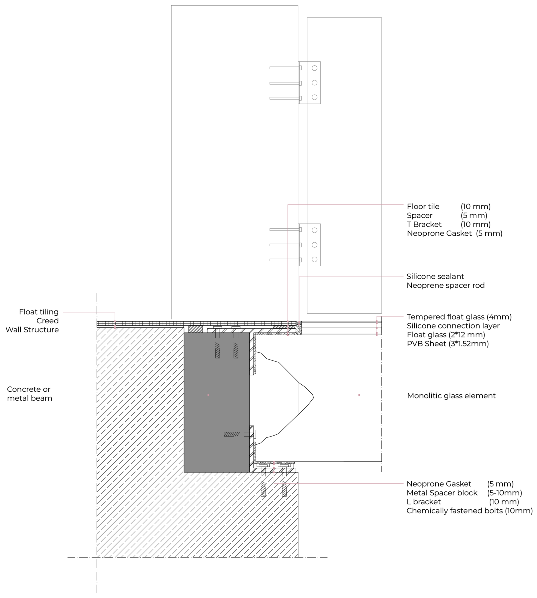


Figure 8.8: Connection Detail (Drawn scale 1.5)

8.4 Final Result

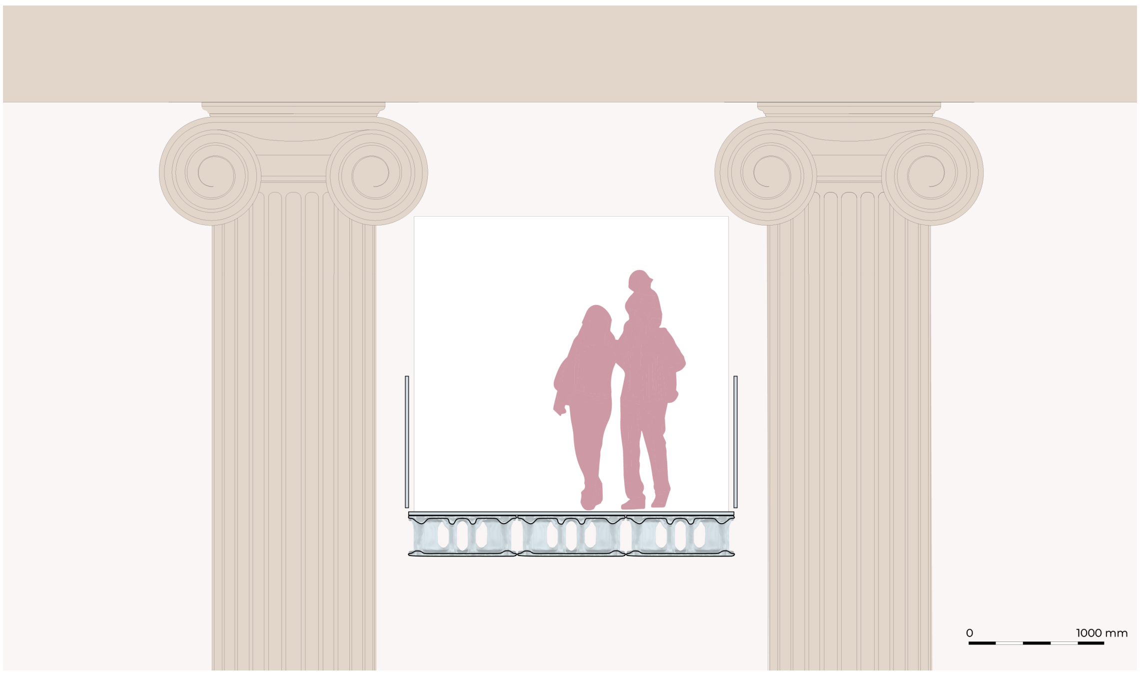


Figure 8.9: Front section

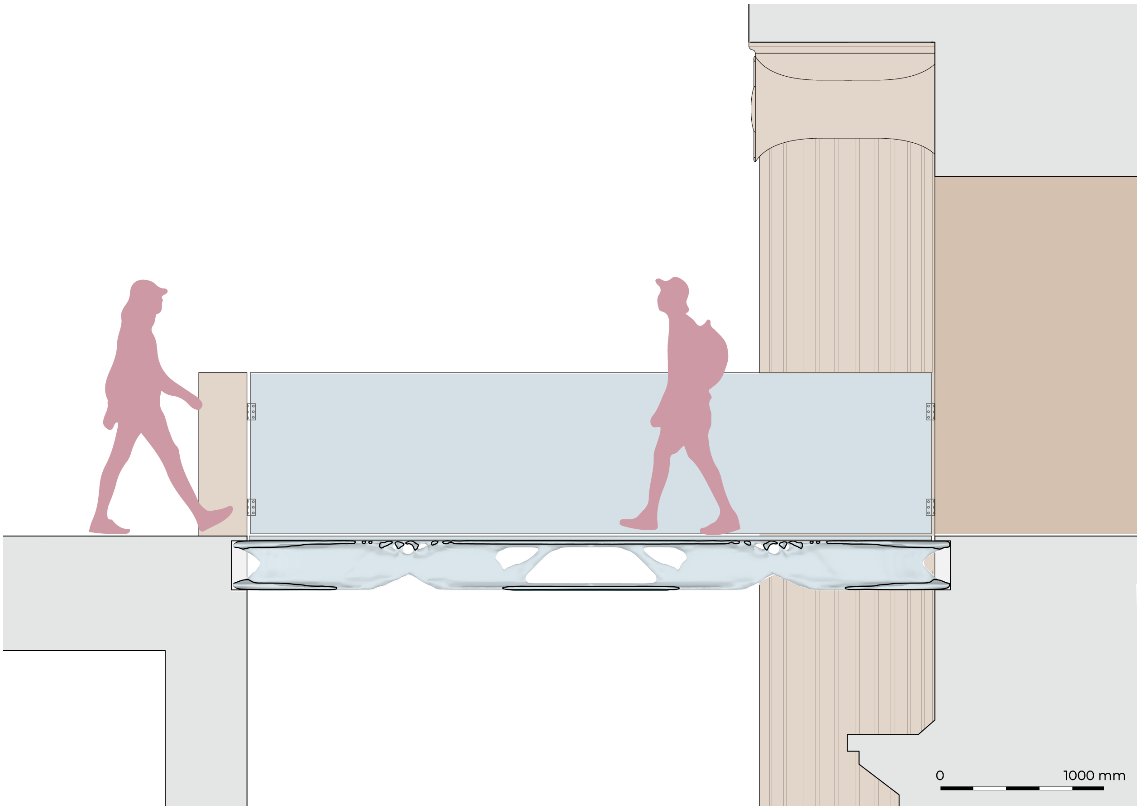


Figure 8.10: Side Section

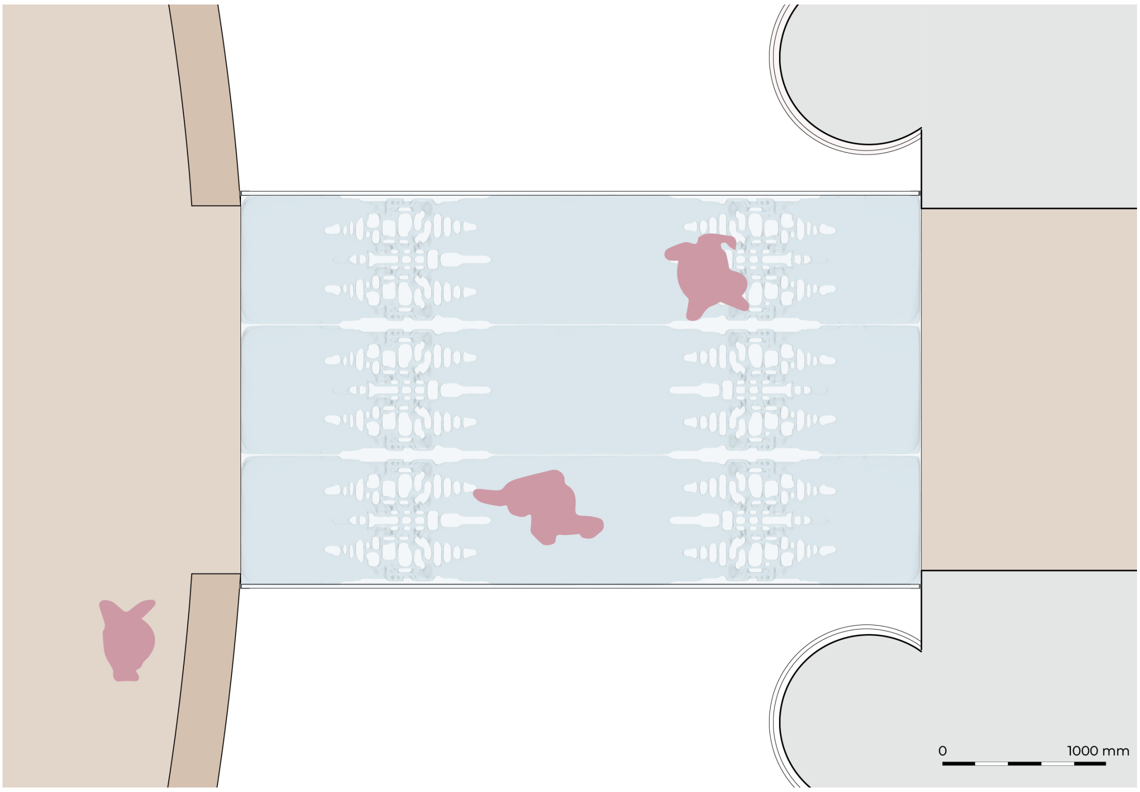


Figure 8.11: Plan View

Figure 8.12: Visualisation Final Design Front View

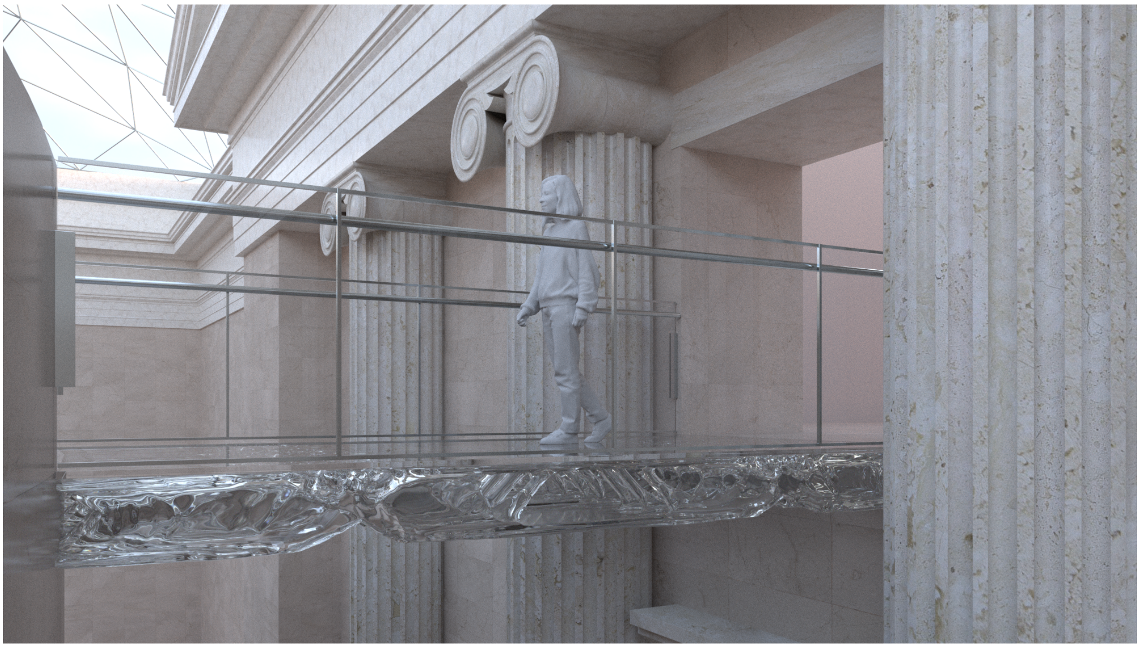


Figure 8.13: Visualisation Final Design Side View

Figure 8.14: Visualisation Final Design Bottom View

Figure 8.15: Visualisation Final Design Top View

9
Conclusion and Outlook

The final chapter will conclude on the research presented in this thesis. The research questions are answered based on the results of the research. The limitations of the research will be expressed and finally recommendations and guidelines are presented for future work within the topic.

9.1 Conclusion

At the start of this thesis the following overarching research question was defined: *What are the main aspects and limitations of a three-dimensional topology optimization for the design of massive cast glass structures?* This question was split up in several questions that can be answered based on the research and results of this thesis.

What are the design constraints posed by the chosen case study?

A glass slab was chosen as the design case. The total domain of the design was subdivided in three sections to increase the safety of the structure by introducing redundancy in case of failure. Additionally, two layers of float glass are added on top of the optimized geometry. These layers protect the glass giant from accidental impact and span over the voids of the structure. Furthermore, these layers ensure a flat walking surface.

What are the structural and manufacturing constraints for cast glass, and where do they differ for 2D and 3D optimization?

Cast glass has two critical structural constraints, both stemming from the amorphous molecular structure of glass This structure limits the plastic deformation, as a result, glass is brittle and exhibits a large difference in tensile and compression strength. To avoid breakage caused by of plastic deformation, the maximum deflection of the component is limited.

To ensure that the geometry can be fabricated, constraints for the maximum and minimum cross-sectional sizes must be incorporated into the algorithm. The maximum cross-sectional size constraint serves to keep the total annealing time of the structure under an acceptable threshold. Additionally, the relationship between the maximum and minimum cross-sectional sizes ensures a homogeneous mass distribution.

Most of the structural and manufacturing constraints are consistent for both two-dimensional and three-dimensional optimization. The main difference lies in the mathematical adaptations required to formulate these constraints effectively for three-dimensional calculations. The only additional manufacturing constraint relates to the avoidance of entirely internal voids.

Which algorithm methodology is most suitable for the Topology Optimization process?

In this thesis, the SIMP method has been selected as the topology optimization methodology. A previous thesis and other sources prove that this is a suitable methodology that provides fast and robust solutions. Furthermore, the formulation is clear and easier to comprehend than some of the other topology optimization methodologies like the level set method. The level set method can provide clearer boundaries, however, the formulation of the optimization problem is more complex and would require knowledge of lower-level coding languages.

Which finite element type is most suitable for the Topology Optimization Process?

After evaluation of the different finite elements, it was concluded that 8-node hexahedron elements best fit the needs of this project. While one-dimensional elements can be used to generate a three-dimensional structure. This would require extra post-processing to design the connecting nodes. Furthermore, the results of the existing two-dimensional optimization follow the typology of a spatial truss. One-dimensional elements would, because of the nature of the 1D element, also result in a spatial truss typology. Using 3D elements gives more insight into whether this material distribution is still preferred when optimizing in three dimensions.

How can the computational time be limited to ensure a feasible calculation?

The computational time of the algorithm depends on two main factors: the configuration of the optimization problem and the code's performance. The configuration of the optimization problem results in a total number of elements. This number is the result of the domain size and the finite element size. Larger domains with a greater number of elements lead to a significant increase in computational time. In cases where symmetry exists and the mirroring principle can be applied, it's possible to calculate just a section of the total design, thus improving efficiency.

To improve the performance of the algorithm the following steps were found crucial: Initializing matrices that are non-changing and continuously used during the optimization loop before entering the optimization loop; Vectorizing the code to avoid the usage of for loops; Using the preconditioned conjugate gradients method for solving the equilibrium equation in case the total amount of elements exceeds a certain threshold; Adding gradient functions to both the constraint and objective function to ensure faster convergence of the optimization. Including these methods brought down the computational time by a significant amount.

How do different objective functions influence the outcome of the topology optimization?

Several differences are observed between the geometries and structural performance of the domains optimized with different objectives. The volume optimization exhibits a larger degree of symmetry. Additionally, it creates a geometry that is more akin to that of an I beam. While the compliance optimization results in a box section. Moreover, the compliance optimization adheres to the volume constraint, whereas the volume optimization is able to reduce the mass of the structure further.

How do the results of the 3D optimization compare to the outcome of the 2D optimization in terms of structural performance, material utilization and geometry?

A comparative analysis of the 2D and 3D geometries reveals several notable distinctions. Firstly in terms of geometry. While the optimizations showcase similarities in material distribution, different structural typologies emerge when the algorithm is able to utilize the width of the structure. The two-dimensional geometry resembles a spacial truss. The three-dimensional optimization shows a clear preference for a web and flange geometry. In terms of performance, the three-dimensional optimization is able to find a more efficient material distribution than the two-dimensional optimization. The final compliance of the structure is lower, while the maximum deflection and peak stresses are well within the constraints. Additionally, this results in the case of the volume optimization in a lighter structure with 5% more material reduction.

What are the main benefits and limitations of the developed 3D algorithm compared to existing 2D algorithms for cast glass components?

The three-dimensional optimization has multiple benefits compared to the two-dimensional algorithm. Firstly, the geometries reached have a higher structural stiffness. Secondly, a larger material reduction can be reached. Lastly, but maybe most importantly, the three-dimensional optimization offers more spatial information. As the algorithm is able to utilize the width of the structure, it is not limited to generating a truss structure, but other structural typologies are explored.

The main limitation of the developed 3D algorithm remains the computational time and computational power needed. The increased complexity can cause memory issues when increasing the domain size. As a result, the calculation of larger domain sizes relies on access to a computer with a high RAM. Furthermore, the addition of the third dimension adds complexity to the code, making it harder to quickly and accurately make changes in the boundary conditions if one is not familiar with the code.

9.2 Research Limitations

This section outlines the various limitations and constraints of this thesis. Firstly, the scope of this thesis was limited to one case study design. All the testing considered a symmetric domain with an evenly distributed load and fully fixed sides. Although additional experiments were conducted with alternative boundary condition configurations, within the given time frame, these did not yield significant insights.

For larger calculations, an external supercomputer was employed. This supercomputer operates under a fair share principle, but it prioritizes highly parallel jobs. The optimization algorithm developed for this thesis works sequentially, leading to lengthy queuing times during busy periods and limiting the number of tests that could be executed. Additionally, the supercomputer imposes a 24-hour time constraint, which led to the termination of optimizations before convergence was achieved.

Moreover, this thesis was restricted to computer simulations due to time constraints. Prototyping the complex geometry could offer valuable insights into the manufacturing process, aiding in the specification of manufacturing constraints. Furthermore, prototypes could be used for structural testing to compare computer simulations with physical tests. Lastly, additional time should be invested in the further development of the post-processing process. Given the time constraints of this thesis, only one set of results was post-processed using a labour-intensive, manual approach.

9.3 Discussion and Future Recommendations

This thesis was able to take significant steps in the development of a customizable optimization tool for the design of large cast glass structures. A multitude of challenges had to be overcome and solved in order to reach the point of a three-dimensional optimization. However, many challenges still remain and the research can be extended in many directions.

9.3.1 Reevaluation of the Algorithm and Optimization Method

The developed algorithm uses 8-noded cubic elements and follows the SIMP methodology for the optimization. While considerable steps were taken to speed up the calculation time, the algorithm still requires a high RAM and significant time for larger problems. Furthermore, the combination of discretization and optimization methodology results in a geometry that has very hard edges. The post-processing steps do keep the important features of the geometry, however, the shape is significantly changed which adds a step of the final analysis of the geometry.

To improve the algorithm, several directions can be taken. While a fine enough mesh size can result in a better boundary, this greatly increases the computational time. Further research could be conducted in other finite element types. Changing to either higher order elements or trianglular elements, could result in a geometry with fewer hard edges, which consequently would bring down the need for post-processing. Additionally, different optimization methods can be researched. The level-set method is a method that directly results in a smooth boundary, so no additional post-processing is necessary. Parallel computing, artificial intelligence or evolutionary algorithms (not to be confused with the ESO and BESO method) can be approaches for calculating a finer mesh size while keeping computational time feasible.

9.3.2 Design Constraints

The design constraints integrated into the algorithm ensure a feasible design. While most constraints were successfully integrated into the optimization, the final geometry does not confirm to all constraints. Most prominent is that the minimum cross-section is not achieved at every location and that islanding does occur. Furthermore, the three-dimensional optimization requires a constraint to avoid complete internal voids. While the obtained geometries do not show any of these voids, this does not guarantee that these internal voids will not appear in other case studies.

To improve the algorithm, additional research is needed towards methods that ensure a minimum cross-section size. Literature points out the possibility of changing the minimum cross-section size constraint to a minimum void constraint. Furthermore, the algorithm can be improved by integrating the compressive and tensile stress constraint into one constraint using the Drucker-Prager stresses.

9.3.3 Boundary Conditions, Loading and Optimization Setup

With the boundary conditions chosen for the case study the component does not utilize the compressive strength of glass. While the compressive strength of glass is a factor 100 higher than the tensile strength, the compressive and tensile stresses are comparable. Testing different boundary conditions, showed that this highly influences the geometry and utilization of the compressive strength. However, no convergence was reached for larger components. To improve the design, further research should be done to determine other potential configurations for the supports. Additionally, more accurate modelling of the support conditions has the potential to result in further material reduction and a more accurate prediction of the structural behaviour of the component. Furthermore, both the compliance and volume constraint use a factor that is estimated by the user. The results achieved in this thesis all fall well within the limits of the structural constraints. Indicating that there is a chance that further material optimization is possible by fine-tuning the volume and compliance constraints.

Aside from additional testing, the algorithm can be further developed to make a wider range of domains possible. While the integration of grasshopper in the algorithm makes it easier to define complex boundary and loading conditions, the current algorithm is only able to optimize rectangular domains. This severely limits the possible applications. One way to calculate more complex domains would be the introduction of passive elements. These elements are set to always either be void of solid. This way sections of a rectangular domain could be subtracted to achieve other domain shapes. A downside of this approach is that the elements are part of the optimization and will increase the calculation time.

9.3.4 Post-Processing and Production Method

Due to the discretization in cubic elements, the geometry obtained from the optimization needs to undergo a labour-intensive post-processing phase. As mentioned previously, changing to either higher-order elements or elements of a different shape could result in a geometry that needs less post-processing. Furthermore, the grasshopper plugin Weaverbird seems to be a promising tool to directly post-process the geometry in Rhino. This would severely decrease the post-processing time, as no transferring between software would be needed.

Aside from improving the post-processing phase in Grasshopper, the complex geometry asks for additional research into the production method. A complex mould, consisting of multiple parts will be needed for larger cast glass structures. Extending the post-processing to include the design of the mould could give valuable input into the manufacturability of the component.

Additionally, this thesis only considered casting as the fabrication method for the component. This is, however, not the only method to produce complex shapes in glass. Methods have been developed for the additive manufacturing of glass. This production method would pose a different set of constraints on the geometry and further research into this could expand the possibilities of manufacturing structural glass components of complex shapes.

10
Reflection

1. How are research and design related?

This thesis is part of a larger ongoing research project at the TU Delft regarding structural glass as a novel construction material. Here, the focus is directed towards casting of glass to create completely free-form geometries. Thus far, only smaller components have been created, but a multitude of previous student work explored how the current limitations in the fabrication process could be overcome by using topology optimization to design larger cast glass components. This thesis continues this research by integrating the limitations and remarks concluded from the previous work to further explore the potential of designing large cast glass components using topology optimization.

While commercial topology optimization software exists, the application of those for the design of optimizing glass structures is hampered by their use of von Mises stresses. Glass is a brittle material, with considerably higher compressive strength than tensile strength. The usage of von Mises stresses makes it impossible to utilise both the compressive and tensile strength. A two-dimensional optimization algorithm has been created that integrates the two strength constraints as well as design-specific constraints related to cast glass as annealing and manufacturing criteria. This thesis aims to continue the development of this algorithm and to create a tool that can help generate three-dimensional mass-optimized designs.

2. How is the graduation topic positioned in the studio?

Within the larger building technology studio, this thesis is related to 'Structural Design' and 'Artificial Intelligence'. The Structural Design Studio focuses on the innovation of novel structures that are able to respond to the growing demand for structures in a sustainable way. This thesis researches how computational tools like topology optimization can be utilized in creating mass-optimized structures. Reducing the amount of glass needed for a structure consequently limits the energy needed to produce the component.

3. How did the research approach work out & did it lead to the results you aimed for?

The aim of this thesis was to develop a three-dimensional topology optimization algorithm. As this optimization is quite computationally expensive the chosen approach was to gradually scale-up the experimenting size. During the first steps of this process, the results can be compared with results found in the literature as well as with the outcomes from previous theses.

The step-wise approach made assessing the performance of the algorithm possible. The scaling up of the algorithm proved to have its difficulties. However, adopting new approaches resulted in a design tool that can efficiently optimize a three-dimensional design.

4. To what extent are the results applicable in practice?

The developed algorithm is able to take a large range of constraints into consideration in order to reach a result that is feasible within the structural and design constraints. While the obtained geometry is complex, a recent thesis shows the potential of using coatings on 3D-printed sand moulds to create transparent glass structures with complex shapes.

Some geometrical errors still occur in the structure and a post-processing procedure is needed to obtain a final smooth geometry. Further research can be done to avoid any geometrical issues and non-connected elements in the final structure. Furthermore, physical testing is needed to evaluate the structural performance and validate the results from the computer simulations.

5. To what extent has the project innovation been achieved?

The innovation in this thesis lies in the creation of a custom algorithm to make a three-dimensional calculation possible. A lot of progress was made to limit the computational time of the algorithm to make optimizations over a larger domain possible. Including gradient functions in the optimization was revealed to be the key factor in making a three-dimensional optimization feasible within a reasonable time frame. By increasing the computational effectiveness the algorithm can be used as a base to quickly get an insight into the most optimal material layout for a design. However, still a post processing step is needed, as the optimization result can have geometrical errors.

6. What is the impact of the project on sustainability & sustainable development

The possibility of mass-optimizing glass designs contributes significantly to more sustainable structures. It reduces material use and lessens the energy used for the fabrication of components. Furthermore, the parametric nature of the algorithm makes it possible to use the tool for a multitude of glass types. Making it possible to apply the algorithm in design cases that work with recycling waste glass. Consequently, the algorithm can also be used for materials with a similar behaviour to glass, like concrete.

7. What is the socio-cultural and ethical impact of the project and what is the relation between this project and the wider social context

The growth of the populating and the ageing of the existing infrastructure is putting a high demand on the building industry. Currently, the building industry is already one of the largest emitters of CO2. Thus reducing the use of resources and including recyclability in designs are becoming increasingly important. This project adds to the research into decreasing material usage by optimizing the structural performance of designs. Furthermore, while the structural use of glass is still very novel, high compressive strength give the possibility to compete with common construction materials like concrete and steel. Here it should also be noted that an added benefit of glass is that it is infinitely recyclable.

8. How does the project affect architecture and the built environment

In short, the algorithm offers architects and designers a way to explore the possibilities of creating large cast glass structures and potentially introducing a new architectural language. This is because the algorithm makes it possible to use cast glass on a scale that was previously not feasible to fabricate. The three-dimensionality of the algorithm showcases the spatial qualities that topologically optimized designs can have. Using the algorithm as a tool, designers can play with boundary settings to create intricate designs that utilize structural performance to minimize material usage. The algorithm ensures structural performance which in turn encourages the usage of cast glass for self-standing structural components. Furthermore, this enables the change in mindset toward glass as a structural material.

Acknowledgements

This thesis marks the end of a 6 years journey at the TU Delft. During these years I have continuously been surrounded by amazing people that have motivated me to work hard and find the best version of myself and would like to sincerely thank everybody that has been part of this journey.

First and foremost, I would like to express my sincere gratitude to my mentors **Dr. Charalampos Andriotis** and **Dr. Faidra Oikonomopoulou**. From the moment we first discussed the topic, you ignited my enthusiasm for the research and the prospect of working with you. Although I initially had reservations about the project's complexity, your motivation and trust gave me the confidence to achieve this result. Thank you for your patience, guidance and feedback.

To **Anna Maria Koniari**, without your work on topology optimization this thesis would not have been possible. Thank you for sharing your work and being available to answer questions and give advise on the topic.

Mom, thank you for forgiving me for all the dinners I missed because I was working on deadlines, and feeding me regardless. **Dad**, thank you for all the car rides to Delft and reassurance that everything will be okay whichever journey I choose. **Linde** and **Melle** you are the best siblings a person could wish for.

Lastly, I would like to express by gratitude to all the friends that supported me throughout the last 6 years. **Martin, Sandy** and **Valerie**, I am incredibly grateful for your support during the bachelor and the friendships that followed. **Alina**, **Lotte** and **Pranay** thank you for your friendship and unwavering support these crazy last two years. **Patrick** thank you for your coffee's and amazing render skills. **Niccolo**, thank your for your reassurance and kindness.

Finally, i would like to devote this work in the memory of **Amber**. We started this journey together and while yours was way to short, your enthusiasm and love shaped much of my time in Delft.

Bibliography

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*(1), 1–16. https://doi.org/10.1007/s00158-010-0594-7

Bendsøe, M. P., & Sigmund, O. (2004). *Topology Optimization. Theory, Methods, and Applications*. Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-05086-6

Bendsøe, M. P., & Kikuchi, N. (1988). Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, *71*(2), 197–224. https://doi.org/https://doi.org/10.1016/0045-7825(88)90086-2

Bhatia, I. S. (2019). *Shaping Transparent Sand in Sand. Fabricating topologically optimized cast glass column using sand moulds* (Master's thesis). Technical University Delft. Delft.

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Bristogianni, T., Oikonomopoulou, F., Yu, R., Veer, F. A., & Nijsse, R. (2020). Investigating the flexural strength of recycled cast glass. *Glass Structures & Engineering*, *5*(3), 445–487. https://doi.org/10.1007/s40940-020-00138-2

Bruggi, M. (2008). On an alternative approach to stress constraints relaxation in topology optimization. *Structural and Multidisciplinary Optimization*, *36*(2), 125–141. https://doi.org/10.1007/s00158-007-0203-6

Bruggi, M., & Duysinx, P. (2012). Topology optimization for minimum weight with compliance and stress constraints. *Structural and Multidisciplinary Optimization*, *46*. https://doi.org/10.1007/s00158-012-0759-7

Bruns, T., & Tortorelli, D. (2001). Topology optimization of non-linear structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, *190*, 3443–3459. https://doi.org/10.1016/S0045-7825(00)00278-4

Chandrupatla, T., & Belegundu, A. (2002). *Introduction to Finite Elements in Engineering* (3rd ed.). Prentice Hall Inc. https://doi.org/10.1017/9781108882293

Cheng, G. D., & Guo, X. (1997). E-relaxed approach in structural topology optimization. *Structural optimization*, *13*(4), 258–266. https://doi.org/10.1007/BF01197454

Collet, M., Bruggi, M., & Duysinx, P. (2017). Topology optimization for minimum weight with compliance and simplified nominal stress constraints for fatigue resistance. *Structural and Multidisciplinary Optimization*, *55*(3), 839–855. https://doi.org/10.1007/s00158-016-1510-6

Damen, W. (2019). *Topologically Optimised Cast Glass Grid Shell Nodes. Exploring Topology Optimization as a design tool for Structural Cast Glass elements with reduced annealing time.* (Master's thesis). Technical University Delft. Delft.

Damen, W., Oikonomopoulou, F., Bristogianni, T., & Turrin, M. (2022). Topologically optimized cast glass: A new design approach for loadbearing monolithic glass components of reduced annealing time. *Glass Structures & Engineering*, *7*(2), 267–291. https://doi.org/10.1007/s40940-022-00181-1

Fernández, E., Yang, K.-k., Koppen, S., Alarcón, P., Bauduin, S., & Duysinx, P. (2020). Imposing minimum and maximum member size, minimum cavity size, and minimum separation distance between solid members in topology optimization. *Computer Methods in Applied Mechanics and Engineering*, *368*, 113157. https://doi.org/https://doi.org/10.1016/j.cma.2020.113157

Guest, J., Prevost, J., & Belytschko, T. (2004). Achieving minimum length scale in topology optimization using nodal design variable and projection functions. *International Journal for Numerical Methods in Engineering*, *61*, 238–254. https://doi.org/10.1002/nme.1064

Guest, J. K. (2009). Imposing maximum length scale in topology optimization. *Structural and Multidisciplinary Optimization*, *37*(5), 463–473. https://doi.org/10.1007/s00158-008-0250-7

Haldimann, M. (2006). Fracture strength of structural glass elements: Analytical and numerical modelling, testing and design. https://doi.org/10.5075/epfl-thesis-3671

Huang, X., & Xie, Y. (2007). Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. *Finite Elements in Analysis and Design*, *43*(14), 1039–1049. https://doi.org/10.1016/j.finel.2007.06.006

Ioannidis, M. (2023). *Bringing Glass Ginats to Life. Fabrication of mass-optimized structural glass components of complex form* (Doctoral dissertation). Technical University Delft. Delft.

Jewett, J. L., & Carstensen, J. V. (2019). Topology-optimized design, construction and experimental evaluation of concrete beams. *Automation in Construction*, *102*, 59–67. https://doi.org/https://doi.org/10.1016/j.autcon.2019.02.001

Jipa, A., Bernhard, M., Meibodi, M., & Dillenburger, B. (2016). *3D-Printed Stay-in-Place Formwork for Topologically Optimized Concrete Slabs*. https://doi.org/10.3929/ethz-b-000237082

Koniari, A. M. (2022). *Just Glass. Development of a Topology Optimization ALgorithm for a Mass-Optimized Cast Glass Component* (Master's thesis). Technical University Delft. Delft.

Kumar, P. (2017). *Synthesis of Large Deformable Contact-Aided Compliant Mechanisms using Hexagonal cells and Negative Circular Masks* (Doctoral dissertation).

Le, C., Norato, J., Bruns, T., Ha, C., & Tortorelli, D. (2010). Stress-based topology optimization for continua. *Structural and Multidisciplinary Optimization*, *41*(4), 605–620. https://doi.org/10.1007/s00158-009-0440-y

Li, Q., Chen, W., Liu, S., & Fan, H. (2018). Topology optimization design of cast parts based on virtual temperature method. *Computer-Aided Design*, *94*, 28–40. https://doi.org/https://doi.org/10.1016/j.cad.2017.08.002

Liolios, P. (2020). Deviatoric stress and invariants. https://pantelisliolios.com/deviatoric-stress-and-invariants/

Liu, G. R., & Quek, S. S. (2014). Chapter 9 - FEM for 3D Solid Elements. In G. R. Liu & S. S. Quek (Eds.), *The Finite Element Method (Second Edition)* (Second Edition, pp. 249–287). Butterworth-Heinemann. https://doi.org/https://doi.org/10.1016/B978-0-08-098356-1.00009-6

Liu, K., & Tovar, A. (2014). An efficient 3D topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, *50*(6), 1175–1196. https://doi.org/10.1007/s00158-014-1107-x

Liu, S., Li, Q., Chen, W., Tong, L., & Cheng, G. (2015). An identification method for enclosed voids restriction in manufacturability design for additive manufacturing structures. *Frontiers of Mechanical Engineering*, *10*, 126–137. https://doi.org/10.1007/s11465-015-0340-3

Meibodi, M. A., Bernhard, M., Jipa, A., & Dillenburger, B. (2017). THE SMART TAKES FROM THE STRONG: 3D PRINTING STAY-IN-PLACE FORMWORK FOR CONCRETE SLAB CONSTRUCTION. In *Fabricate 2017*. UCL Press. https://doi.org/10.2307/j.ctt1n7qkg7.33

Öchhsner, A., & Merkel, M. (2018). *One-Dimensional Finite Elements. An introduction to the FE method* (2nd ed.). Springer International Publishing AG. https://doi.org/10.1007/978-3-319-75145-0

Oikonomopoulou, F. (2019). *Unveiling the third dimension of glass. Solid cast glass components and assemblies for structural applications.* (PHD dissertation). Technical University Delft. Delft.

Oikonomopoulou, F., Bhatia, I., van der Weijst, F., Damen, W., & Bristogianni, T. (2020). *Rethinking the Cast Glass Mould. An Exploration on Novel Techniques for generating Complex and Customized Geometries*.

Osher, S., & Fedkiw, R. P. (2001). Level Set Methods: An Overview and Some Recent Results. *Journal of Computational Physics*, *169*(2), 463–502. https://doi.org/https://doi.org/10.1006/jcph.2000.6636

Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, *79*(1), 12–49. https://doi.org/https://doi.org/10.1016/0021-9991(88)90002-2

Petraroia, D. (2023). Stiffness Matrix for 8-Node Hexahedron (https://www.mathworks.com/matlabcentral/fileexchange/673 stiffness-matrix-for-8-node-hexahedron), Retrieved September 27, 2023. (https://www.mathworks.com/matlabcentral/fileexchange/67320-stiffness-matrix-for-8-node-hexahedron

Querin, O. M., Victoria, M., Alonso, C., Ansola, R., & Martí, P. (2017). *Topology Design Methods for Structural Optimization*. Academic Press. https://doi.org/10.1016/B978-0-08-100916-1.00001-5.

Shand, E. B., & Armistead, W. H. (1958). *Glass engineering handbook*. McGraw-Hill.

Shimels, G. H., Engida, W. D., & Mohd, H. F. (2017). A comparative study on stress and compliance based structural topology optimization [Publisher: IOP Publishing]. *IOP Conference Series:*

Materials Science and Engineering, *241*(1), 012003. https://doi.org/10.1088/1757-899X/241/1/012003

Sigmund, O. (2001). A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, *21*(2), 120–127. https://doi.org/10.1007/s001580050176

Sigmund, O. (1994). *Design of Material Structures Using Topology Optimization* (PhD Thesis).

Sigmund, O. (2007). Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, *33*(4), 401–424. https://doi.org/10.1007/s00158-006-0087-x

Sigmund, O., & Maute, K. (2013). Topology optimization approaches. *Structural and Multidisciplinary Optimization*, *48*(6), 1031–1055. https://doi.org/10.1007/s00158-013-0978-6

Stefanaki, M. I. (2020). *Glass Giants. Mass-optimized massive cast galss slab* (Master's thesis). Technical University Delft. Delft.

Wai, C. M., Rivai, A., & Bapokutty, O. (2013). Modelling optimization involving different types of elements in finite element analysis. *IOP Conference Series: Materials Science and Engineering*, *50*(1), 012036. https://doi.org/10.1088/1757-899X/50/1/012036

Xie, Y. M., & Steven, G. P. (1993). A simple evolutionary procedure for structural optimization. *Computers & Structures*, *49*(5), 885–896. https://doi.org/https://doi.org/10.1016/0045-7949(93)90035-C

Xie, Y. M., & Steven, G. P. (1997). *Evolutionary Structural Optimization*. Springer London. https://doi.org/10.1007/978-1-4471-0985-3_2

Yago, D., Cante, J., Lloberas-Valls, O., & Oliver, J. (2022). Topology Optimization Methods for 3D Structural Problems: A Comparative Study. *Archives of Computational Methods in Engineering*, *29*(3), 1525–1567. https://doi.org/10.1007/s11831-021-09626-2

Yang, R. J., & Chen, C. J. (1996). Stress-based topology optimization. *Structural optimization*, *12*(2), 98–105. https://doi.org/10.1007/BF01196941

Yin, L., & Ananthasuresh, G. (2003). Design of Distributed Compliant Mechanisms. *Mechanics Based Design of Structures and Machines - MECH BASED DES STRUCT MECH*, *31*, 151–179. https://doi.org/10.1081/SME-120020289

List of Figures

List of Tables

A
Appendix

A.1 Read Me

The three-dimensional topology optimization algorithm for the design of cast glass components developed for this thesis can be found at: https://github.com/ESchoenmaker/3D_TO_cast_glass.

'Functions used' folder contains functions used in the algorithm. 'TO algorithm' folder contains different variations of the topology optimization code. There are two versions that completely function in matlab, the boundary conditions here have to be defined in code. The two versions are a volume optimization and a compliance optimization. For both versions there is also a version that uses a grasshopper script to define the boundary conditions.

How to use the algorithm in combination with grasshopper:

1. Create a excel file and copy the path to the panel labeled 'Excel to write to' in the 'Boundary_to_excel.gh' file.
2. Alter the domain and boundary conditions to liking using the sliders connected to the python component, the worksheet name will be used to read the coordinates in matlab.
3. Enable toggle to write coordinates of nodes to the excel file (file needs to be closed)
4. Save the excel folder in the same folder as the MATLAB algorithm
5. Open the matlab script
6. Change the 'name_excel' variable to the name of the excel folder containing the boundary conditions in the 'F_data_gh' function. Additionally in this file changes can be made to the material properties as well as naming the run.
7. Under boundary conditions in the main script, use the funtion f_read_nodes_gh to read the coordinates out of grasshopper. This function will automatically change them to the right nodal numbers and degrees of freedom corresponding to the algorithm setup. The last input 'mat_dofs' refers to in which direction the degree of freedom should be calculated [-2 -1 0] for [y x z] multiple can be kept in case of a node that is fixed for multiple directions.
8. Define the load on the individual nodes. The differentiation between load_middle and load_sides ensures that the side nodes have less load applied to them than those carrying a larger surface.
9. Ensure that under 'choose constraint combination' the optimization uses all the constraints you want to optimize with
10. Run the code
11. When finished matlab should show the density map
12. If desired, the density map can be transferred to rhino using the 'Excel_to_Geometry.gh' script

Additional changes output

The output of the algorithm can be a new excel file, a density matrix and or figure of the geometry. Changes can be made in the F_output_case function to specify the preferred output. Overwriting the excel with the output of each iteration is advised. Not plotting the figure at each iteration can save a lot of time, and is advised for larger optimizations. Saving the density plot of each iteration is advised for long calculations.

There are three plotting function integrated in the algorithm. 'F_plot_results' will plot exactly the design domain as entered without mirroring. 'F_plot_results_mirror' offers the possibility to mirror the plot in the x and y direction. 'F_plot_stress_FEA_final' will plot two graphs displaying the tensile and compressive stresses in the final design.

For all three plot options it is possible to plot either all elements, or the elements from a certain density threshold. This threshold can be set by the user, the standard value is set to plot any element with a density above 0.3.

Possible Error messages

Errors regarding issues using reshape or arrays not having the correct sizes for matrix multiplication can often times be solved by clearing the workspace.

A.2 Construction B matrix

For 8-node hexahedral elements the lagrange shape functions can be derived from the master element **add image**.

$$N_i = \frac{1}{8}(1+\xi_i\xi)(1+\eta_i\eta)(1+\zeta_i\zeta)$$

The shape functions are used to define the displacements at any point.

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \dots & N_8 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \dots & 0 & N_8 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \dots & 0 & 0 & N_8 \end{bmatrix} * \begin{bmatrix} q_1 & q_2 \dots q_8 \end{bmatrix}^T$$

$$\begin{aligned} x &= N_1x_1 + N_2x_2 + \cdots + N_8x_8 \\ y &= N_1y_1 + N_2y_2 + \cdots + N_8y_8 \\ z &= N_1z_1 + N_2z_2 + \cdots + N_8z_8 \end{aligned}$$

We need to express the derivatives of a function in x, y -z coordinates in terms of its derivatives in Lagrangian coordinates. However from equation f can be seen as an implicit function of ξ, η and ζ. $f = f[x(\xi,\eta,\zeta)\ y(\xi,\eta,\zeta)\ z(\xi,\eta,\zeta)$

Now the chain rule can be used to get the partial derivatives expressed in the Lagrangian coordinates.

$$\begin{Bmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \\ \frac{\partial f}{\partial \zeta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{Bmatrix}$$

However we want ε

$$\begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{Bmatrix} = J^{-1} \begin{Bmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \\ \frac{\partial f}{\partial \zeta} \end{Bmatrix}$$

Which can be seen as expresions of the displacements u v and w

$$\begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial v}{\partial \xi} & \frac{\partial w}{\partial \xi} \\ \frac{\partial u}{\partial \eta} & \frac{\partial v}{\partial \eta} & \frac{\partial w}{\partial \eta} \\ \frac{\partial u}{\partial \zeta} & \frac{\partial v}{\partial \zeta} & \frac{\partial w}{\partial \zeta} \end{bmatrix}$$

Using the shape functions we get:

$$\begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \dots & \frac{\partial N_8}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \dots & \frac{\partial N_8}{\partial \eta} \\ \frac{\partial N_1}{\partial \zeta} & \frac{\partial N_2}{\partial \zeta} & \dots & \frac{\partial N_8}{\partial \zeta} \end{bmatrix} \begin{bmatrix} q_1 & q_2 & q_3 \\ \vdots & \vdots & \vdots \\ q_{22} & q_{23} & q_{24} \end{bmatrix}$$

Finally this can be reshaped to fit ε

$$\begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \end{Bmatrix} = [B] \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ \dots \\ q_{24} \end{Bmatrix}$$

Now the stress in the element can be expressed by $\sigma = DBq$ And the stiffness matrix can be expressed as

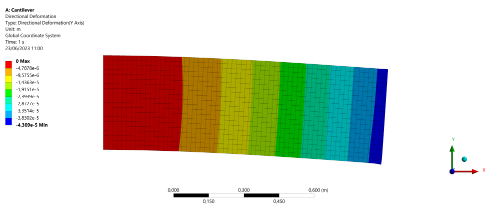
$$k^e = \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} B^T DB |\mathsf{det} J| d\varepsilon d\eta d\zeta$$

Which is integrated using the gauss points. For the algorithm a function written by .. cite is used to compute the stiffness matrix.

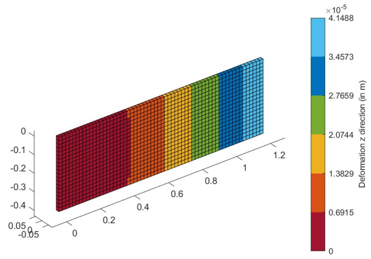
A.3 Verification of the Structural Model

The following sections contain the results from the structural verification.

A.3.1 Cantilever

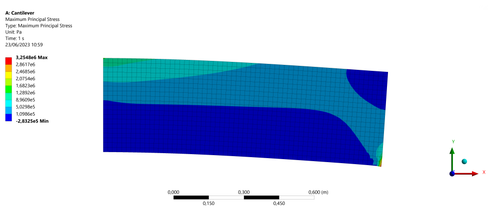


(a) Result ANSYS

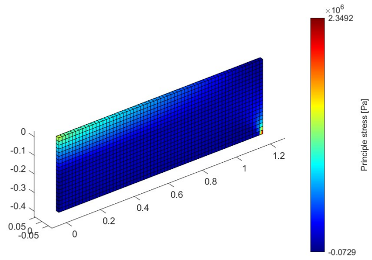


(b) Result Matlab

Figure A.1: Cantilever Deformation

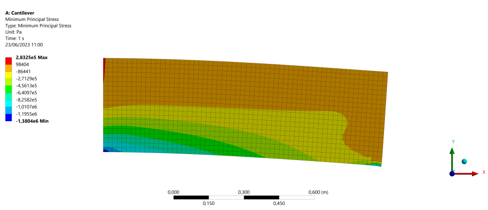


(a) Result ANSYS

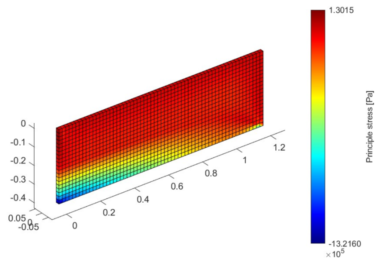


(b) Result Matlab

Figure A.2: Cantilever Maximum principal stress



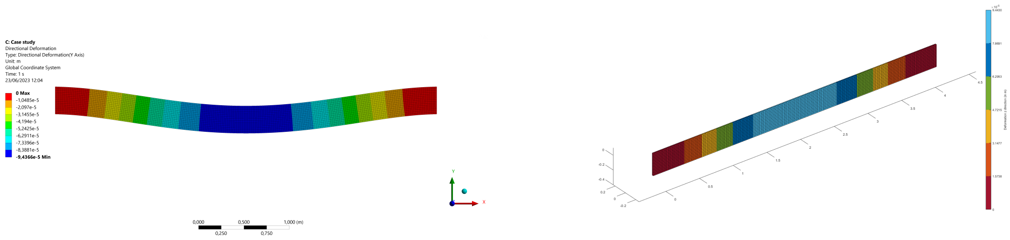
(a) Result ANSYS



(b) Result Matlab

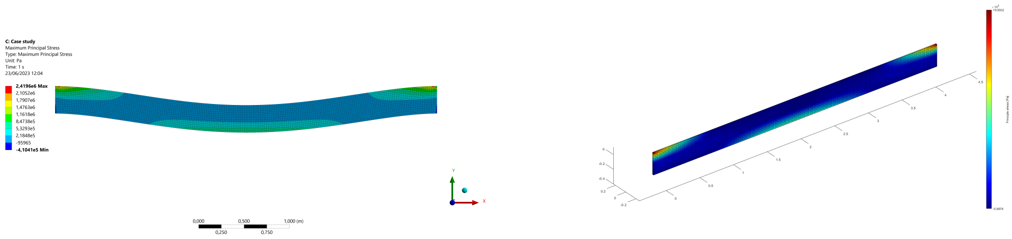
Figure A.3: Cantilever Minimum principal stress

A.3.2 Case study



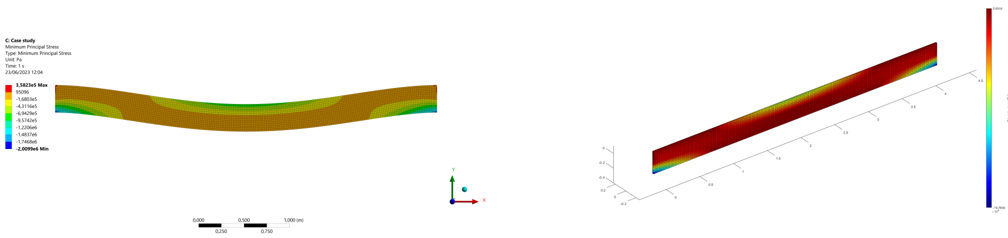
(a) Result ANSYS **(b)** Result Matlab

Figure A.4: Case Study Deformation



(a) Result ANSYS **(b)** Result Matlab

Figure A.5: Case Study Maximum principal stress



(a) Result ANSYS **(b)** Result Matlab

Figure A.6: Case Study Minimum principal stress

	Maximum displacement (mm)	Maximum principal stress	Minimum principal stress
ANSYS	-0.4309	0.003254	-0.001380
Matlab	-0.4148	0.002349	-0.001321

Table A.1: Results Cantilever (line load)

	Maximum displacement (mm)	Maximum principal stress	Minimum principal stress
ANSYS	-0.94366	0.002419	-0.002099
Matlab	-0.94430	0.001993	-0.001976

Table A.2: Results Case study (distributed load)

A.4 Additional Calculations Algorithm Performance

A.4.1 Benchmark Testing

The benchmark domain was used to test the individual constraints.

Volume Constraint — Compliance optimization

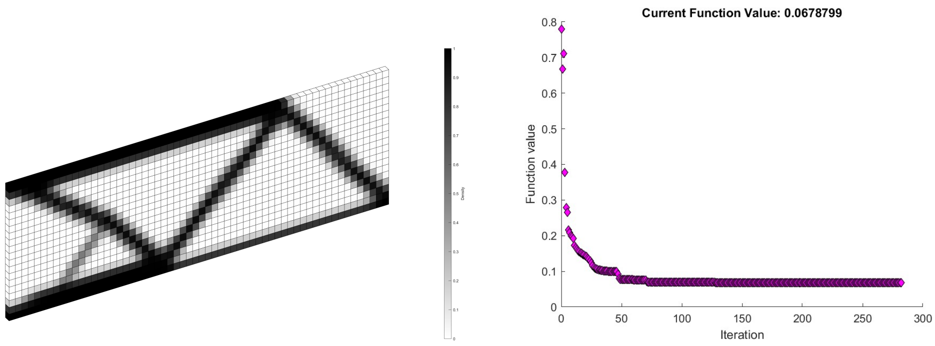


Figure A.7: Results **compliance** optimization with just volume constraint; $v_{frac} = 0.3$

Compliance Constraint — Volume optimization

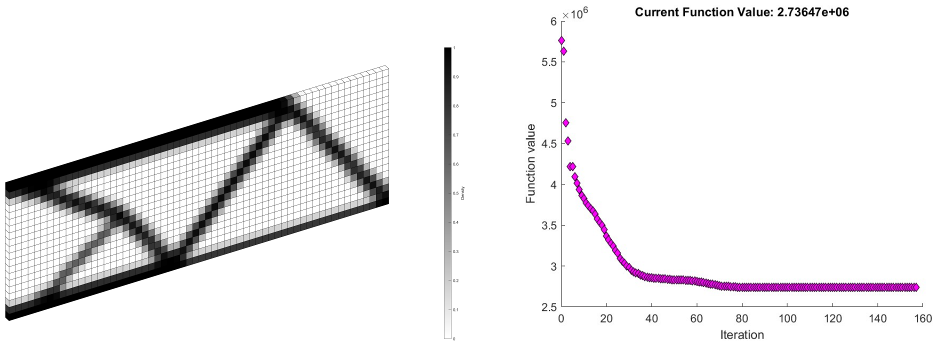


Figure A.8: Results **volume** optimization with just compliance constraint; $a_c = 3.5$

Principal Stress Constraint

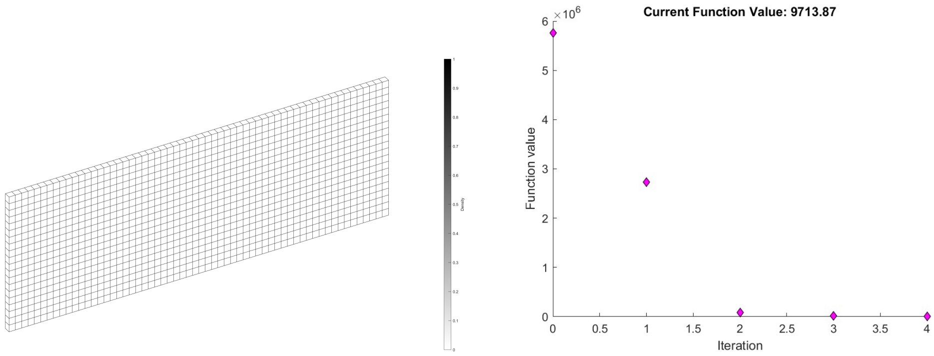


Figure A.9: Results **volume** optimization with just principal stress constraint;$q = 2.8$

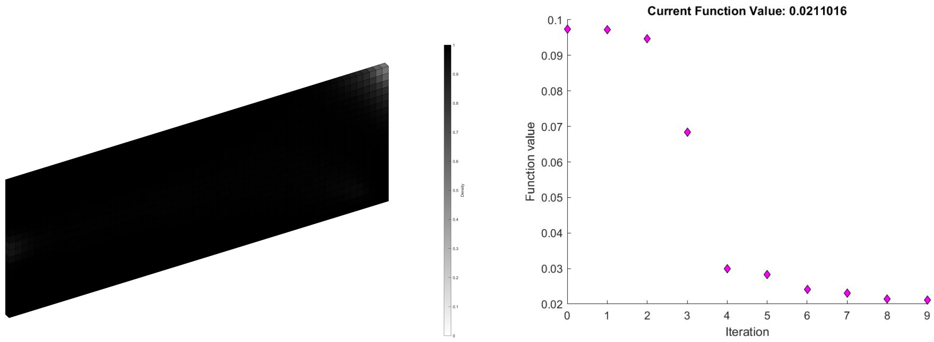


Figure A.10: Results **compliance** optimization with just principal stress constraint;$q = 2.8$

Deflection Constraint

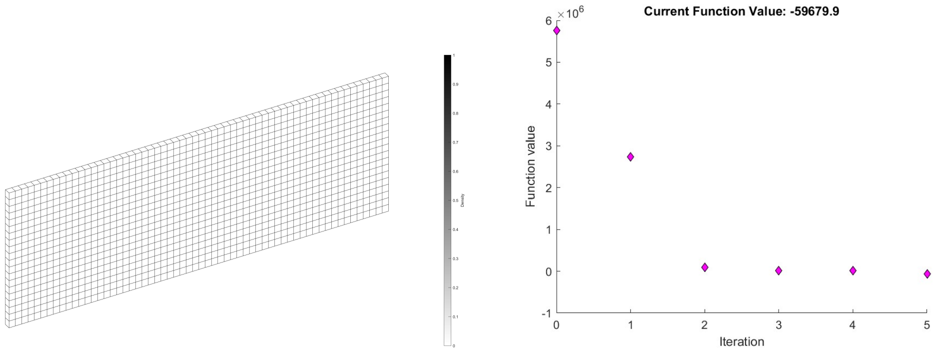


Figure A.11: Results **volume** optimization with just deflection constraint

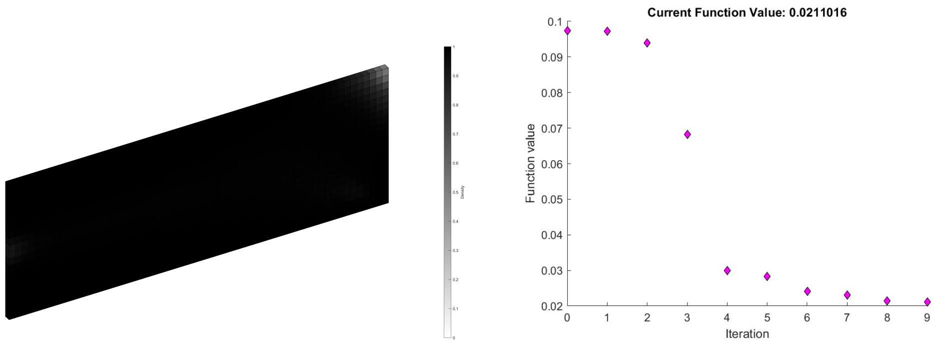


Figure A.12: Results **compliance** optimization with just deflection constraint

Annealing Constraint

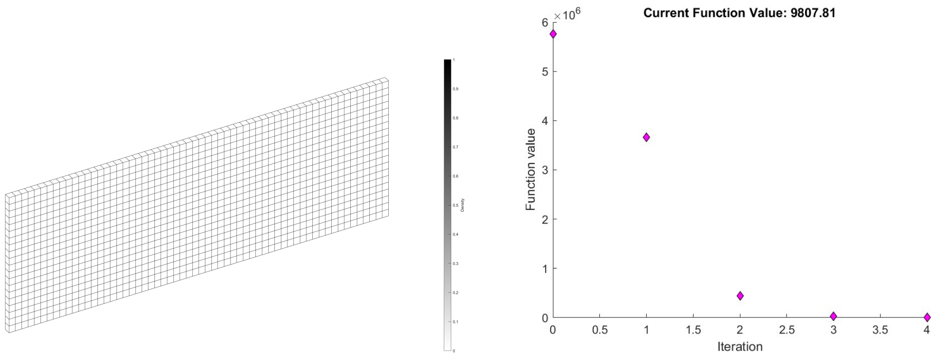


Figure A.13: Results **volume** optimization with just annealing constraint;$a_v = 0.9$

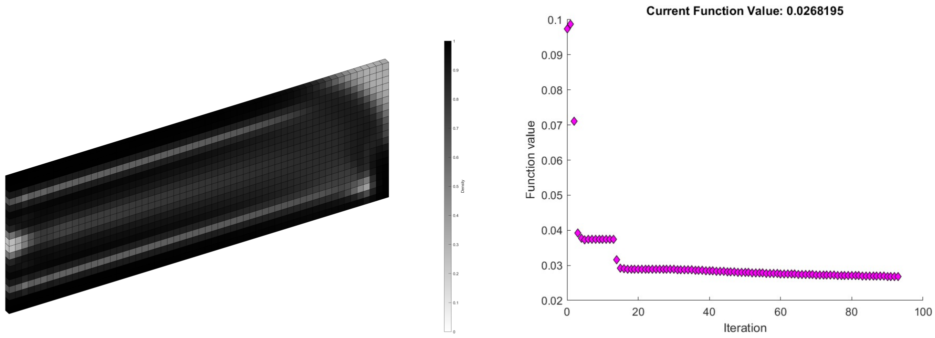
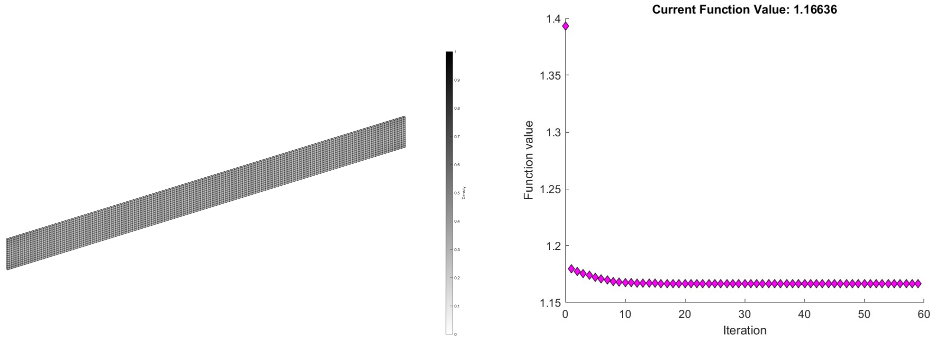


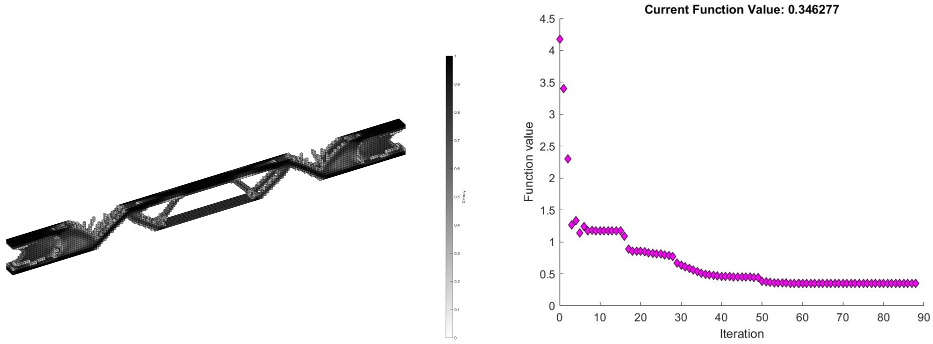
Figure A.14: Results **compliance** optimization with just annealing constraint;$a_v = 0.9$

A.4.2 Additional Case Study Calculations



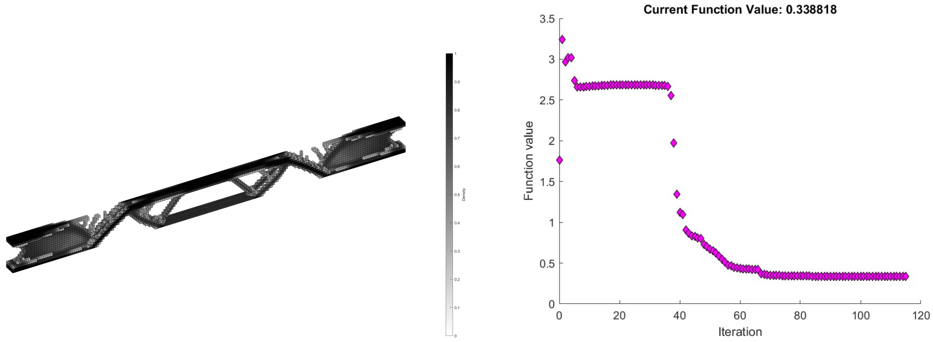
(a) Result Geometry **(b)** Plot of function value per iteration

Figure A.15: Results Compliance optimization with stress and volume constraint ($q = 2.8$ $v = 0.3$ $v_{initial} = 0.3$)



(a) Result Geometry **(b)** Plot of function value per iteration

Figure A.16: Results 3D Beam compliance optimization with All constraints (vi 0.3 a 2.8 av 0.9)



(a) Result Geometry **(b)** Plot of function value per iteration

Figure A.17: Results 3D Beam compliance optimization with All constraints (vi 0.4 a 2.5 av 0.9)

A.4.3 Test Setup 3D compliance optimization

The 2D test gave insight in how the outcome of the algorithm changes depending on the setup of the constraints. The results were taken as the starting point for the 3D optimization. The parameters found for the two dimensional tests did not directly result in good outcomes for the compliance optimization. The following table shows different setups that were tried to find a working optimization setup. Multiple tests showed that for the compliance optimization it is crucial to have a initial volume that is the same as the volume constraint. For the volume optimization the opposite is true, where the initial volume must be set higher than the approximated final volume percentage.

Setup	Constraints	Initial volume	Volume Constraint	Stress Relaxation	Result	F value
1	all	0.4	0.3	2.8	Bad	9.051
2	all	0.8	0.3	2.8	Worse	18.562
5	all	0.3	0.3	2.5	Good	2.054

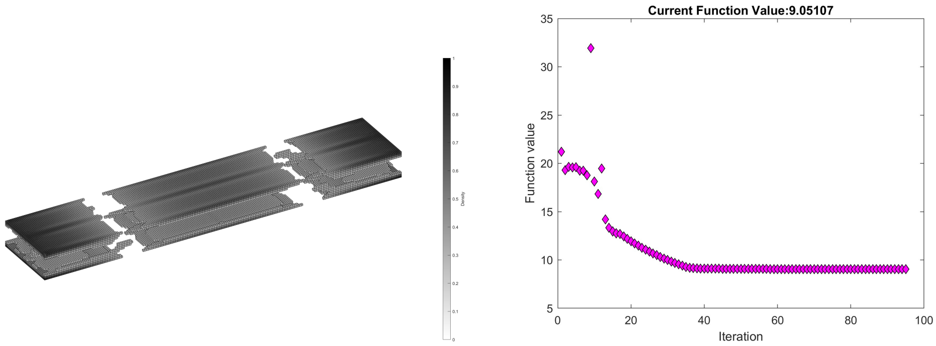


Figure A.18: Results 3D compliance optimization setup 1, all constraints, initial volume 0.4

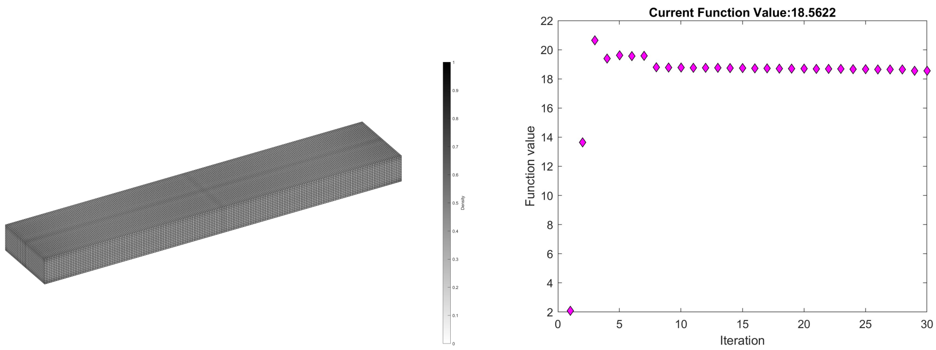


Figure A.19: Results 3D compliance optimization setup 2, all constraints, initial volume 0.4

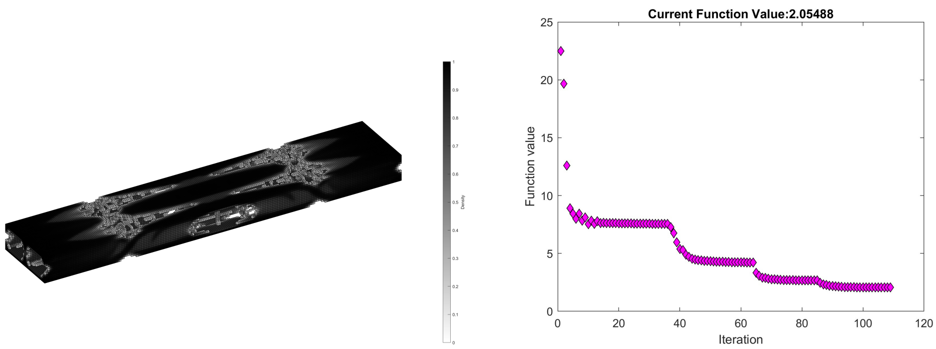


Figure A.20: Results 3D compliance optimization setup 5, all constraints, initial volume 0.3

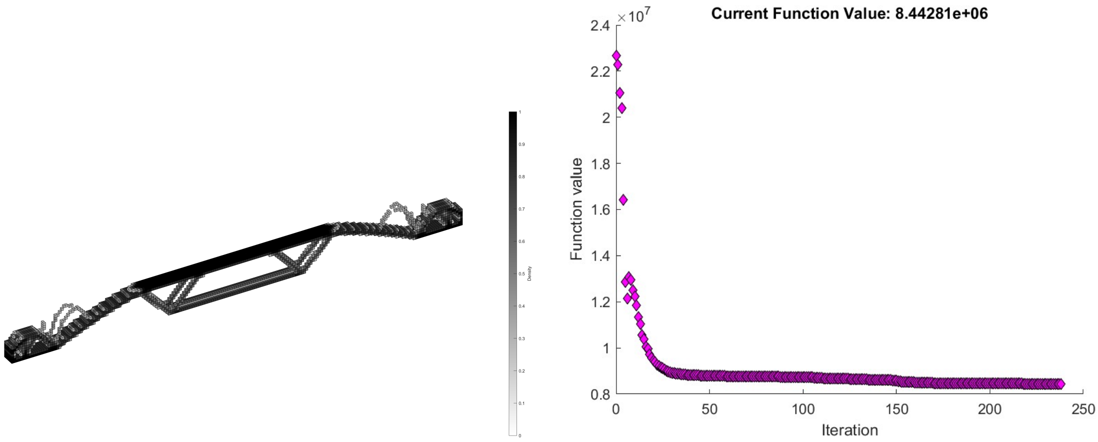
A.5 Alternative Design Options

The following sections will discuss some alternative design options that have been tested out, but not included in the main report.

A.5.1 Alteration Support Condition

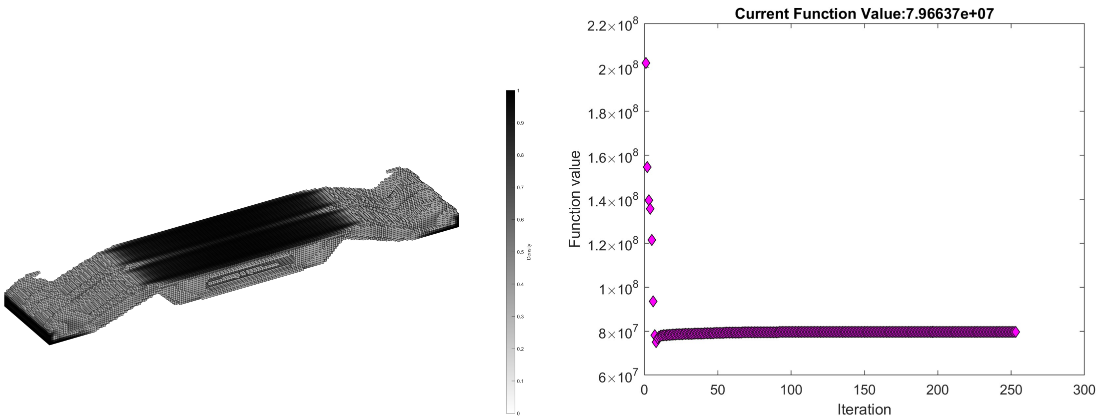
A test was done to see the result of alternative support condition. Instead of having fully fixed design options, the structure is simply supported at the side. It is assumed that the 5 first rows at the bottom of the structure are fully fixed, as well as that the bottom five rows of nodes at left face are fixed in the x direction.

For a design domain of a small beam this gives very good results. Achieving a volume reduction to 20%, a maximum compressive stress of -.0080 blabla and a maximum tensile stress of 0.0047. Indicating that with slight changes to the boundary conditions the compressive strength can be further utilized. However, explaining the design to a larger domain does not result in a converged design. Further exploration why this design does not converge could give more insight into the algorithm. However, because of time limitations this was seen as outside the scope of this thesis.



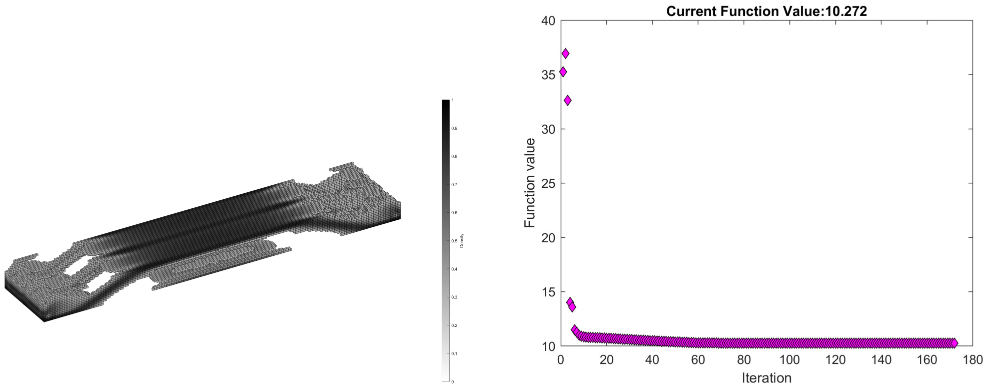
(a) Result Geometry **(b)** Plot of function value per iteration

Figure A.21: Results 3D Beam volume optimization with alternate support condition



(a) Result Geometry **(b)** Plot of function value per iteration

Figure A.22: Results case study domain with altered supports, volume optimization



(a) Result Geometry **(b)** Plot of function value per iteration

Figure A.23: Results case study domain with altered supports, compliance optimization

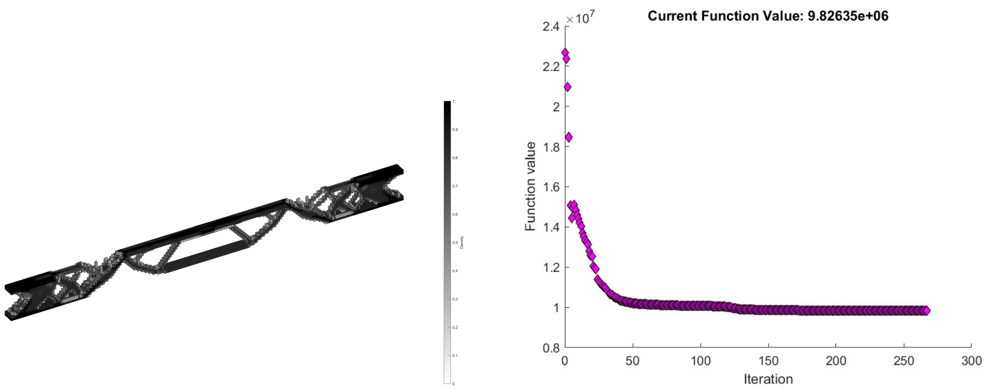
A.5.2 Alteration Glass Type

A test was done to see how changing the glass type would influence the design results. Instead of borocilicate glass, the optimization is run with the material properties of soda lime glass. While the structural and mechanical properties are similar, the thermal property is quite different. Soda lime has a higher thermal expansion coefficient, which influences the total annealing time needed. To ensure the same annealing time, the algorithm will impose a smaller cross section size. The following material properties are changed for this optimization:

	Symbol	Units	Soda lime	Borosilicate
Young's Modulus	E	GPa	70	70
Poisson's ratio	v	-	0.2	0.2
Density	p	kg/m3	2500	2500
Initial Cooling range	ΔT	C	553-485 (=68)	530-460(=70)
Thermal expansion coefficient	a_t	1/k	8.5 *10^{-6}	3.25*10^{-6}
Thermal Conductivity	K	W/(m*K)	1.06	1.15
Specific heat capacity	c_p	J(kg*K)	870	800

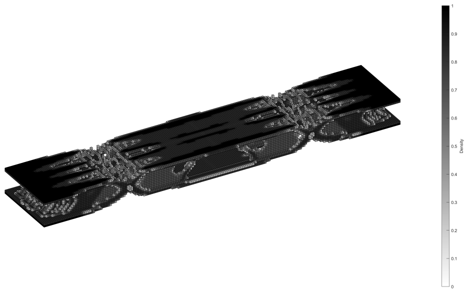
Table A.3: Imput values for the algorithm (borosilicate and soda lime glass) Altered from Koniari (2022)

The optimization was first tested on the beam design domain. The resulting geometry is very similar to that of the optimization with borosilicate glass. However, the result has a slight difference in stress value, with the maximum tensile stress being 0.0051 and the maximum compressive stress being -0.0060.

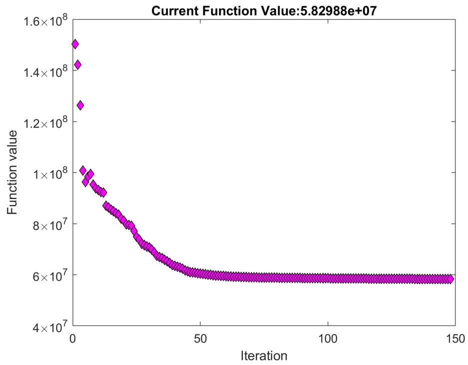


(a) Result Geometry **(b)** Plot of function value per iteration

Figure A.24: Results Volume optimization case study Soda Lime glass

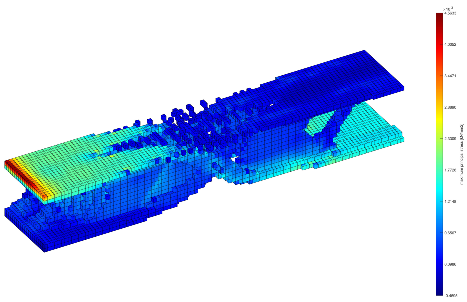


(a) Result Geometry

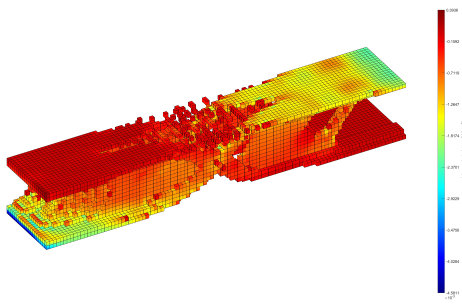


(b) Plot of function value per iteration

Figure A.25: Results Volume optimization case study Soda Lime glass



(a) Maximum principal stress



(b) Minimum principal stress

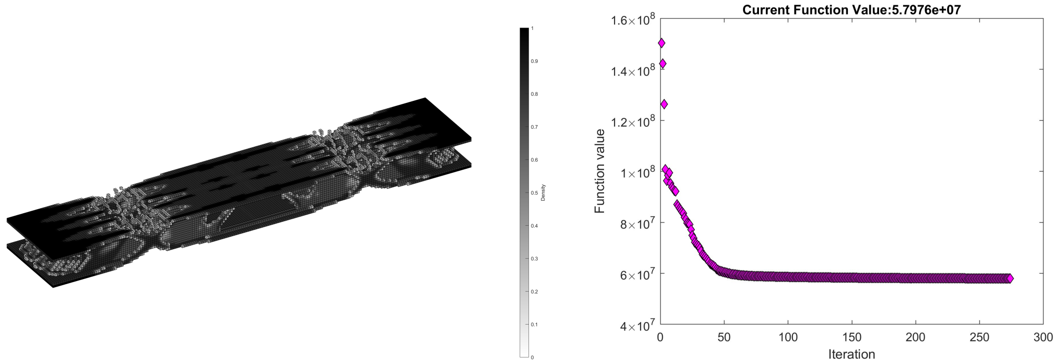
Figure A.26: Principal stresses in the volume optimized soda glass geometry, plotted for a quarter of the domain

A.6 Results Design Exploration

A.6.1 Geometries and Function Value Plots

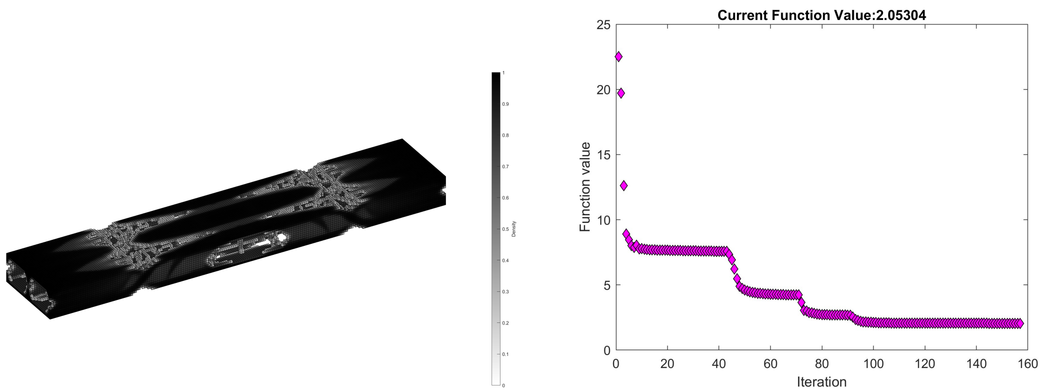
	Iteration	F count	F value	Feasibility	First order Optimality	Time (seconds)
Split Volume	274	518	$57.976e^6$	0	1546.25	86093
Split Compliance	157	352	2.053	0	$7.80e^{-5}$	28762
Complete Volume	135	304	$19.105e^6$	0	38194	4190
Complete Compliance	310	761	7.357	0	$18.61e^{-5}$	13746

Table A.4: Overview Optimization Output



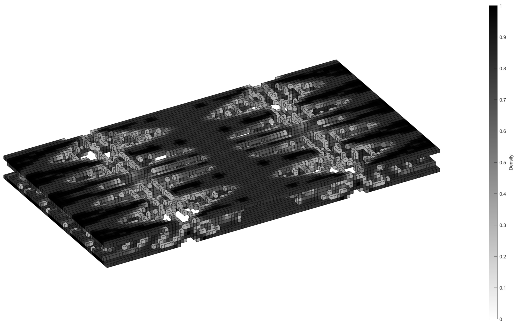
(a) Result Geometry **(b)** Plot of function value per iteration

Figure A.27: Result Volume Optimization, split domain

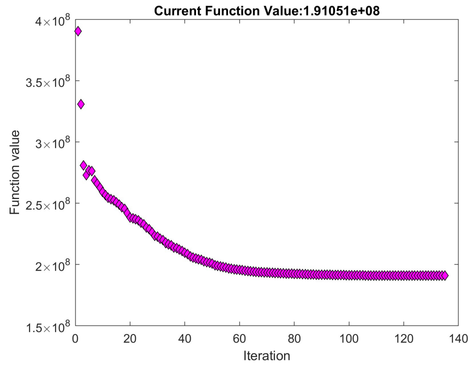


(a) Result Geometry **(b)** Plot of function value per iteration

Figure A.28: Result Compliance Optimization, split domain



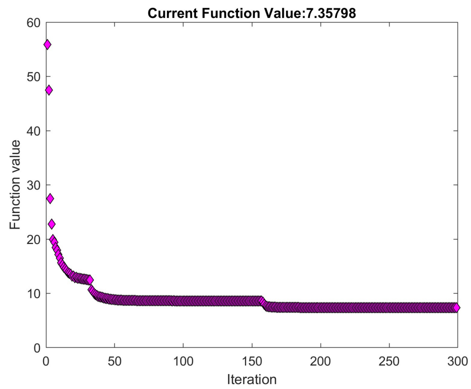
(a) Result Geometry



(b) Plot of function value per iteration

Figure A.29: Result Volume Optimization, Complete Domain

(a) Result Geometry

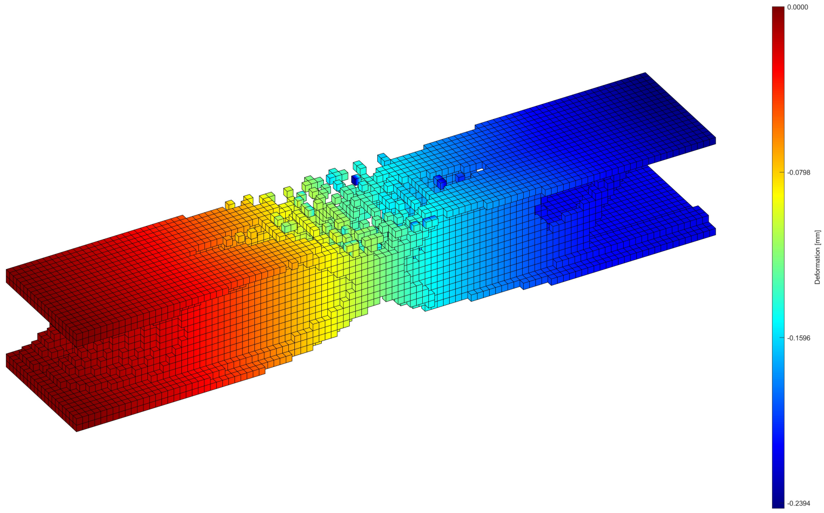


(b) Plot of function value per iteration

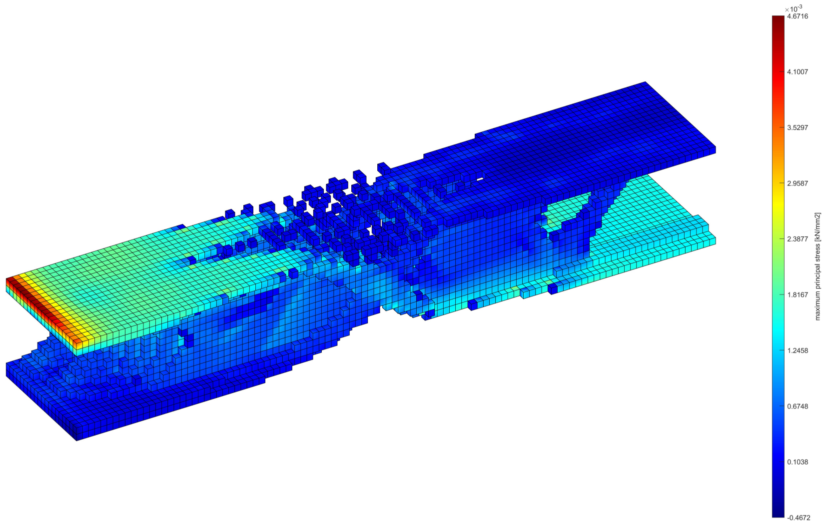
Figure A.30: Result Compliance Optimization, Complete Domain

A.6.2 Structural Results Design Exploration

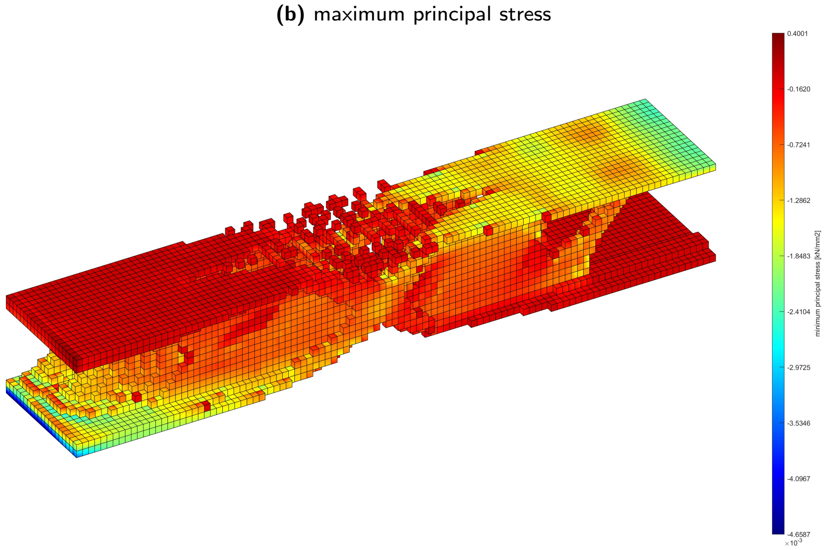
Split domain optimized with Volume Objective



(a) Deformation



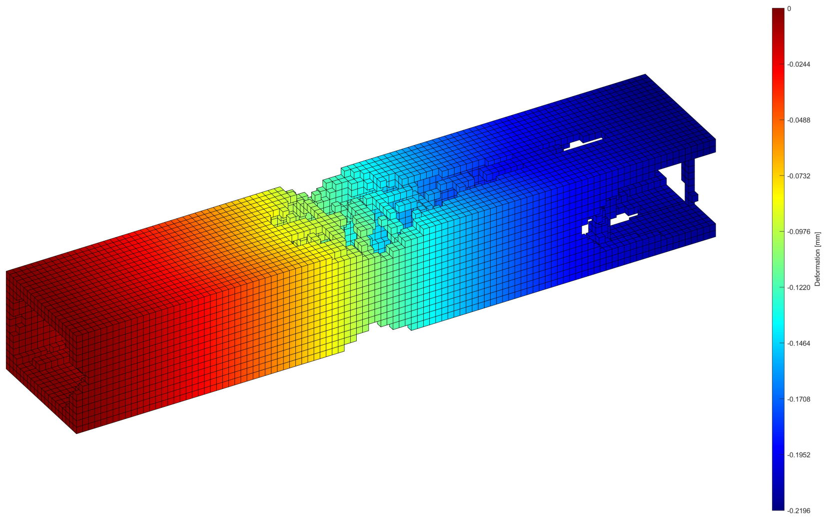
(b) maximum principal stress



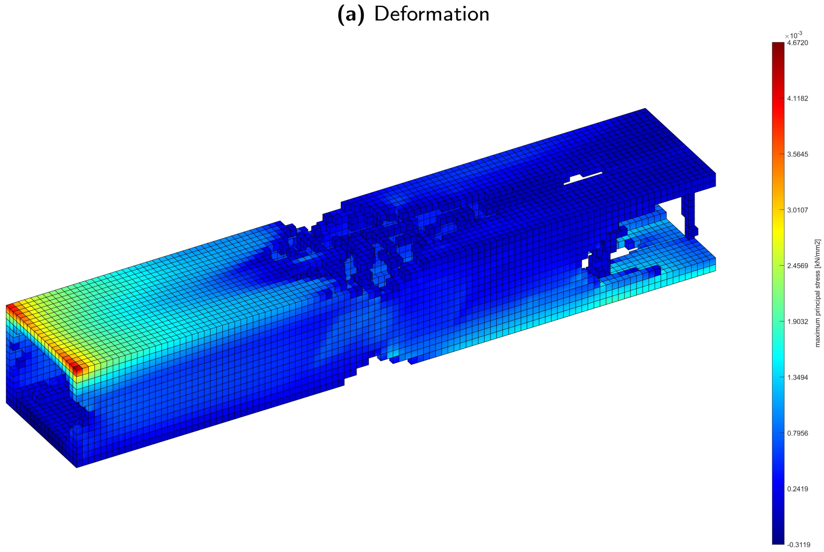
(c) Minimum principal stress

Figure A.31: Structural results volume optimization split domain, plotted for a quarter of the domain

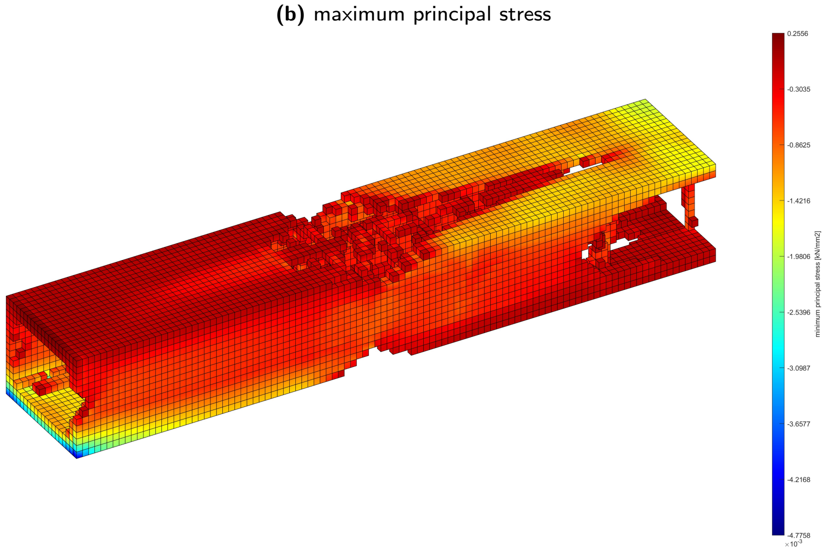
Split domain optimized with Compliance Objective



(a) Deformation



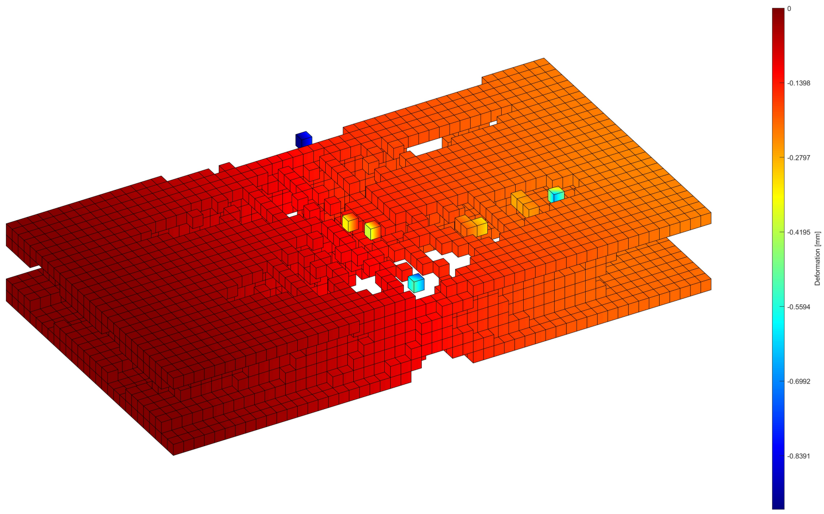
(b) maximum principal stress



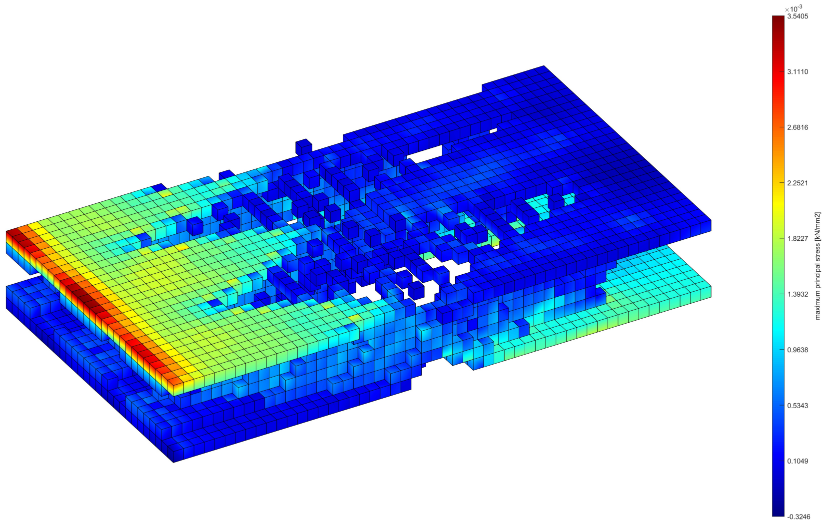
(c) Minimum principal stress

Figure A.32: Structural results compliance optimization split domain, plotted for a quarter of the domain

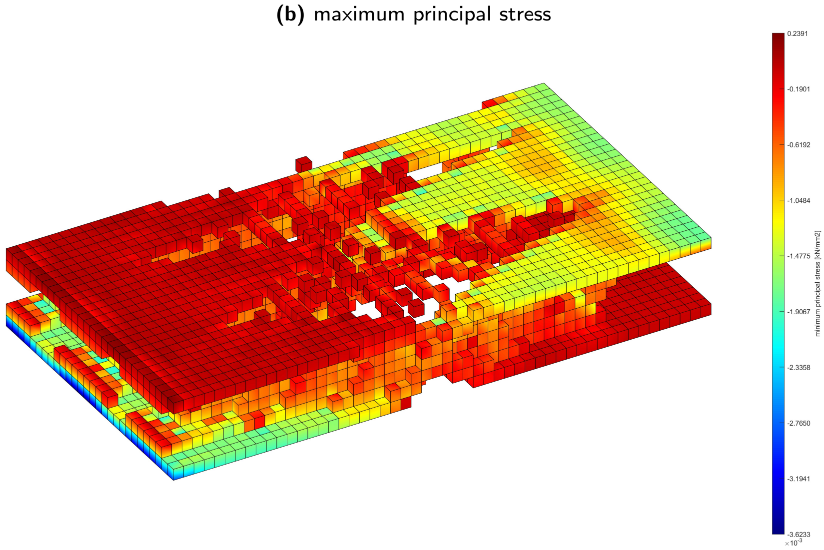
Complete domain optimized with Volume Objective



(a) Deformation



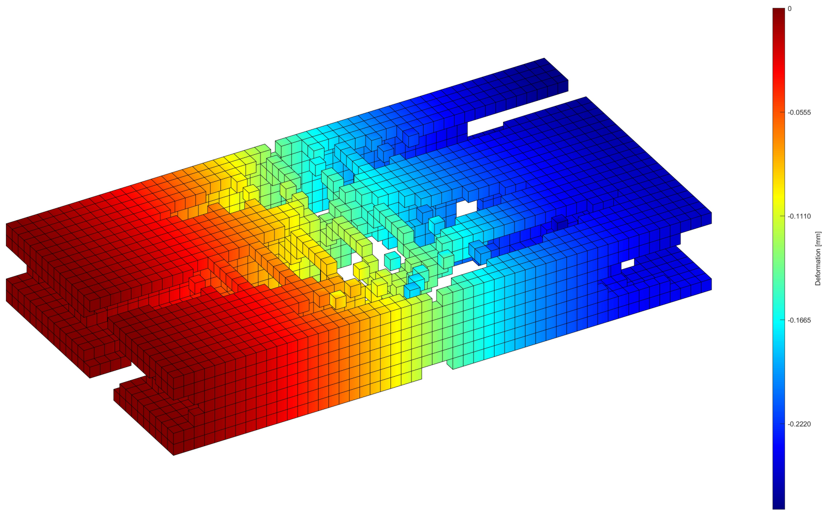
(b) maximum principal stress



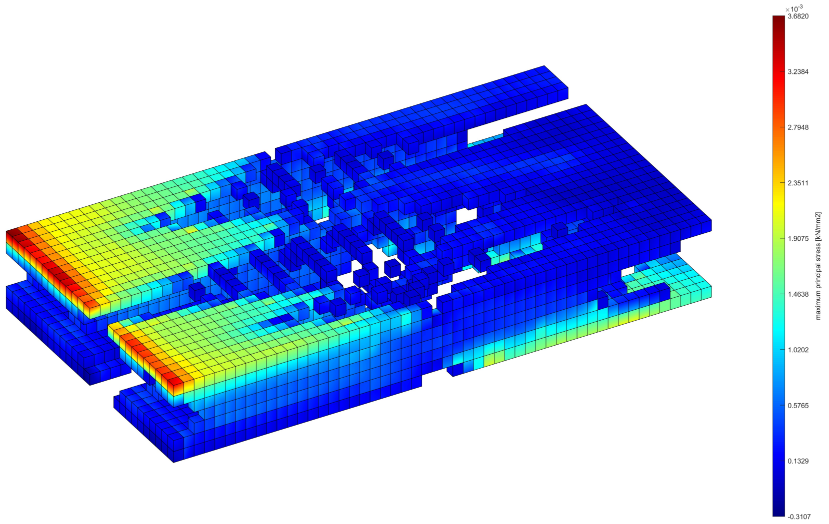
(c) Minimum principal stress

Figure A.33: Structural results volume optimization full case study, plotted for a quarter of the domain

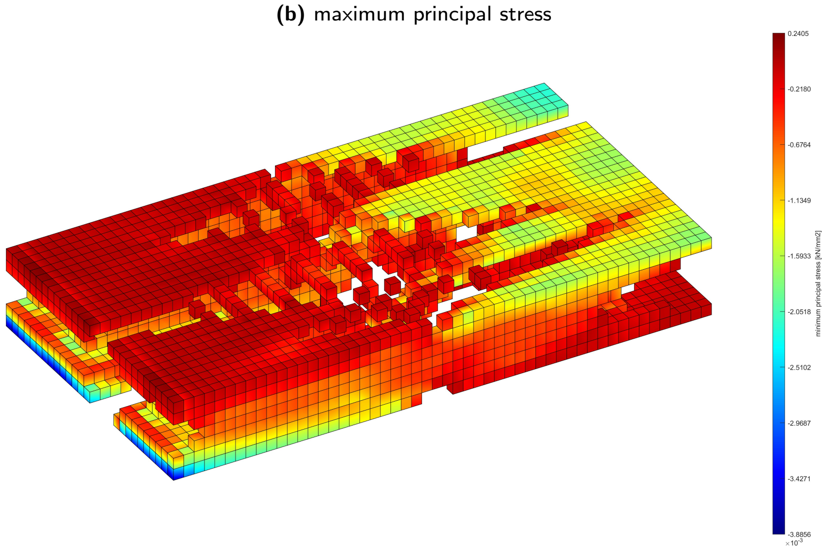
Complete domain optimized with Compliance Objective



(a) Deformation



(b) maximum principal stress



(c) Minimum principal stress

Figure A.34: Structural results compliance optimization full case study, plotted for a quarter of the domain

A.7 Structural Verification Post-processed Design

The final post-processed design has been verified using ANSYS. For this verification, one component has been calculated, with the inclusion of the additional layers of float glass.

A.7.1 Deformation and Stress

The following images show the ANSYS results. It has to be noted that, in order to be able to transfer the geometry to ANSYS, it had to be reduced significantly. Resulting in some geometrical errors. This can be seen in the results as well, where an uneven mesh discritization causes peak stresses at one location near the supports (figure A.38. However, even this peak stress is within the given limits of glass.

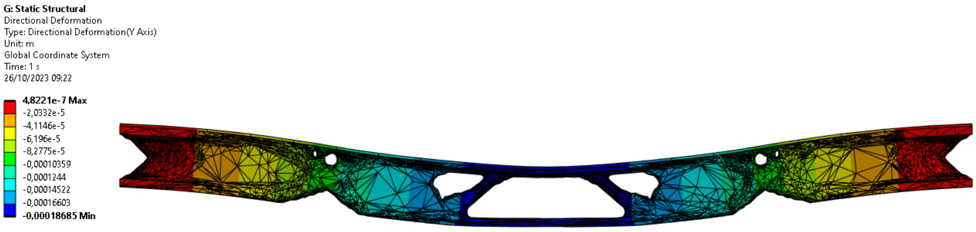


Figure A.35: Deformation post processed Geometry

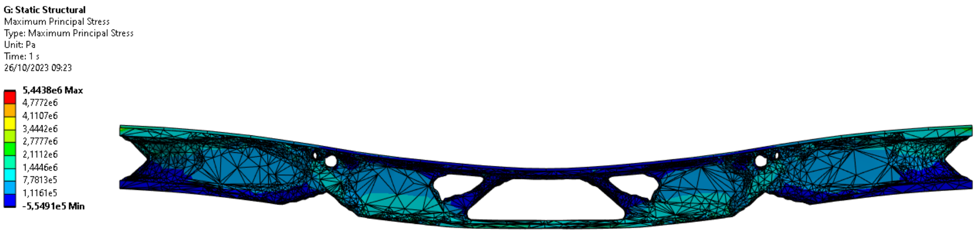


Figure A.36: Minimal principal stresses in post processed geometry

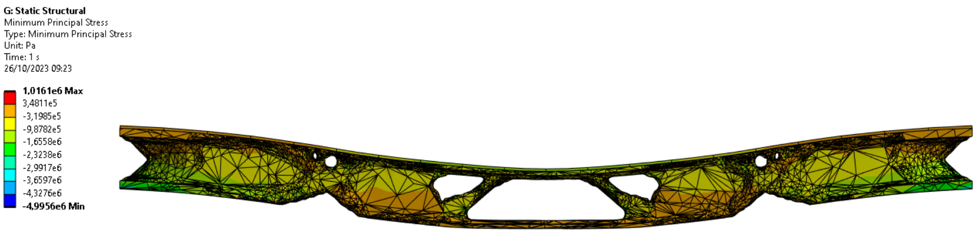


Figure A.37: Maximum principal stresses in post processed geometry

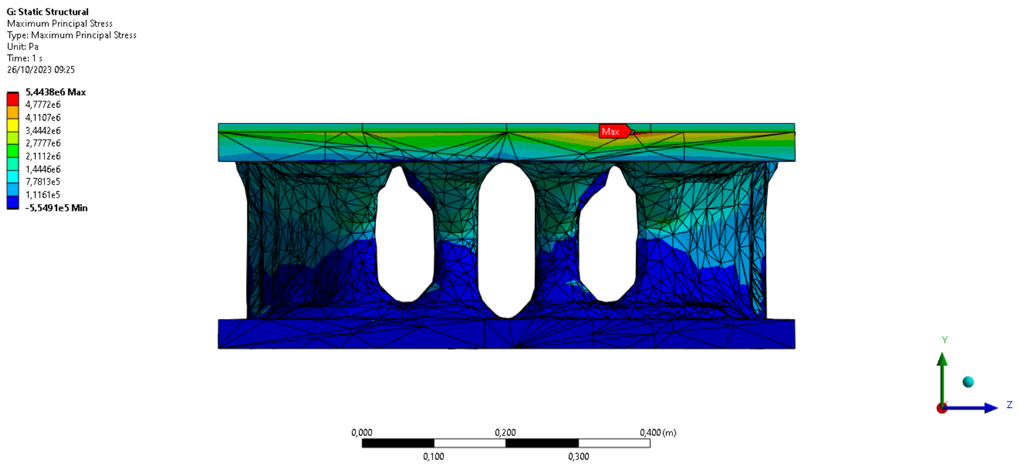


Figure A.38: Location maximum principal stress

A.7.2 Buckling

As the optimization is based on a static structural model, the final geometry has been checked for possible buckling. Using ANSYS for this check, the resulting buckling modes are all with a negative force (table A.5. Indicating that the structure will only buckle if an opposite force is applied. However, an additional hand calculation has been carried out to check two critical components of the structure (figure A.40 & A.41. These are the top plate and the diagonal members in the structure. An additional ansys model has been made of just the top plate (figure A.42), to again check for buckling. The resulting buckling modes (table A.6)indicate that a very high force is needed for the structure to buckle at this location.

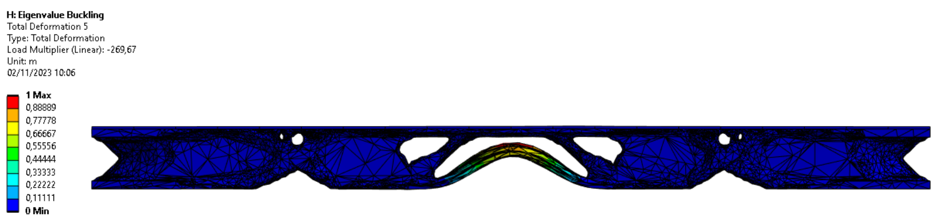


Figure A.39: Ansys result for Buckling under mode 5

Mode	Load Multiplier
1,	-1106,6
2,	-885,68
3,	-582,71
4,	-442,93
5,	-269,67

Table A.5: Resulting buckling modes Eigenvalue Buckling calculation full structure

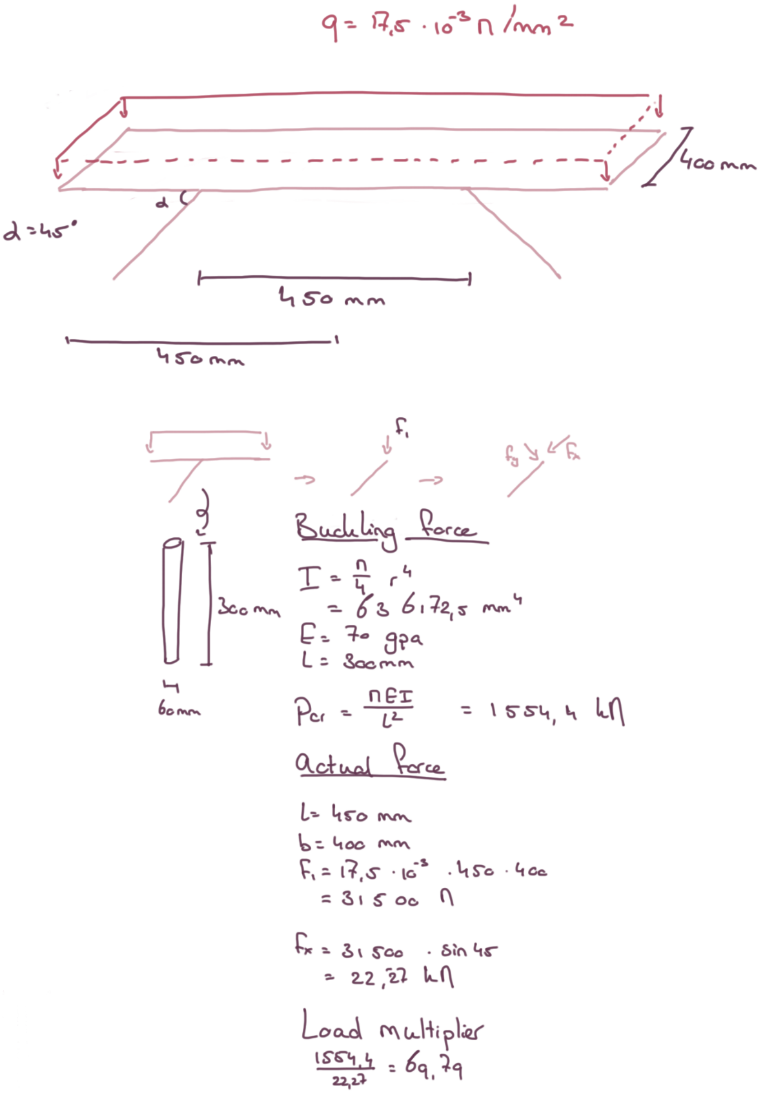


Figure A.40: Hand calculation buckling in bar

Mode	Load Multiplier
1,	20979
2,	21874
3,	22547
4,	23063
5,	23858

Table A.6: Resulting buckling modes Eigen value Buckling, calculation for top plate

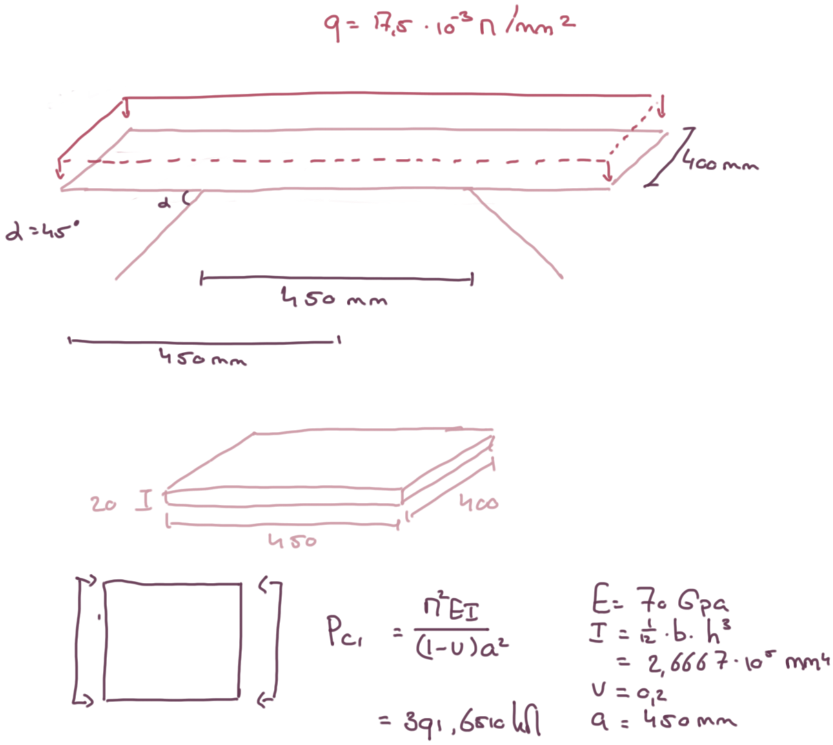


Figure A.41: Hand calculation buckling in plate

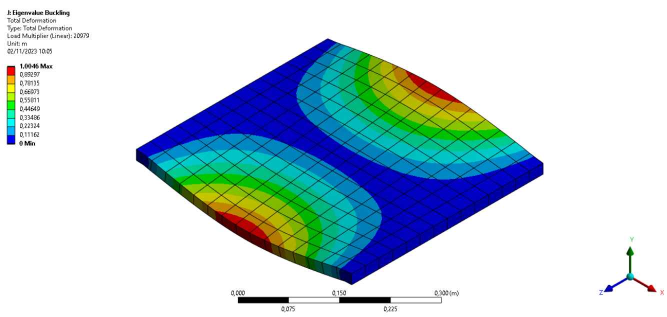


Figure A.42: Ansys result for buckling under mode 1

A.8 Calculation Shell Thickness Mould

Taking part design closest to the supports as simplified I section
web ≈ 0,06 m thick
0,8 m long

Thrust force on the side of the mould:

$F_A = p_a A = ((p_t + p_b)/2) A = (\rho g \frac{(h_t + h_b)}{2}) A$

F_A = avarage trust force
p_a = avarage pressure vertical suface
A = area bottom surface
$p_t; h_t; p_b; h_b$ = pressure / height at top / bottom

$\rho = 2500$ kg/m³
$g = 9,81$ m/s
$h_t = 0,3$ m
$h_b = 0$
$A = 0,06 \cdot 0,8$

$F_A = (2500 \cdot 9,81 \cdot \frac{0,3+0}{2} \cdot (0,06 \cdot 0,8))$
$= 176,72$ N

Bending moment

$M_{max} = F_a a = 176,72 \cdot \frac{0,3}{3}$
$= 17,672$ N m

Minimum thickness (stefanaki, 2020)

$\sigma_{bend} = \frac{M_y}{I} = \frac{M \frac{d}{2}}{\frac{1}{12} b d^3}$

$d = \sqrt{\frac{6 \cdot M}{\sigma_{bend}\, b}} = \sqrt{\frac{6 \cdot 17,672}{220 \cdot 10^4 \cdot 0,8}}$
$= 0,78$ cm

Figure A.43: Hand calculation shell thickness