Delft University of Technology
Master's Thesis

Optimising 3D Printed Concrete Structures

Concrete Additive Manufacturing & Topology Optimisation

P.A. Martens

TU Delft

BEM Next lab

TU/e Technische Universiteit Eindhoven University of Technology

Optimising 3D Printed Concrete Structures

Concrete additive manufacturing and topology optimisation

by

P.A. Martens

in partial fulfilment of the requirements for the degree of

Master of Science

in Civil Engineering

at the Delft University of Technology

An electronic version of this thesis is available at `http://repository.tudelft.nl/`

TU Delft BEM Next lab TU/e Technische Universiteit Eindhoven University of Technology

Student:	Pascal Alexander Martens
Thesis Committee:	*Chairman:* prof. dr. ir. J.G. Rots Delft University of Technology
	Company Supervisor: dr. ir. J.L. Coenders White Lioness technologies
	dr. ir. F.B. Bos Eindhoven University of Technology
	dr. ir. P.C.J. Hoogenboom Delft University of Technology
	prof. dr. ir.-arch. M. Schevenels KU Leuven
	dr. ir. H.R. Schipper Delft University of Technology

"Scientists study the world as it is;
engineers create the world that has never been."

- Theodore Von Karman

Preface

The Master's thesis that lies before you concludes my degree at the Delft University of Technology. It was written as the final part of my Master Building Engineering at the Faculty of Civil Engineering and Geosciences. The research was carried out in collaboration with the Eindhoven University of Technology, Delft University of Technology and White Lioness technologies.

During my research I was working part-time at the office of White Lioness technologies in Amsterdam. Being able to work there was really inspiring and motivating. I received a lot of help writing codes, which was very useful, since I never really used coding languages before. I would like to thank everyone at the office and especially the founders, Anke Rolvink and Jeroen Coenders, for this experience.

I would like to thank Jeroen Coenders as well for being my daily supervisor for the largest part of my research. He provided me with a fascinating topic and gave me the opportunity to write papers for several conferences. I wrote papers for the International Association for Shell Structures (*IASS*) and the International Federation for Structural Concrete (*fib*). I also presented my progress in Maastricht on the *fib* conference and in Delft for *Bouw 4.0*, an inspiring meeting of innovators and pioneers in the field of construction.

In addition, I would like to express my gratitude to all the members of my thesis committee for their feedback on my progress. I would like to thank Pierre Hoogenboom, Mattias Schevenels, Freek Bos, Roel Schipper and Jan Rots for their comments and their assistance. I would also like to thanks Jan Rots for being the chairman of this committee and Anneke Meijer for helping me organise the meetings.

The research started as a Lighthouse Project, a collaborative project between two technical universities (Delft and Eindhoven) and a company (White Lioness technologies). I would like to thank 4TU.Bouw for supporting this project and giving me the opportunity to present my work at the Infratech fair. In the starting phase of the project I collaborated with the student assistants of the BEMNext lab. I would like to thank Niels Hofstee, Michou Mureau, Vicente Plaza and Maarten Mathot for a Lighthouse project that was conducted in a positive atmosphere of cooperation.

Moreover, I am very grateful for the help of researchers and writers of the articles I read. I contacted some of them to assist me in my work. I would like to thank Matthijs Langelaar, Alexander Verbart, Krister Svanberg and Pierre Duysinx for sharing their expertise and helping me out whenever I got stuck in the process. I would also like to thank former student Christopher Magan for sharing his knowledge in the early stage of my research. Because of the information he provided after finishing his Master's thesis I was able to get off to a flying start.

Finally, I would like to thank my friends and family for supporting me over the time of my study and this research. Special thanks go to my parents who never doubted me and supported me in every way possible.

Hopefully this thesis is not merely the research that concludes my degree at the university. I hope, however, that it will be used as a small step towards more material efficient structures that will be constructed in a more automated way.

Thank you and enjoy reading my thesis!

Pascal Alexander Martens

Nieuwerkerk aan den IJssel, June 2018

Summary

Additive manufacturing and 3D printing are rapidly developing digital fabrication techniques (Lu et al. 2015). After the first steps in printing of metals (Frazier 2014) and plastics (Gibson et al. 2014) have been made, research from various groups around the world is now also focusing on printing in concrete (Lim et al. 2012) and moving to larger scales. Using this technique it will be possible to create customised concrete designs in one go at low costs and high construction speeds. Additionally, this new technology will provide opportunities to create more efficient structures. Structures can already be optimised in the early stages of the design for weight and structural performance, but the resulting optimised structures are often difficult to manufacture due to their shape. Additive manufacturing can be the key to make this possible without high costs for moulds and labour.

This thesis will present a novel methodology to include material and manufacturing constraints of 3D printed concrete in the optimisation process. The study examines the possibility to optimise concrete structures in the design phase. In order to save material and thus create more sustainable and more cost efficient structures, a topology optimisation tool has been created specifically for 3D printed concrete. Traditional topology optimisation methods consider isotropic and linear elastic material and will not necessarily produce realisable and reliable optimised structures. In the algorithm presented constraints of the printing process and material properties from physical testing of this layered material are both considered in the optimisation. By adopting this methodology more realistic and feasible optimal concrete structures can be designed.

The methodology is created from existing topology optimisation tools and improved to conveniently change the design domain, boundary conditions and loads. Furthermore, the existing tools are altered to create an algorithm that creates optimal designs for 3D concrete additive manufacturing. In topology optimisation a design domain is divided into elements. These elements are given a virtual density between zero (no material) and one (material). The aim is to minimise a certain objective function with respect to certain constraints. In most existing tools the total compliance is minimised (or stiffness is maximised) for a certain volume constraint. For some elements the virtual densities are steered towards material, for others to no material, which ultimately leads to an optimal topology.

An important change made in the method is the modification of the material model. By changing the compliance matrix the considered material is no longer isotropic. This leads to different optimised designs for different Young's moduli in different directions, as is the case with concrete layered manufacturing.

In topology optimisation different filters can be applied to avoid numerical difficulties. The robust filtering method is used in this research in order to obtain results that can

be printed. Due to this filter unprintable patterns and intermediate virtual densities are eliminated from the design.

In addition to the robust filter an additive manufacturing filter is incorporated in the method as well. This filter takes into account manufacturing constraints of the 3D printing process. The filter assures that a certain maximum overhang of 45 degrees is not exceeded. Because in some printing projects an overhang is not possible, a second filter with an overhang of zero degrees is added to the method as well.

To fully employ the possibilities of topology optimisation it is possible to change the objective and constraint functions of the optimisations. The standard objective to minimise the compliance can be changed to a volume minimisation. The volume constraint can be adapted as well. Because the method of moving asymptotes is applied and used as the solver it is even possible to use multiple constraints, for example a stiffness and a strength constraint. The strength constraint is incorporated in the optimisation as a global stress constraint. The von Mises stress criterion and the Drucker-Prager equivalent stresses can both be used to find an optimal design that does not exceed certain stress limits.

It can be concluded that by adapting all these changes to the standard topology optimisation methods, more realistic and feasible results can be created. Large amounts of material can be saved by combining topology optimisation and additive manufacturing of concrete. However, additional research is needed before this promising combination can be used for actual manufacturing. More knowledge should be gained about the material behaviour, for example the effects of the printing process on the strength and stiffness. Another aspect that needs attention is the link between the optimised design and the actual construction. Because the results of the optimisations are consisting out of solid and void elements, an additional step needs to be made to create actual printing paths for the 3D printer. Furthermore, the computational time needed to optimise large designs, or designs with stress constraints, has to be reduced.

Contents

Acronyms

3DCP	3D Concrete Printing
AM	Additive Manufacturing
BESO	Bi-directional Evolutionary Structural Optimisation
CC	Contour Crafting
CONLIN	Convex Linearisation
CP	Concrete Printing
ER	Evolutionary Rate
ESO	Evolutionary Structural Optimisation
FDM	Fused Deposition Modelling
FEA	Finite Element Analysis
FEM	Finite Element Method
GCMMA	Globally Convergent Method of Moving Asymptotes
KKT	Karush-Kuhn-Tucker
KS	Kreisselmeier-Steinhauser
LM	Layered Manufacturing
MMA	Method of Moving Asymptotes
OC	Optimality Criteria
SIMP	Solid Isotropic Material with Penalisation Method
SOMP	Solid Orthotropic Material with Penalisation Method
SQP	Sequential Quadratic Programming
TO	Topology Optimisation

1 Introduction

Architects and engineers have been designing challenging structures for ages. To predict the behaviour of these structures before they were built, designers have created physical models to make simulations of this behaviour or they simply designed based on their knowledge and experience from previous projects. Development of software and computers started in the late 20th century and created the possibility to easily design complex geometrical structures on the computer. The translation from these designs to actual structures however is still challenging, because not everything that can be designed, can be constructed in an efficient way. Using 3D software such as Rhinoceros with Grasshopper developed by McNeel and Associates (2017) and Dynamo with Revit by Autodesk (2018), every architect or structural engineer is now able to create buildings and components in forms only limited by their imagination. In addition to the architectural design, structural calculations can now be completed with software as well. An example of a structural analysis program is GSA by Oasys (2017). Despite all the positive improvements the construction and manufacturing of the elements, for example double curved surfaces or complex joints, is a part of the design process that still needs more attention. The construction process is still material and labour-intensive and therefore costly.

In addition to the fact that designers and engineers can use the available techniques to create aesthetically pleasing models, software can also be used to optimise the structures. Structures can be optimised in many ways. One method is the topology optimisation method (Bendsøe and Sigmund 2003). This computational technique makes it possible to distribute a limited amount of material in a design space. In this way connections, building parts or even entire structures can be optimised. As a result of optimisation it might be possible to save large amounts of material, resulting in more sustainable and cheaper structures.

The resulting structures of topology optimisation are often geometrical complex structures and they are therefore difficult to manufacture or they need to be redesigned prior to manufacturing (Atzeni and Salmi 2012). Additive manufacturing or 3D printing can be the solution to actually construct the outcomes. Additive manufacturing or AM adds

a certain material at the necessary positions in a design space, whereas current building processes mainly remove material from the unnecessary positions, resulting in additional waste and costs.

This research project will provide insight in the current topology optimisation methods, the current concrete AM projects and the possibilities of combining these two processes for the building industry.

1.1 Problem definition

Topology optimisation is a useful tool to design more efficient structures (Bendsøe and Sigmund 2003). Concrete Additive Manufacturing can help overcome technical difficulties in manufacturing these special structures. By applying optimisation tools and additive manufacturing, material can be saved. This can lead to the production of more sustainable and cheaper structures. The problem in this approach can be formulated as:

Current topology optimisation methods assume isotropic material behaviour while in reality printed concrete is not isotropic because of the layered built-up. Current optimisation methods will therefore come up with incorrect optimised structures for printed concrete. The correct material properties should be incorporated in the optimisation.

In addition to these material properties the manufacturing constraints can influence the design as well:

Designs created by optimisation methods can be hard, expensive or sometimes even impossible to manufacture. Manufacturing constraints should therefore be part of the optimisation algorithm. If the difficulties in manufacturing are only considered after the optimisation is performed, the structure needs to be adapted to a manufacturable design. In this case the adapted design is not an optimal structure any more. It might even be infeasible to manufacture the structure.

1.2 Domain and scope

In this research different concrete AM projects and several topology optimisation methods will be explained. However, this study will mainly focus on the effects of changing material properties and manufacturing constraints on the outcome of topology optimisation. Existing open source topology optimisation algorithms from amongst others Liu and Tovar (2014) and Andreassen et al. (2011) will be studied, adapted, improved and used to study the effects. Only simple rectangular (2D) and box shaped (3D) design spaces will be investigated.

As stated in the problem definition the material considered is *concrete* or a concrete-like mixture. Although the problem applies to other materials as well, this research will only consider printed concrete. Combinations of materials, for example steel reinforcement bars in the concrete, are not considered either.

Physical testing is outside the scope of this research. Only computer models will be

created and verified using available literature.

1.3 Research question

The main research question of this thesis is formulated as:

What possibilities can concrete additive manufacturing provide in the design and construction of topology optimised structures?

1.4 Research objectives

The following research objectives for this thesis arise from the described problem:

1. *To provide insight in the current state of concrete additive manufacturing and to give an overview of the topology optimisation algorithms currently available.*
2. *To create a material model of the printed concrete, taking into account the different properties in different directions.*
3. *To develop or adapt an algorithm that can be used to optimise a concrete structure or structural component, considering the material model created and the limitations of the concrete printer.*
4. *To design and optimise a concrete structure or component using the created algorithm.*

1.5 Thesis outline

- *Concrete additive manufacturing*
  A clear overview is given in recent and current concrete additive manufacturing projects all over the world. The most ground-breaking and promising projects will be discussed briefly.
- *Structural and topology optimisation*
  In this part of the research a basic introduction is given in the field of optimisation. Some general optimisation problems and methods will be covered, followed by several structural optimisation procedures. The chapter will be concluded with the most promising topology optimisation methods.
- *Development of an improved optimisation algorithm*
  An available standard topology optimisation method will be taken as starting point for an improved algorithm. In this part of the report a work-flow is created to easily create a design problem, solve that problem and visualise the results. The optimisation algorithm will be adapted to come up with more realisable and reliable designs. To make that possible, new filters, solvers, manufacturing constraints and a new material model will be added to the algorithm.

- *Implementation*
  In this chapter an overview of the utilisation of the method will be given. In a step-by-step manner the work-flow is explained by schemes, figures and examples.

- *Results and verification*
  Results of the created algorithm will be presented in this part of the study. Some parameters will be varied and the results of these variations will be compared and evaluated. Different design problems (loads and boundary conditions) will be used to clarify the effects.

- *Case Study*
  In this study a floor slab will be optimised using the created method. The results will be compared with regular slabs. To verify the results the regular slabs will be calculated using a plate theory as well.

- *Conclusions*
  In this part of the thesis the research question will be answered and some general conclusions will be given.

- *Recommendations*
  Finally, the recommendations for further research will be made. This will conclude the research.

2

Concrete Additive Manufacturing

Additive manufacturing, also referred to as 3D printing, is a rapidly developing process in manufacturing, however it is not widespread adopted in the building industry at a large scale yet. All over the world separate projects are being conducted regarding 3D printing of concrete. From a company in China that is printing houses (Charron 2015) to a man printing a castle in his own backyard (Rudenko 2016b). This chapter is a "state-of-the-art" and will give an overview of this relatively new manufacturing process and the most ground-breaking and promising projects.

2.1 3D printing process

The technology that physically builds up 3D models layer by layer is called additive manufacturing or 3D printing. In standard terminology additive manufacturing is described as: "a process of joining materials to make objects from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing methodologies" (ASTM International 2013). The foundation of the 3D printing technique was built around thirty years ago. Chuck Hull invented a solid free-form technique called stereo-lithography (Melchels et al. 2010). It was patented as "a method and apparatus for making solid objects by successively 'printing' thin layers of the ultraviolet curable material one on top of the other" (Hull 1986). He soon realized the method was not limited to ultraviolet curable material and he adapted the definition of the patent to "any material capable of solidification or capable of altering its physical state". Many materials and processes were applied since that period and lately research started focussing on concrete or concrete-like substances as the printable material (Pegna 1997).

Additive manufacturing typically starts with an idea of the structure or product the designer wants to manufacture. This idea is than transformed into a 3D computer model. This model is subsequently being prepared for manufacturing. A machine or 3D printer then creates the physical model layer by layer. The process is called additive because the

machines add material at the locations determined by the 3D model.

2.1.1 Advantages of concrete additive manufacturing

This manufacturing technique has multiple important advantages over methods commonly used in the building industry at the moment:

- *Geometrical design freedom*
  In theory additive manufacturing provides designers with more geometrical freedom to design models compared to existing manufacturing techniques, such as pouring concrete in traditional moulds. The geometry can now be chosen in such a way that the design is aesthetically pleasing, functional and optimised in terms of structural behaviour.

- *Easy to design and create customised items*
  Another advantage of 3D printing is the possibility to easily fabricate customised and non-repetitive elements. Computational modelling techniques can easily divide a design in many customised unique elements or parts, however the manufacturing of these items can be very expensive and time consuming, because of the unique character of the elements. When constructing with a 3D concrete printer, tools and moulds for each single element will not be needed anymore.

- *Automated and efficient process*
  Using additive manufacturing in constructing concrete designs can be a large step in the automation of the building industry. In product design and manufacturing automation, the use of digital models and the help of robots, is current practice, however the building industry is conservative. Additive manufacturing can contribute to the efficiency of the industry. For example, 3D concrete printers can print concrete throughout the day and night which can reduce construction time. Additive manufacturing can also reduce the amount of errors during construction. Using this technique only the preparation of the model has to be done manually, the machine needs to be placed and the concrete printing material has to be transported to the machine. More human interference is hardly needed, which results in less human errors.

- *Sustainability*
  As mentioned above, material is added to a design space in this production method. It is therefore possible to avoid an abundance of material in a structure. Only the material that is actually needed can be placed. Besides, most past and present manufacturing techniques such as milling, drilling and cutting typically remove material where it is not needed. These subtractive processes cause more waste material and additional work, which results in less sustainable results and higher costs.

Additive manufacturing of concrete can thus potentially become a more sustainable, cheaper, faster and easier manufacturing method with more design freedom and possibilities than current construction methods. However, there are some disadvantages and challenges that need attention.

2.1.2 Challenges and disadvantages of concrete additive manufacturing

The innovative building concept of 3D printing might replace existing manufacturing methods in the future, however some challenges must be solved first.

- *Scale of construction*
  3D printing of concrete is inspired by the already existing 3D printing process in other materials. One of the most important differences and challenges is the size of structures compared to the size of the small models printed before. Layer thicknesses need to be significantly larger in order to print parts within a reasonable timespan. Concrete layers can now be printed with a thickness of about 9 to 40 millimetres. A 9 millimetre thickness is already 100 times larger then some plastic layers currently being used in small scale rapid prototyping (Kestelier 2011). Not only the material should be able to handle the larger scale, but the printing machines should be larger as well.

- *Higher cost for large production runs*
  Because the technique is in development production is still very expensive. When the methods and machines are fully developed and more experience is gained, the production costs will presumably decrease. For larger production runs it will probably always be more cost efficient to use a mould, because of the advantages of mass production.

- *Production time*
  Additive manufacturing is a production method which builds up a structure layer by layer. The velocity of the nozzle, the end part of the concrete printer, must have a maximum value. If the nozzle moves too fast, not enough concrete is placed at a certain position. Another aspect which is important considering the speed is the hardening time of the concrete. The concrete should harden just enough before the next layer is placed on top. If the time in between the placements is too long, the concrete hardens too much and the layers will not work together properly. If the time interval is too short, the concrete is still wet and will not support the next layer in a proper way (Lim et al. 2009).

- *Limited tensile strength*
  Additive manufacturing of concrete is using a single printing material: concrete or a concrete-like mixture. For this reason no reinforcement can yet be applied using the printer. Structural concrete always contains reinforcement bars, for resisting tensile forces in the structure. Fibres can be added to the mixture to increase the strength, however this will not be enough in most cases. It is also possible to place steel cables in between the concrete layers. Before safe and real printed structures can be built some kind of reinforcement, testing and verification is needed. Regulations regarding this new manufacturing process should be created.

- *Lower precision*
  Another result of the coarse layering process is a lower precision. Moulds can be very expensive and labour-intensive, however they can help create more exact structures than the 3D printing process. This aspect will certainly improve over the years.

2.2 Recent 3D printing projects

After concrete additive manufacturing was first mentioned by Pegna (1997) several projects started all over the world. The most ground-breaking and promising projects are briefly described in this part of the chapter.

2.2.1 Contour Crafting

Contour Crafting (CC) is a 3D concrete printing technology, proposed by professor Behrokh Khoshnevis of the University of Southern California. The first publications on this fabrication technique are from the late nineties (Khoshnevis 1998).

Automation has grown in almost all production domains other than construction of large civil structures and their sub-components. Aim of the CC project was to automate the construction process as well, in order to improve the speed, labour efficiency, durability and to lower the accident rate and the cost of construction (Khoshnevis 2004).

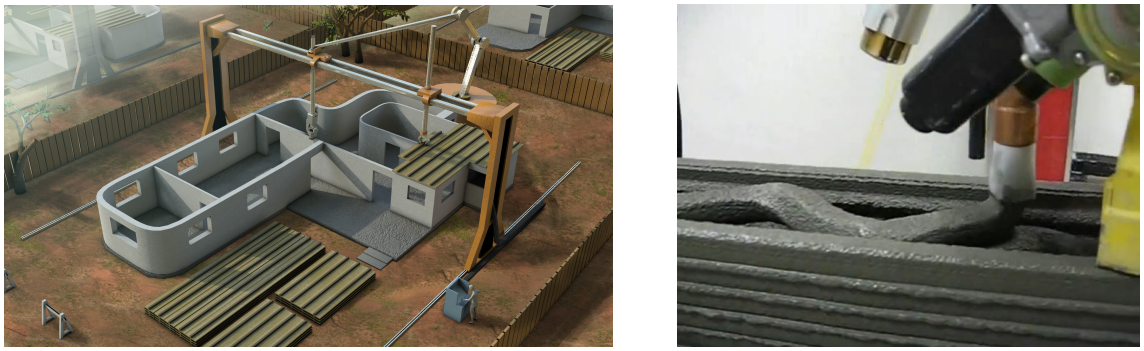


Figure 2.1: Contour Crafting by Behrokh Khoshnevis (contourcrafting.org 2014)

Khoshnevis used the idea of the already existing 3D printing or additive manufacturing processes and scaled these processes up. Instead of plastic or steel, polymer, ceramic slurry, cement and a variety of other materials and mixes are used to print large scale objects with a smooth surface finish. The process is, similar to the original 3D printing process, based on Layered Manufacturing (LM). The CC machines are equipped with two trowels which act as solid planar surfaces. They constrain the extruded mix in vertical and horizontal direction and can thus create exceptionally smooth and accurate surfaces on the object being fabricated (Khoshnevis 2004).

During or after the extrusion of the concrete layers, other objects can be placed by machines, such as reinforcement bars, plumbing pipes and mechanical installations. In this way human interference is minimised even more.

The research also addresses the application of Contour Crafting in building habitats on the Moon or even on other planets. CC can be used to create Lunar structures which provide radiation protection. By applying the CC technique the shielding structures can be built utilising in-situ resources in advance of a manned landing (Khoshnevis et al. 2005).

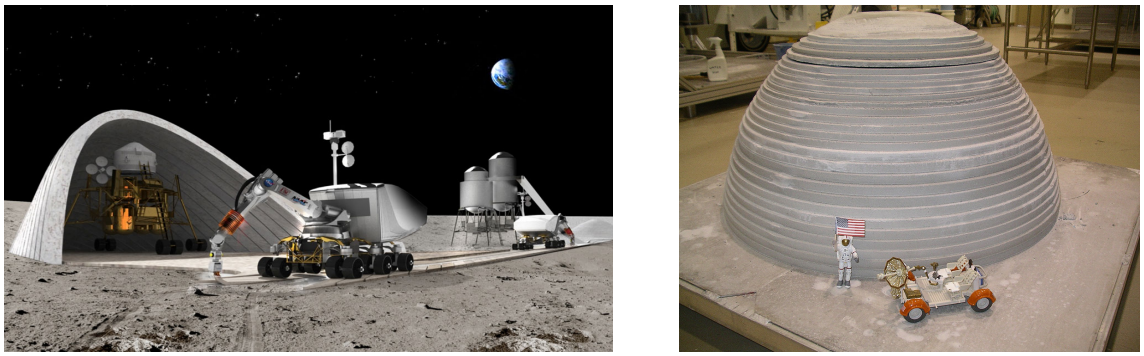


Figure 2.2: Lunar Contour Crafting (contourcrafting.org 2014)

2.2.2 WinSun – Shanghai, China

Ma YiHe is the CEO of WinSun Decoration Design Engineering Co, founded in 2002. Ma used a technique similar to Contour Crafting to actually build houses in printed concrete. Building-parts are first printed and then assembled as prefabricated parts.

Ma claims to have created ten 3D printed houses within 24 hours (Griffiths 2014). The company aims to build sustainable houses by printing in layers of construction waste, industry waste or mining residual mixed with cement. The firm owns around a hundred national patents of construction materials. By using their technique WinSun expects not only to contribute to a sustainable future of the building industry, but they expect construction companies to save up to 50% of their building costs as well.

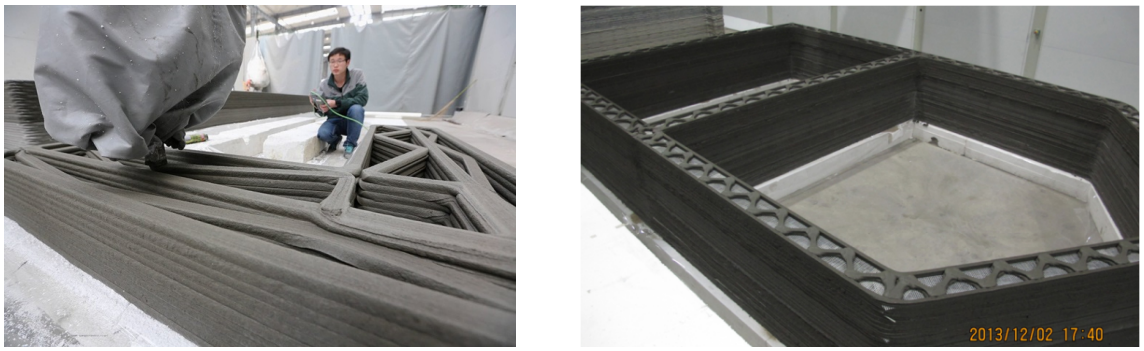


Figure 2.3: 3D printed house by WinSun (yhbm.com 2013)

After the manufacturing of the printed houses, WinSun also built a 3D printed neo-classic villa and a five-storey apartment building (Charron 2015), the world's highest 3D printed residential building. Their latest achievement is a printed Chinese courtyard, inspired by the ancient Suzhou gardens (Buren 2016). According to Dr. Khosnevis the Chinese Ma YiHe is "faking" his projects (Krassenstein 2015). Ma YiHe initially wanted to collaborate with Khosnevis, however after Khosnevis visited China and rejected the collaboration Ma YiHe used the technology he was able to learn from him. The creators of Contour Crafting state that: "They are not 3D printing homes or apartment buildings. Instead they are printing small sections of walls, within their own facility using a very expensive 5-axis gantry system from Italy, which they then simply fit with a concrete pump. The extremely heavy walls then need to be loaded onto a truck, transported to a

building site, and then offloaded and constructed. According to both Dr. Zhang and Dr. Khoshnevis, this technology, which does not have a single patent backing it, is neither efficient nor revolutionary, and instead is much more expensive and inefficient than current manufacturing techniques" (Krassenstein 2015).

Figure 2.4: 3D printed apartment buildings by WinSun (yhbm.com 2014)

2.2.3 TotalKustom – Rudenko USA

Andrey Rudenko, an American contractor, started TotalKustom in Minnesota around 2012 to "develop robotic systems that will facilitate the construction of affordable, faster, zero-energy, and smarter housing" (Rudenko 2016a). Rudenko wanted to create a light, portable and stable printing machine. He started using plastics as material printed by a relative small printer. Then he experimented with larger printers and he started using variations of cement mixes. He claims that after all the testing and tuning of the printer he now can print nearly perfect layers. In comparison with Contour Crafting these layers are much smaller in height (5.0 mm) and width.

In 2014 he printed a concrete castle using mostly his own resources (Krassenstein 2014). He printed the castle in parts in his backyard and afterwards the turrets were lifted and placed on top of the walls. This was a difficult task, so the aim for a next project was to build a structure in one piece. Additionally, he did not want to interrupt the printing process in a successive project. Whenever the printer works 24 hours a day, the construction time decreases greatly.

Figure 2.5: 3D printed castle and villa by Andrey Rudenko (totalkustom.com 2012)

A year later he was asked to manufacture a villa for the Lewis Grand Hotel on the

Philippines. Rudenko prepared his printer in Minnesota, brought it to the Philippines and started printing with a different concrete mixture containing local materials. A component of the mixture was sand with volcanic ash, which lead to satisfying results; the walls became pretty strong and the bonding between the layers was good. The printed hotel suite measures 10.5 by 12.5 meters with a height of 4.0 meters. It includes several rooms and a 3D printed Jacuzzi (Buren 2015).

2.2.4 Concrete Printing (CP) at Loughborough University

At Loughborough University the development of 3D concrete printing started around 2007 (Buswell et al. 2007). Similar to Contour Crafting the idea was to scale up the already known additive manufacturing technology. However, this project is mainly focussing on construction-scale components and the properties of the printed concrete, rather than focussing on the construction of complete buildings. Another difference is the accuracy of the printer. The printer in Loughborough is much more precise than the coarse printer used in Contour Crafting. The diameter of the used nozzles varies between 4 and 22 mm. Because of this precision the printer can achieve a higher 3D freedom and is thus capable of creating small voids in the structure. These voids can be created to save material or to place the building services.

In 2010 a reinforced "Wonder Bench" which weighs about one tonne was printed by their 3D printer (Lim et al. 2011). In 2011 a two square-meter curved panel was printed. Right now the printer can produce components in a build volume of up to 2.0 m x 2.5 m x 5.0 m (Corke 2010).

Figure 2.6: Concrete Printing at Loughborough University (lboro.ac.uk 2011)

The materials used in the CP investigation are cement and gypsum based, because of the common use of these materials in the building industry. The used material has an approximate density of 2300 kg/m^3 and produces a high strength material which is more than three times as strong in compression and in flexure as conventional cast construction materials (Lim et al. 2011).

In 2014 Skanksa, a large Swedish construction and development company has signed an agreement with the Loughborough University to develop, build and commercialise a 3D concrete printing robot (Knutt 2014).

2.2.5 Eindhoven University of Technology

The goal of the research program at the University of Technology in Eindhoven, called 3D Concrete Printing (3DCP), is "to establish concrete printing as a viable new method to manufacture concrete elements and buildings, and to fundamentally understand its processes" (Salet 2016). The research group operates and develops its own 3D concrete printer at the university. The printer consists of a four axis gantry robot which measures 9.0 m x 4.5 m x 3.0 m. The robot is linked with a mixing pump. The robot and the pump are both controlled by a numerical controller.

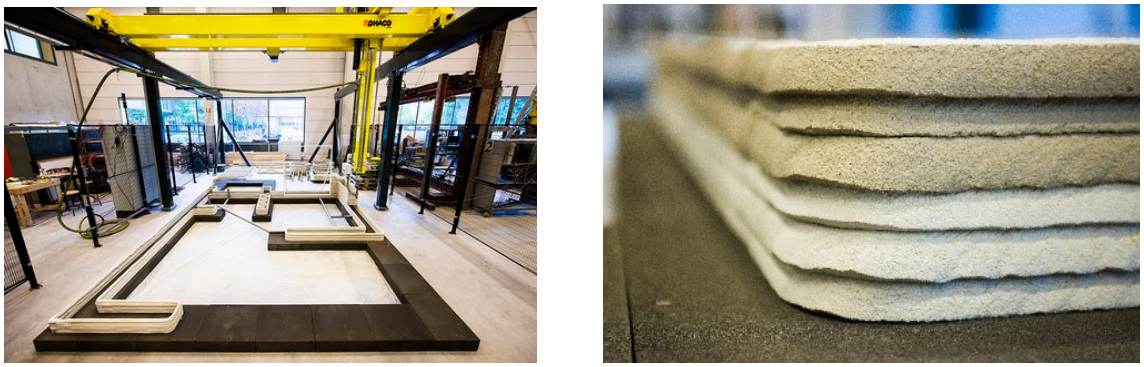


Figure 2.7: 3DCP at the University of Technology Eindhoven (tue.nl 2016)

The printing process is very similar to Contour Crafting, however the focus in Eindhoven is on understanding and controlling the process and the material instead of building real houses.

Figure 2.8: Different results of 3DCP (tue.nl 2016)

Nevertheless, together with the BAM Infra construction company they also printed a bicycle bridge at the university. The bridge is reinforced, pre-stressed and built up with about 800 layers of concrete. Using this technology the eight-metre bridge was created with less material, less waste and less human interference than in the conventional technique in which a mould is filled.

Figure 2.9: 3D printed concrete bridge in Gemert (verkeerinbeeld.nl 2017)

2.2.6 D-Shape

Enrico Dini, the inventor of the D-Shape printer, is an Italian civil engineer. He developed a 6.0 x 6.0 meter printer with about 300 small nozzles and a height of 3 meters. The printer is just as the previously discussed machines printing per layer (Lim et al. 2012).

The difference between this method and the previous discussed processes is that D-Shape solidifies sand, pre-mixed with a catalyst, by a chemical reaction between the sand and an inorganic seawater and magnesium-based inorganic binder. In contrast to the other projects, the printer does not contain a material extruder. After a 3D design has been made in the computer, the design is sliced in 5 millimetre thick layers. After the CAD process is completed the manufacturing starts. A sand bed is deposit on the building site. The sand is spread and flattened, before the binder is applied to the sand. A new layer is finished. This process repeats itself until the model is completed. Finally the supporting powder bed is removed, some additional binder is applied where needed and the surface is treated. Now the structure is finished. The final product is a mineral-like material with a micro-crystalline structure. Compared to concrete, the structures created with the D-Shape have a relative high tension resistance and contain no iron reinforcement (Colla and Dini 2013).

Figure 2.10: 3D printer and a printed structure by Enrico Dini (d-shape.com 2012)

Dini first used epoxy as binding material, however changed that to the magnesium-based one after experiencing problems. In the future, Dini aims to use the printer to create full-scale buildings (Dini 2016).

Similar to Behrokh Khoshnevis, the inventor of Contour Crafting, Enrico Dini is investigating a way to use his printing method on the moon. The program, called "3D printing building blocks for lunar habitation", has the main objective to investigate whether 3D printing of moon dust is a viable construction technology for possible future lunar colonisation (Ceccanti et al. 2010).

2.2.7 WASP

Another Italian project, the World's Advanced Saving Project, or WASP "was born with the dream of printing houses with 100% natural materials or 'zero-mile' homes, using materials found on the surrounding area" (Langenberg 2015). The project started in 2012. During the research a group of engineers and researchers concluded that printing entire buildings does not make sense yet. Therefore the group focused on printing individual structural units, especially structural beams. In the current construction process of concrete beams a large amount of CO_2 is produced. A ton of cement generates about a ton of carbon dioxide (Moretti 2015). To diminish the emission of CO_2 during the manufacturing of beams, the group designed a 3 meter long lightweight modular reinforced beam. The designs they created, modelled with smart software, got rid of the redundant material. In this way only half of the total amount of CO_2 produced in regular manufacturing processes is released into the atmosphere. The beam is build up from separately printed modules, later connected to each other.

Figure 2.11: 3D printed modulated concrete beam (wasproject.it 2015)

The research group created different 3D printers. The first printer was the PowerWASP, a fast 3D printer that mills wood and aluminium. It is possible to equip the PowerWASP with a syringe to print ceramic mixtures. The PowerWASP was succeeded by the range of DeltaWASP printers, which were small and larger printers with a high precision and speed (Moretti 2016). The modular beam discussed before is created with a 4.5 m high delta printer. The DeltaWASP printers are equipped with an extruder for printing ceramic materials. In 2015 WASP created a 12 meter high delta 3D printer, the Big Delta. WASP considers it important that their printers are portable and feature low energy consumption (Thompson 2015).

Figure 2.12: 3D printers created by WASP (wasproject.it)

2.2.8 CyBe Construction

CyBe Additive Industries is a company from The Netherlands, started in 2013 by Berry Hendriks. He was inspired after a presentation of Behrokh Khoshnevis and wanted to create his own concrete 3D printer. His goal was to "develop an industrial, mobile, and modular 3D printer together with printable materials, like concrete" (Wheeler 2015). CyBe is combining the original Contour Crafting technique with a special robot. This robot has a range of 3.15 meter and can simply adjust the speed and the angle of extrusion. The company also uses a special type of mortar, created by one of their partners which reaches a bearable strength within minutes (Goehrke 2015).

CyBe collaborated with Heijmans to create and test useful construction elements. Recently they created a 3D printed mould together. The concrete mould will be filled with reinforcement and poured concrete, however the mould will not only function as temporary support, it will remain a part of the structure as well (Hammer 2016).

Figure 2.13: 3D printed concrete moulds (cybe.eu 2016)

2.2.9 IAAC Minibuilders and Large Scale 3D Printing

The IAAC, or Institute for Advanced Architecture of Catalonia uses a very different approach. Most 3D concrete printers are large and the products they create are smaller. An issue in this approach is the fact that the size of the used printer should always exceed the size of the structure, which can lead to huge printers. The IAAC uses small mobile robots

to print large structures. In the process they make use of three different small robots, the "builderbots". The largest of the three only measures about 42 *cm*. Next to the three small robots there is a "Supplier Robot", which provides the liquid building material to the smaller robots (Jokić and Novikov 2016).

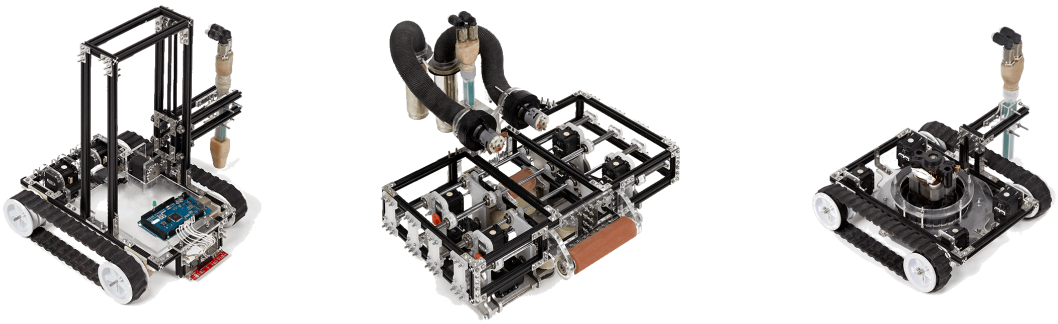


Figure 2.14: IAAC "Builderbots" (robots.iaac.net 2014)

The first small robot is the "Foundation Robot" which constructs the first 15 *cm* of the structure. This first robot follows a path on the floor using ultra red sensors on the front. The nozzle of the concrete extruder is fixed on the side of the robot. With a mechanism the second robot clamps on the layers printed by the first robot and subsequently prints the next layers on top of the first layers. This second robot, referred to as the "Clamp Robot" can print under a small angle, resulting in slightly curved outer surfaces. The third and last robot, the "Vacuum Robot", can print perpendicular to the already printed layers. The robot contains a suction cup at the bottom to stick to the structure. This printer can print on the surface of the wall following a free-form curve.

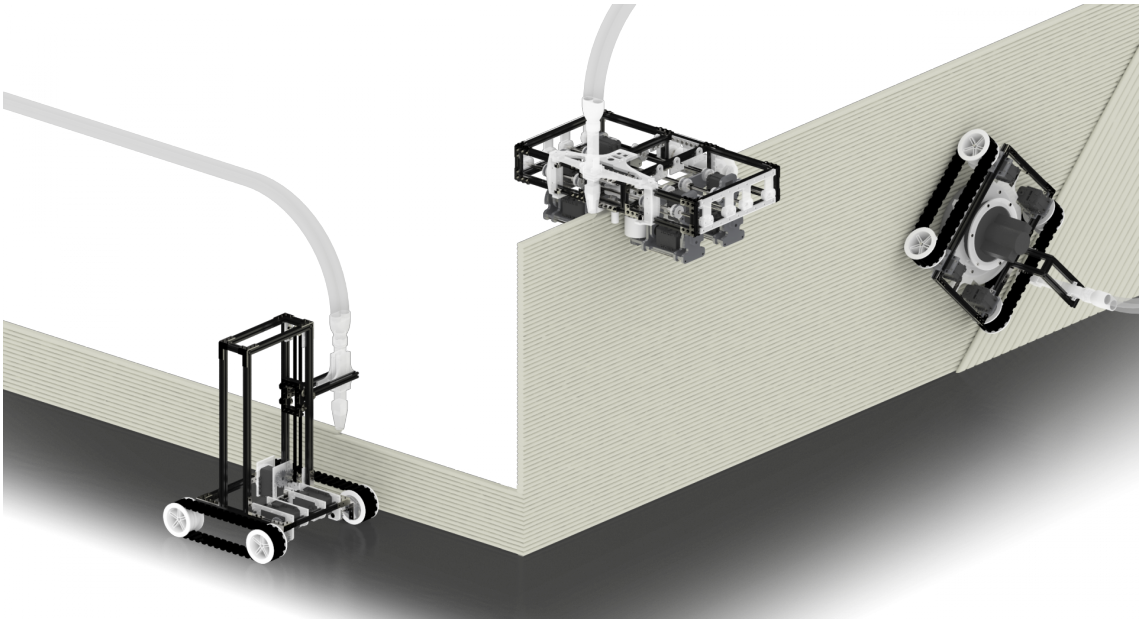


Figure 2.15: Foundation, Clamp and Vacuum Robot (robots.iaac.net 2014)

Another project of the IAAC, in order to bring additive manufacturing to the architectural scale, is the manufacturing of a 3D printed concrete bridge. The created bridge has

a length of 12 meters and a width of 1.75 meters and was printed with micro-reinforced concrete. The form was created using parametric design. In the design the structural performance was maximised and the amount of waste was minimised.

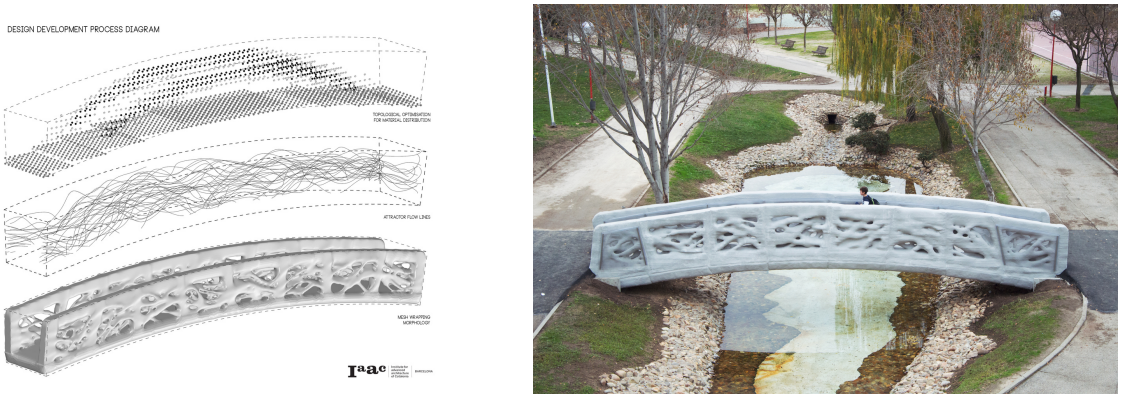


Figure 2.16: 3D Printed Concrete Bridge (Jokić and Novikov 2017)

2.2.10 MIT

The Mediated Matter Group, directed by architect, designer and assistant professor of media arts and sciences at the MIT Media Lab, Neri Oxman, focuses on Nature-inspired Designs and Design-inspired Nature (Oxman 2012). One of their projects is focussing on the density of concrete. The comparison is made between a natural structure, a human bone, and a structural element. The density of bones naturally varies from place to place. Where needed the density is higher. The Mediated Matter Group tried to apply the same approach to structural elements. By controlling the density spatially, by means of 3D printing of different density gradients, materials can be used more efficiently and thus it is possible to create lighter and stronger structures (Oxman et al. 2014).

Figure 2.17: Concrete with varying densities (Oxman et al. 2014)

The Mediated Matter Group does not only create biological digital designs, however focusses on fabrication as well. An example is the Print-in-Place technology. In this process a form-work made of fast-curing polyurethane foam is robotically 3D printed. After casting the concrete the foam is not removed, however it functions as a good thermal insulation material.

Another interesting MIT project is the development of a solar powered robotic system that can print the basic structure of a building, the Digital Construction Platform. The system consists of a vehicle with a large, industrial robotic arm attached to it. On the

end of this arm a smaller robot is mounted which is more precise. Using this system it is possible to print structures in a much larger manufacturing space than other 3D printers. In most other cases this space is limited by the size of the printer.

Figure 2.18: Digital Construction Platform (Oxman 2016)

An advantage of the smaller robot arm at the end is that it can be equipped with different nozzles. For example, nozzles used for pouring concrete, as well as for spraying (insulation) material. As a proof of concept, the researchers at MIT printed the basic structure of the walls of a dome with a height of about 3.7 meters and a width of about 15 meters. The prototype was created in less than 14 hours.

2.2.11 Apis Cor

The last interesting project that is discussed in this chapter is the work of a Russian company called Apis Cor (Sakin and Kiroglu 2017). This company printed a concrete house in Moscow with an area of just under 40 m^2. The techniques are similar to the previously discussed extrusion printers. However, the special aspects of this project were the costs and the construction time. The costs of this house were only 10,000 dollars and it was built in under 24 hours. The structure was printed on site with only a mobile concrete 3D printer. This demonstrates the huge potential of 3D printing of concrete.

Figure 2.19: Printed house by Apis Cor (Sakin and Kiroglu 2017)

3

Structural and topology optimisation

Traditionally structural design is focussing on creating structures which are safe in terms of strength, stiffness and stability. In structural mechanics books from amongst others Carpinteri (2013) and Al Nageim (2003) guidelines can be found on how to ensure this safety. Over the years many structures were over-dimensioned to ensure the safety or because proper manufacturing of different shapes was too difficult or too expensive. Now that scientists and engineers are all aware of the environmental impact of the building industry and now that new manufacturing techniques are being developed, engineers should not only consider strength, stiffness and stability anymore. They should aim to meet the structural regulations with as little material as possible. By changing design variables it is possible to minimise the amount of material used to manufacture a structure. This process is a form of optimisation.

In this chapter a brief introduction is given in the field of optimisation in general and in current structural optimisation techniques. This part of the research is concluded with a more in depth explanation of several topology optimisation methods, especially focussing on the SIMP-method (Bendsøe and Sigmund 2003).

3.1 Basics of optimisation in general

Optimisation can be defined as the process of searching for the minimum (or maximum) value of a set of criteria, defined by an object function, within a given set of boundaries, often defined by parameters or variables. In the context of a structure subject to multiple loads and support conditions the optimal shape is that which best satisfies the constraints, with the degree of satisfaction not necessarily the same for all the constraints (Coenders 2008).

The general form of an optimisation problem is:

$$\begin{array}{lll} \textit{minimise} & F(x); & x \in R^n \\ \textit{such that} & g_i \leq 0; & i = 1, ..., p \\ & h_j = 0; & j = 1, ..., q \end{array} \tag{3.1}$$

$$\textit{where} \qquad F(x) = \begin{bmatrix} f_1(x) \\ ... \\ f_n(x) \end{bmatrix}$$

where $F(x)$ is the objective function, g_i are the inequality constraint functions, h_j are the equality restriction functions, $f_i(x)$ is the i-th object function and x is a vector of design variables or parameters. These parameters can be bounded by box constraints, for instance $x_{min} \leq x_i \leq x_{max}$. This is an example of a "variable bound problem". The so called "search space" is the space spanned by these variables. The "feasible domain" is the part of the search space where all the constraints are satisfied as well.

Optimisation problems can be solved using different methods. Some of these methods are *approximation* methods, others are *optimisation* methods. Approximation methods try to 'learn' the objective function and find an optimum of that obtained function. In optimisation methods the algorithm tries to optimise the objective function by searching in the search space spanned by the possible variables, boundaries and constraints.

In optimisation methods *discrete* and *continuous* design variables are possible. Numerical methods often use a discrete definition of the variables, hence discrete optimisation. When continuous objective, restriction and other functions are assumed the methods are mostly analytical and called continuous.

An important challenge of optimisation methods is the problem of *local optima* of the objective function. Most functions have multiple local optima, in which the derivative is zero. These positions are tops or bottoms, however not necessarily the highest top or lowest bottom. Many methods get stuck in local optima and will therefore never find the real optimum or *global optimum*.

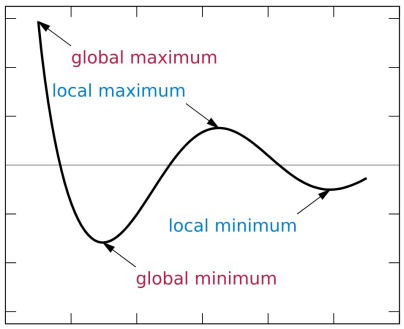


Figure 3.1: Local and global optima

Multi-objective or *multi-criteria optimisation* is possible as well. This class of optimisations is characterised by multiple objective functions. The problem is how to define such a problem that these objectives can be optimised. An example of this class is weighted sum-

mation of the objective functions. The outcome of the optimisation is strongly depending on the chosen preferences of the designer.

3.2 Structural optimisation

Structural optimisation deals with the optimisation of structures. Often mathematical problems and methods can be used to describe and solve structural engineering problems. Different structural optimisation methods have been developed over the years. Structural optimisation is basically the application of optimisation techniques on structures.

Aim of structural optimisation is to minimise or maximise a certain physical property. The property can be load dependent, like the stress in or the deflection of a structure. The optimised property can also be load independent, like the volume or the weight of the structure.

"Structural optimisation seeks to achieve the best performance for a structure while satisfying various constraints such as a given amount of material." (Huang and Xie 2010) Below a short overview is listed of the most important methods.

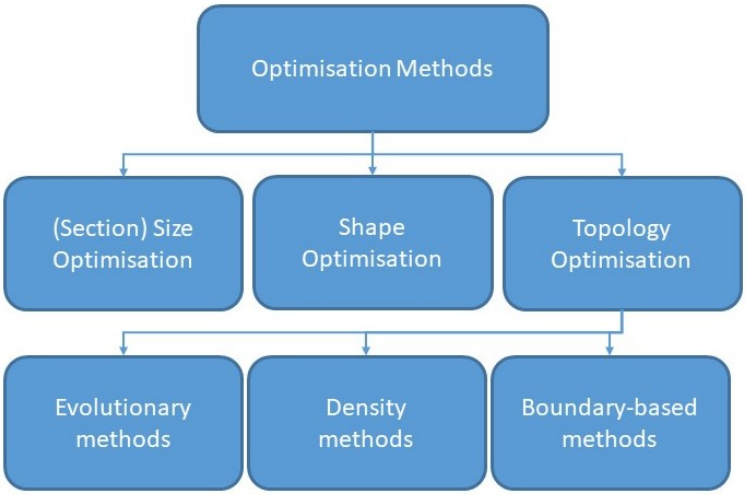


Figure 3.2: Structural optimisation scheme

3.2.1 (Section) Size optimisation

To optimise a certain structure the material in a section is one of the most significant variables. For example, concrete is very effective in compression. Steel on the other hand is preferred for structural parts loaded in tension. Combinations of materials are possible as well; a variable can be the distribution of the reinforcement bars or the layer sequence in composite materials.

Size optimisation is the investigation of the best sizes for a cross-section (Christensen and Klarbring 2008). Nowadays a structural system and the matching cross-sectional sizes are mostly chosen following rules of thumb. Afterwards checks are performed to ensure

the safety of the structure. If the structure is over-dimensioned the sizes of cross-sections are reduced to obtain an improved design. By using software different properties of the cross-section (e.g. width or height) can also be varied to find a structure with a minimum weight, while fulfilling constraints like the maximum stress or displacement at certain points.

3.2.2 Shape optimisation

A different way of optimising a structure in order to reduce material is shape optimisation (Christensen and Klarbring 2008). In shape optimisation the overall shape of the structure, including possible voids, is considered and improved to obtain a structure with a uniform stress distribution. By altering the shape of the structure stress concentrations are minimised or even avoided. The topology of the structure remains constant in this form of optimisation, however the geometry and consequently the load carrying behaviour change completely.

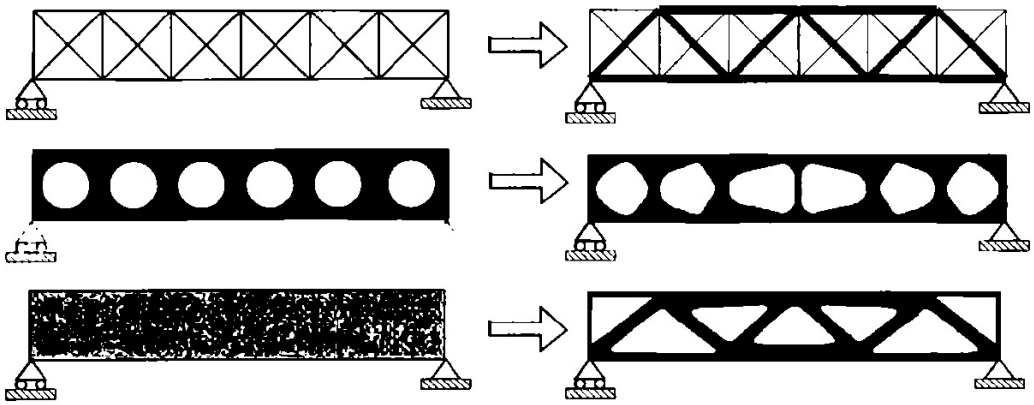


Figure 3.3: Size, shape and topology optimisation (Bendsøe and Sigmund 2003)

3.2.3 Topology optimisation

Neither the shape, nor the size are predefined in topology optimisation (Christensen and Klarbring 2008). In this optimisation method only a design space, boundary conditions and the acting external loads are predefined. The optimisation distributes material in the design space on the necessary positions in the most efficient way. It is technically challenging, however economically rewarding. Rather than limiting the changes to the sizes of structural components, topology optimisation provides much more freedom and allows the designer to create totally novel and highly efficient conceptual designs (Huang and Xie 2010).

3.3 Topology optimisation methods

A topology optimisation problem has the objective to find an optimal distribution of material in a prescribed design domain. Most topology optimisation methods divide that

design domain into a large amount of elements. This process is called the discretisation of the design space. In the optimisation these elements are steered towards material or no material. The final result is therefore a layout consisting of solids and voids. The optimisation problem is a binary problem.

In most cases the compliance, the work done by external forces, is minimised for a certain volume or mass fraction of the initial design domain. In other cases the volume or mass of a structure, subjected to stiffness or strength constraints, is minimised. Many researchers looked into these problems and developed methods to determine the layout of optimised structures.

The main topology optimisation methods are grouped in three sorts of methods: evolutionary methods, density methods and boundary-based methods.

3.3.1 Evolutionary methods

In evolutionary methods the binary problem is solved by generating an initial design. Then a certain value is evaluated for every element, for example the stress in an element. After this evaluation some elements are removed from, or added to the structure. This process is then repeated to obtain an optimal design. Different evolutionary methods are discussed below.

3.3.1.1 ESO method

The ESO method or Evolutionary Structural Optimisation method was first proposed by Xie and Steven (1993) and has since been continuously developed to solve a wide range of topology optimisation problems as mentioned by Xie and Steven (1997).

ESO is based on the simple concept of gradually removing inefficient material from a structure. Through this process, the resulting structure will evolve towards an optimal shape and topology. Theoretically, it is not guaranteed that such an evolutionary process would always produce the best solution, because it is possible that elements are removed in early iterations which are required in a later stage of the optimisation to be part of the optimal design. However, the ESO technique provides a useful tool for engineers and architects who are interested in exploring structurally efficient forms and shapes during the conceptual design stage of a project (Huang and Xie 2010).

3.3.1.1.1 ESO based on stress levels

In this method (Huang and Xie 2010) a design space is subdivided into small elements. By conducting a finite element analysis the stress level in each element is determined. If the stress level in multiple elements is low, the material is not used in an efficient manner. In the ideal situation the stress level in all the elements is around the same level. This level should not be too small, to avoid unnecessary material, however it should not exceed the strength of the used material either. The ESO method compares, for example, the von Mises stress level in a particular element σ_e^{vm} with the maximum allowed von Mises stress of the structure σ_{max}^{vm}. After each finite element analysis or FEA, elements which

satisfy the following condition are removed from the design space.

$$\frac{\sigma_e^{vm}}{\sigma_{max}^{vm}} < RR_i \tag{3.2}$$

where RR_i is the rejection ratio. This cycle of analysis and removal of elements is repeated until no more elements satisfy the condition: a steady state is reached. Now an evolutionary rate, ER, is added to the RR.

$$RR_{i+1} = RR_i + ER \tag{3.3}$$

The evolutionary process continues until there are no elements in the structure containing a stress level below a desired level. A desired (local) optimum is reached.

3.3.1.1.2 ESO for stiffness or displacement optimisation

A structure can be optimised for stiffness as well. In the ESO method commonly the mean compliance C, the inverse of the overall stiffness of a structure, is considered (Huang and Xie 2010). The optimisation problem can be stated as:

$$\begin{aligned} &minimise \quad C; \\ &s.t. \qquad\quad V^* - \sum_{i=1}^{N} V_i x_i = 0; \\ &\qquad\quad\; x_i = 0 \quad or \quad x_i = 1 \end{aligned} \tag{3.4}$$

Where V_i is the volume of an individual element and V^* the prescribed total structural volume. N is the total number of elements in the system. The binary design variable x_i indicates whether an element is absent (0) or present (1).

The mean compliance can be defined by the total strain energy of the structure or the external work done by applied loads as:

$$C = \frac{1}{2} f^T u \tag{3.5}$$

where f is the external force vector and u is the displacement vector. In FEA, the static equilibrium equation of a structure is expressed as:

$$Ku = f \tag{3.6}$$

where K is the global stiffness matrix. When the ith element is removed from the structure, the stiffness matrix will change by:

$$\Delta K = K^* - K = -K_i \tag{3.7}$$

where K^* is the stiffness matrix of the resulting structure after the element is removed and K_i is the stiffness matrix of the ith element. It is assumed that the removal of the element has no effect on the applied load f. By varying both sides of Equation (3.6) the change of the displacement vector is obtained as:

$$\Delta u = -K^{-1} \Delta K u \tag{3.8}$$

From Equations (3.5) and (3.8) we obtain:

$$\Delta C = \frac{1}{2} f^T \Delta u = -\frac{1}{2} f^T K^{-1} \Delta K u = \frac{1}{2} u_i^T K_i u_i \tag{3.9}$$

where u_i is the displacement vector of the ith element. The so called sensitivity number for the mean compliance can thus be defined as:

$$\alpha_i^e = \frac{1}{2} u_i^T K_i u_i \tag{3.10}$$

The above equation implies that the increase in the mean compliance as a result of the removal of an element is equal to its strain energy. To minimise the mean compliance, which is equivalent to maximising the stiffness, through the removal of elements, it is clear that the most effective way to do that is to eliminate the elements which have the lowest values of α_i so that the increase in C will be minimal.

The number of elements to be removed is determined by the element removal ratio (ERR) which is defined as the ratio of the number of elements removed at each iteration to the total number of elements in the initial or the current FEA model (Huang and Xie 2010).

Similar to the ESO based on stress levels this method starts with a division of the design space using a fine mesh of finite elements. Then the FEA is carried out for the structure. The sensitivity numbers are calculated using Equation (3.10) and subsequently a number of elements, with the lowest sensitivity numbers, according to a predefined ERR, are removed. The FEA and the removal of the elements is repeated until the mean compliance, or the maximum displacement, of the structure reaches a prescribed limit.

An important advantage of the ESO method is its *simplicity*. It can be easily understood by the users and easily linked to existing finite element analysis software. The algorithm applies to both 2D and 3D problems equally. Another positive aspect of the method is the *speed*. The elements that are removed in an iteration can be ignored in further steps. This will lead to a smaller number of equations and thus a significant reduction in computation time every iteration. The removal of elements in an early stage can also be a disadvantage. As stated before, some elements might seem unnecessary in early iterations, however they might be a part of the optimal design in the end. The ESO method is not able to recover elements. This is why the ESO method will improve the design in most cases, however it may not result in the absolute optimum.

The ESO method was a concept for further research. Numerical problems in topology optimisation, such as existence of solution, checker-board, mesh-dependency and local optimum, etc. were neglected. To overcome the deficiencies of the ESO method an improved algorithm has been developed: the BESO method.

3.3.1.2 BESO method

The BESO method or bi-directional evolutionary structural optimisation method is an extension of the ESO method and now allows material to be removed and added simultaneously. The first research in the BESO method was conducted by Yang et al. (1999) for stiffness optimisation. In their study, the sensitivity numbers of the void elements are estimated through a linear extrapolation of the displacement field after the finite element

analysis. Then, the solid elements with the lowest sensitivity numbers are removed from the structure, and the void elements with the highest sensitivity numbers are changed into solid elements. The numbers of removed and added elements in each iteration are determined by two unrelated parameters: the rejection ratio (RR) and the inclusion ratio (IR) respectively (Huang and Xie 2010).

Disadvantage of this method is the fact that the ratios have to be chosen carefully to actually reach an optimal solution (Rozvany 2009). Another problem of the early versions of BESO is that the computational efficiency is quite low because of the large number of iterations usually involved, compared to the previously described ESO method.

3.3.1.2.1 BESO method based on stress levels

The concept of BESO is applied on "full stress designs" by using the von Mises criterion Querin et al. (2000). The elements with the highest von Mises stress are switched on (solid) and the elements with a stress below a certain value are removed (void). The number of elements that will be removed or added in a next iteration is depending on the rejection or inclusion ratio again.

3.3.1.2.2 BESO method for stiffness optimisation

Huang and Xie (2010) presented a BESO algorithm for stiffness optimisation which addresses many issues related to topology optimisation of continuum structures such as a proper statement of the optimisation problem, checker-board pattern, mesh-dependency and convergence of solution.

Topology optimisation is often aimed at searching for the stiffest structure given a certain volume. The optimisation problem is again stated as:

$$\begin{aligned} &minimise \quad C = \frac{1}{2} f^T u; \\ &s.t. \qquad\quad V^* - \sum_{i=1}^{N} V_i x_i = 0; \\ &\qquad\qquad x_i = 0 \quad or \quad x_i = 1 \end{aligned} \tag{3.11}$$

where f and u are the applied external forces and displacement vectors respectively and C is the mean compliance. V_i is the volume of an individual element and V^* the prescribed total structural volume. N is the total number of elements in the system. The binary design variable x_i indicates the absence (0) or presence (1) of an element.

When an element is removed from the structure, the mean compliance changes. This change is called the sensitivity number in the BESO method:

$$\alpha_i^e = \Delta C_i = \frac{1}{2} u_i^T K_i u_i \tag{3.12}$$

where u_i is the nodal displacement vector of the ith element, K_i is the elemental stiffness matrix.

By deleting elements based on the original sensitivity numbers a checker-board pattern can originate. Structures with a pattern like this cannot be easily manufactured.

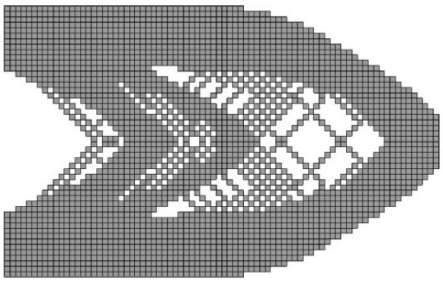


Figure 3.4: Typical checker-board pattern in the ESO method (Bendsøe and Sigmund 2003)

To avoid this pattern a smoothing algorithm is created. A different problem that can occur is called the mesh-dependency problem. This term refers to the problem of obtaining different topologies when using different finite element meshes. To avoid both mentioned problems a filter scheme for the BESO method is developed (Bendsøe and Sigmund 2003).

Before applying this filter scheme, nodal sensitivity numbers without any physical meaning on their own are defined by averaging the elemental sensitivity numbers

$$\alpha_j^n = \sum_{i=1}^{M} w_i \alpha_i^e \tag{3.13}$$

where M represents the total number of elements connected to the jth node, w_i is the weight factor of the ith element and $\sum_{i=1}^{M} w_i = 1$.

$$w_i = \frac{1}{M-1}\left(1 - \frac{r_{ij}}{\sum_{i=1}^{M} r_{ij}}\right) \tag{3.14}$$

where r_{ij} is the distance between the centre of the ith element and the jth node. So the elemental sensitivity number has a larger effect on the nodal sensitivity number when it is closer to the node.

The next step is to calculate the improved sensitivity number of the ith element

$$\alpha_i = \frac{\sum_{j=1}^{K} w(r_{ij})\alpha_j^n}{\sum_{j=1}^{K} w(r_{ij})} \tag{3.15}$$

where K is the total number of nodes in the sub-domain Ω_i and $w(r_{ij})$ is the linear weight factor defined as

$$w(r_{ij}) = r_{min} - r_{ij} \quad (j = 1, 2, ..., K) \tag{3.16}$$

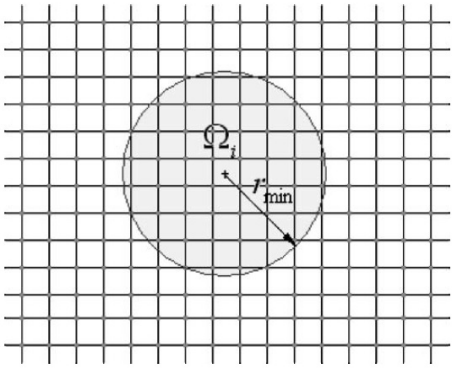


Figure 3.5: The nodes in the circular domain Ω_i are used in the filter scheme for the ith element (Huang and Xie 2010)

In some situations the objective function may not be convergent. For that reason a simple averaging scheme is applied to stabilize the evolutionary process. Huang and Xie (2007) found that averaging the sensitivity number with its historical information is an effective way to solve this problem.

$$\alpha_i = \frac{\alpha_i^k + \alpha_i^{k-1}}{2} \tag{3.17}$$

where k is the current iteration number. Now $\alpha_i^k = \alpha_i$ and can be used for the next iteration. In that way the updated sensitivity number includes the whole history of the sensitivity information in the previous iterations.

Before every new iteration a target volume for the next step V_{k+1} needs to be calculated. The evolution of the volume can be described as

$$V_{k+1} = V_k(1 \pm ER) \quad (k = 1, 2, 3, ...) \tag{3.18}$$

where ER is the evolutionary volume ratio. When the volume reaches its constraint or $V_{k+1} = V^*$ the volume will not change any more for the remaining iterations.

Now the sensitivity numbers of all the elements in the design space are determined according to the formulas described. The elements with a sensitivity number lower than α_{del}^{th} will be removed and void elements with a sensitivity number higher than α_{add}^{th} will be added. α_{del}^{th} is always smaller than or equal to α_{add}^{th} The exact values of α_{del}^{th} and α_{add}^{th} can be determined following the next three steps (Huang and Xie 2010).

1. Start with $\alpha_{add}^{th} = \alpha_{del}^{th} = \alpha_{th}$. In this way α_{th} can be determined by V_{k+1}. For example, there are 1000 elements in the design domain and $\alpha_1 > \alpha_2 ... > \alpha_{1000}$ and if V_{k+1} corresponds to a design with 725 solid elements then $\alpha_{th} = \alpha_{725}$.

2. Determine AR, the volume additional ratio, which is the number of added elements divided by the total number of elements in the design domain. If AR is larger than the maximum volume ratio AR_{max}, α_{del}^{th} and α_{add}^{th} need to be redetermined in the next step. If that ratio is smaller than AR_{max}, step 3 is not needed anymore.

3. Calculate α_{add}^{th} by sorting the sensitivity number of void elements. The number of elements that will become solid equals AR_{max} times the total amount of elements

in the design domain. The sensitivity number of the element ranked just below the last added element is the new α_{add}^{th}. The value of α_{del}^{th} should be chosen in a way that the removed volume equals $(V_k - V_{k+1} + V_{added\,elements})$.

The iterations consisting of the finite elements analysis and the removal and addition of elements will continue until both the volume constraint V^* as well as the convergence criterion (defined in terms of change in the objective function) are satisfied.

$$error = \frac{\left| \sum_{i=1}^{N} C_{k-i+1} - \sum_{i=1}^{N} C_{k-N-i+1} \right|}{\sum_{i=1}^{N} C_{k-i+1}} \leq \tau \tag{3.19}$$

where k is the current iteration number, τ is the allowable convergence tolerance and N is an integer that determines the length of the last iterations considered. $N = 5$ means that the change should be small enough over the last 10 iterations (Huang and Xie 2007).

An overview of the BESO method is visualised in a flowchart in Figure 3.6.

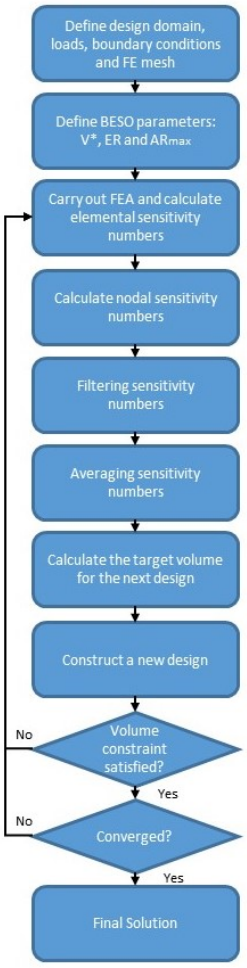


Figure 3.6: Flowchart of the BESO method by Huang and Xie (2010)

3.3.2 Density methods

In density methods the binary conditions are relaxed. The material densities can now have intermediate values between 0 and 1. The resulting structures can thus contain elements with intermediate densities. In some methods, for example the homogenisation method, these elements are part of the final structures, while in other methods a penalisation procedure is applied to avoid these intermediate densities. The latter is the case in the SIMP method.

3.3.2.1 Homogenisation method

The homogenisation method (Bendsøe and Kikuchi 1988) starts with the discretisation of the design space. All the elements are given a density factor between 0 (void element) and 1 (solid element) corresponding to a certain micro-structure. The homogenisation method creates a continuum problem, which means that not only values of 0 or 1 are possible, but intermediate values as well. In practice this means that porous elements or micro-perforated composites can be part of the optimal structure. The method can be described as micro-structure sizing.

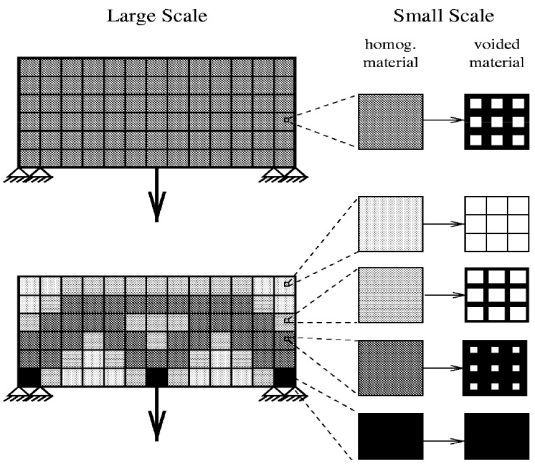


Figure 3.7: Homogenisation method (Belblidia and Bulman 2002)

In this method a material model with micro-scale voids is introduced and the optimisation problem is defined by seeking the optimal porosity of such a porous medium using one of the optimality criteria. The porosity is varied throughout the structure to obtain a more optimal structure. A disadvantage of the method is the fact that it often produces designs with infinitesimal pores in the material that make the structure difficult or non manufacturable.

3.3.2.2 SIMP method

The SIMP, or Solid Isotropic Material with Penalisation method is the most commonly used approach to solve the topology optimisation problem. The objective of the minimum compliance problem is to minimise the deformation of the structure subjected to

certain loads and boundary conditions. A global measure for the deformation is the total compliance (c) of the structure, defined by:

$$c(x) = f^T u(x) \tag{3.20}$$

Where f is the vector of external nodal forces and $u(x)$ is the vector containing the nodal displacements. The compliance should be minimised for a certain prescribed volume (V^*). Now the problem can be presented as:

$$\begin{aligned} &minimise && c = f^T u \\ &s.t. && V^* - \sum_{i=1}^{N} V_i x_i = 0 \\ & && 0 < x_{min} \leq x_i \leq 1 \end{aligned} \tag{3.21}$$

An important difference with the evolutionary methods is that the variable x_i can now vary between a very small value x_{min} and 1. In the ESO and BESO method x_i was either 0 or 1. To steer solutions to void or solid designs the SIMP method uses a material interpolation scheme with penalisation. In the standard SIMP method the Young's modulus of intermediate elements is interpolated as a function of the element density using:

$$E(x_i) = E_0 x_i^p \tag{3.22}$$

in which E_0 is the Young's modulus of the solid material and p is the penalty exponent. To obtain void or solid designs normally $p \geq 3$ is used in the SIMP method. Important to notice is that the elements are not removed, however the virtual density and the stiffness of the void-like elements is changed into a very small value.

The global stiffness matrix can be expressed by:

$$K = \sum_i x_i^p K_i^0 \tag{3.23}$$

where K_i^0 is the elemental stiffness matrix of a solid element.

Similar to the BESO method a sensitivity number is calculated for the elements in the design space (Huang and Xie 2010).

$$\frac{\delta c}{\delta x_i} = -p x_i^{p-1} u_i^T K_i^0 u_i \tag{3.24}$$

The problem can be solved in different ways. One sort of method is the Optimality Criteria (OC) method. A standard OC updating scheme for the design variables by Bendsøe and Sigmund (2003) can be formulated as:

$$x_i^{k+1} = \begin{cases} max(x_{min}, x_i^k - m) & if\ x_i^k B_i^\eta \leq max(x_{min}, x_i^k - m) \\ min(1, x_i^k + m) & if\ min(1, x_i^k + m) \leq x_i^k B_i^\eta \\ x_i^k B_i^\eta & otherwise \end{cases} \tag{3.25}$$

where x_i^k is the value of the design variable at iteration k, m is the positive move-limit, η is a numerical damping coefficient (typically equal to 0.5) and B_i is found from the optimality condition as

$$B_i = \lambda^{-1} p x_i^{p-1} u_i^T K_i^0 u_i \tag{3.26}$$

where λ is a Lagrangian multiplier that can be determined using a bisection method or a Newton method (Bendsøe and Sigmund 2003).

To ensure that the optimal design is mesh-independent and checker-board-free the following sensitivity filter scheme is introduced by Sigmund and Petersson (1998).

$$\frac{\delta c}{\delta x_i} = \frac{1}{x_i \sum_{j=1}^{N} H_{ij}} x_j \frac{\delta c}{\delta x_j} \tag{3.27}$$

where N is the total number of elements in the mesh and H_{ij} is the mesh-independent weight factor defined as

$$H_{ij} = r_{min} - r_{ij}, \quad \{i \in N \mid r_{ij} \leq r_{min}\} \tag{3.28}$$

where r_{ij} is the distance between the centres of element i and j. The weight factor H_{ij} is zero outside the circular filter area shown in Figure 3.5.

An overview of the SIMP method is visualised in a flowchart in Figure 3.8. An algorithm using the SIMP method and a more in-depth explanation of the method is provided in the next chapter.

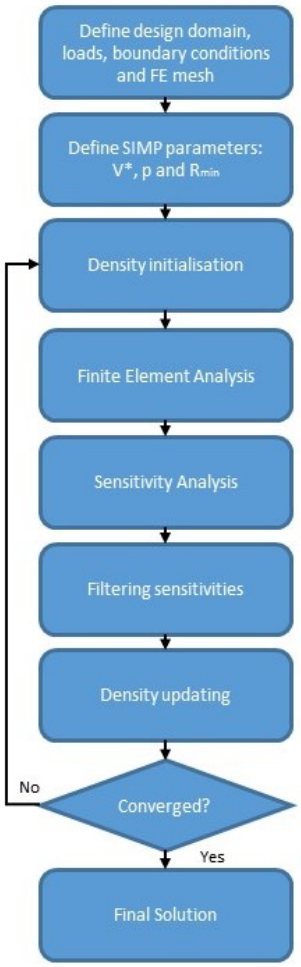


Figure 3.8: Flowchart of the SIMP method

3.3.3 Boundary-based methods

Similar to the evolutionary methods the boundary-based methods solve the binary problem without relaxation of the density. The boundary-based methods are not gradient based, in contrast to the density methods, so the elements are either void or solids and so the boundaries of the final result are clear. In this kind of methods the design variables directly control the exterior and interior boundary shapes of the structure. An example of a boundary-based method with the capability of handling topology changes is the level-set method.

3.3.3.1 Level-set method

In shape optimisation the shape of the structure and thus the voids can be adapted to obtain a better design, however the topology remains constant. The level-set method, a method to track interfaces and shapes, is used for a different kind of shape optimisation. In this method, originally developed by Osher and Sethian (1988), the topology can be changed. Lines and surfaces can merge, split, appear or disappear to form a new topology. The movements of these boundaries are depending on the stress in or the stiffness of the structure. The level-set method is a non-gradient method and therefore only has two values for the densities, 0 (void) and 1 (solid). This results in clear boundaries.

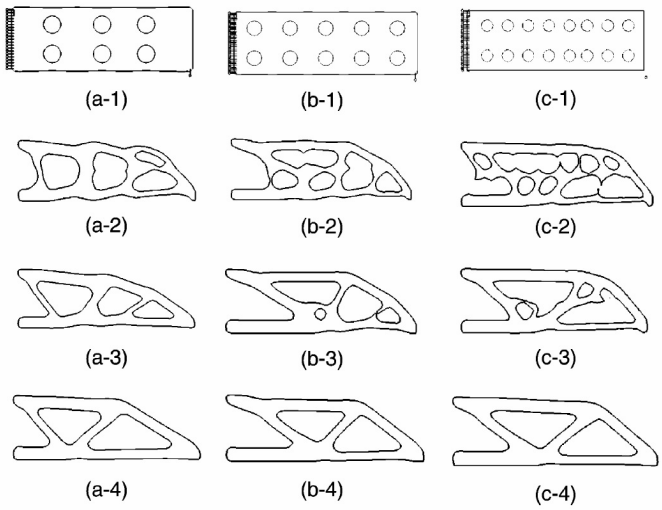


Figure 3.9: Topology optimisation using the level-set method (Wang et al. 2003)

Predefined boundaries can split into pieces to form multiple boundaries, however it is also possible for boundaries to come together to make a single boundary. The method is referred to as "a steepest descent method by combining the shape sensitivity analysis with the Hamilton-Jacobi equation for moving the level-set function, for doing topology design of structures" (Huang and Xie 2010).

Similar to other topology optimisation methods the level-set method aims to find the optimal material distribution Ω_{mat} in a design domain Ω. The domain and the structural boundary can be described by the function $\phi(x)$, the level set function. The aim is to find

the optimal boundary Γ of a distribution Ω_{mat}.

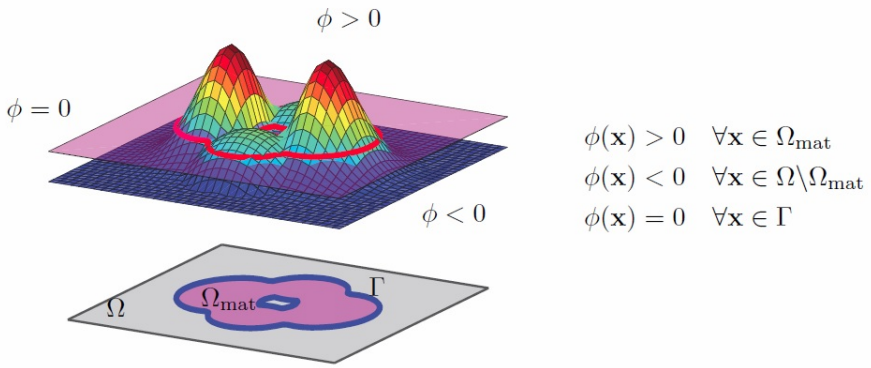


Figure 3.10: Level-set (Verbart et al. 2012)

The level-set method is a promising optimisation method, however it has not reached the stage of regular industrial applications yet (Rozvany 2009).

4

Development of an improved optimisation algorithm

In the previous chapter multiple topology optimisation methods are discussed. The methods described are implemented in multiple software packages which are commercially available. Examples are ANSYS, Autodesk Fusion and Solidworks. Most packages are user-friendly, however they are expensive and the exact codes are not visible and adaptable to individuals. To see the actual code, to change the material model or to add constraints an open-source algorithm is needed.

Two interesting and available algorithms are the 2D TopOpt algorithm by Andreassen et al. (2011) and the Top3D algorithm by Liu and Tovar (2014). Both scripts are written in MATLAB by MathWorks (2010) and thus readable and adaptable for students with a MATLAB license. Unfortunately, a MATLAB licence needs to be purchased to work with the code if your not connected to a university.

The mentioned algorithms are all based on the SIMP method which is briefly described before and more profound by Bendsøe and Sigmund (2003). From the mentioned topology optimisation methods in the previous chapter two are used frequently: the BESO method and the SIMP method. The SIMP method is probably chosen by the researchers because the SIMP method with continuous design variables guarantee that its solution is at least a local optimum (Huang and Xie 2010).

Hard-kill ESO/BESO methods can come up with highly inefficient local optima because these evolutionary methods use the gradient information in the sensitivities to formulate discrete decisions. In that way far less efficient structures can be created by removing the wrong elements.

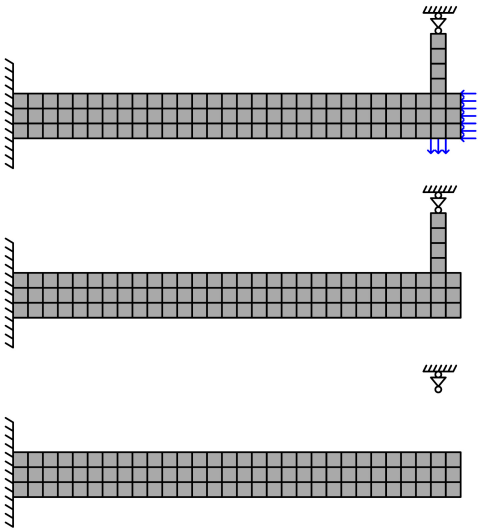


Figure 4.1: A design domain with forces and boundary conditions (top), the initial coarse mesh (middle) and a highly inefficient local optimum (bottom) (Zhou and Rozvany 2001)

As illustrated in Figure 4.1 the initial design can change in a cantilever beam by removing the wrong elements. In that case the compliance will be much higher than with the vertical tie included in the design. This problem can occur when applying the SIMP method as well, however the BESO method will not recover the removed elements in a next iteration. It will add elements in the region with the highest strain energy density, which is at the left-bottom of the cantilever.

In this chapter a modified SIMP method is described more precisely and a work-flow is created for topology optimisation for concrete additive manufacturing using Rhino/-Grasshopper to model the problem and visualise the outcome. The latest Python version (3.6.5) is used to perform the optimisation. First the problem will be defined, then a material model will be created and explained that can be used to model 3D printed concrete. Different filters are explained that resolve numerical difficulties and assure a void and solid outcome. Additionally a filter is explained that takes into account some constraints of the additive manufacturing process. Finally, the FEA is briefly discussed, the sensitivity analysis is explained and the solvers are discussed.

4.1 Definition of the problem

In this section the problem is defined. A designer usually starts with a space in which a structure will arise, the external loads are given by the Eurocodes, depending on the future function of the structure and finally the designer chooses where and how the structure or the structural part will be attached to other elements or to the foundation.

4.1.1 Design domain, boundary conditions and loads

Topology optimisation typically starts with the description of the *design domain*. This design domain is a 2D or 3D space in which the final design will be formed. This space is

divided into small parts, the elements. In this research simple square (2D) or box-shaped (3D) elements with predefined dimensions are stacked together to obtain a design domain.

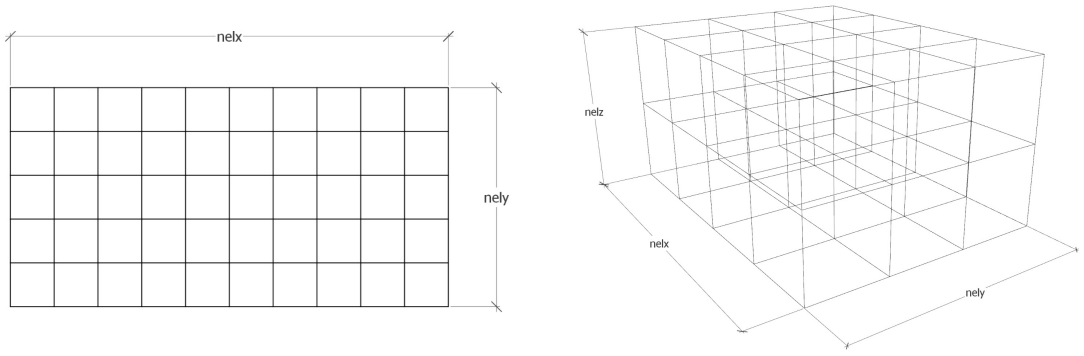


Figure 4.2: 2D and 3D design space

The created 2D elements are so called 4 node quadrilateral elements and the 3D elements are 8 node hexahedrons.

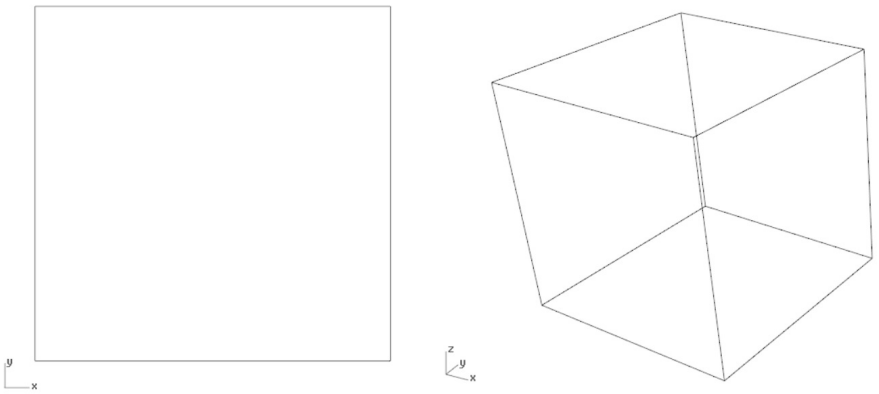


Figure 4.3: Used elements in 2D (4 node quadrilateral element) and 3D (8 node hexahedron)

The corner nodes contain several degrees of freedom. In 2D the nodes can shift in both x- and y-direction. In 3D the points contain an additional degree of freedom in the z-direction. In the algorithm the elements, the nodes and the degrees of freedom are numbered. The elements and nodes are numbered from top-to-bottom, from left-to-right and from back-to-front. The numbers of the degrees of freedom in 3D are obtained by: x-degree of freedom = 3N, y-degree of freedom = 3N+1 and z-degree of freedom = 3N+2. In 2D this is: x-degree of freedom = 2N and y-degree of freedom = 2N+1.

By constraining certain degrees of freedom the boundary conditions can be assigned. It is possible to select any node and to fix it in whatever direction needed.

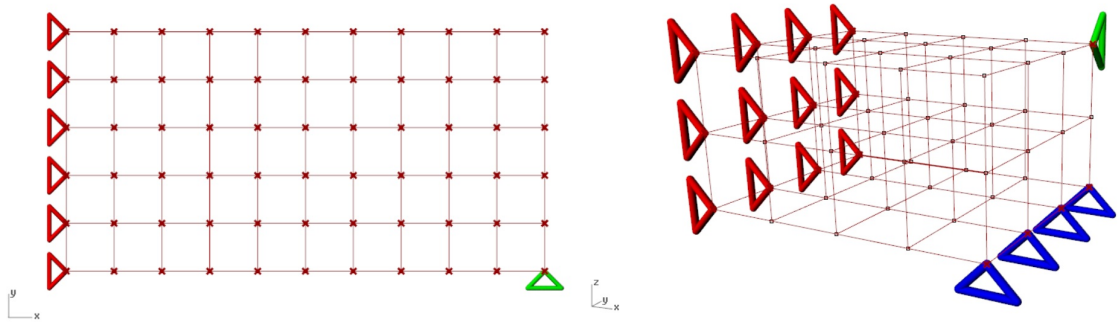


Figure 4.4: Boundary conditions in 2D and 3D

In a similar manner the external forces can be assigned to the created grid. In the algorithm the magnitude of the force and the direction can be described. The external forces are now assigned to the selected nodes. To make sure the resulting structure is safe, the magnitudes that are assigned here are the load values for a certain structural function multiplied by the factors of safety.

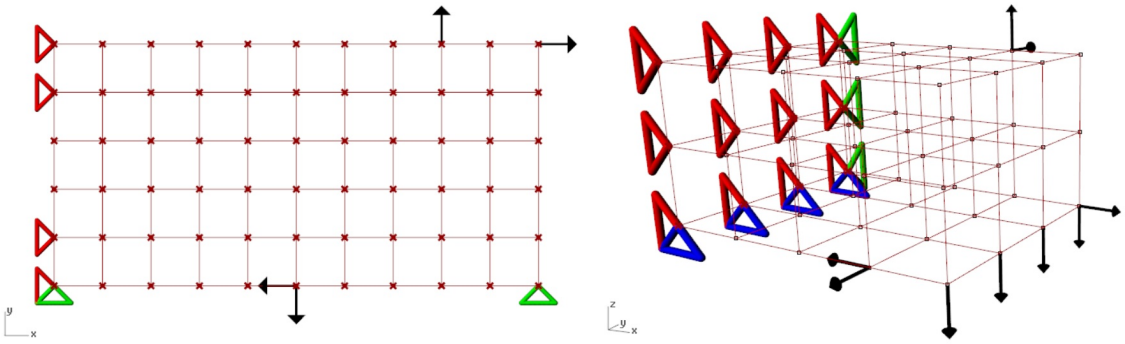


Figure 4.5: Assignment of the external forces in 2D and 3D

It is possible to consider self-weight of the elements in the optimisation as well. In that case the nodal force vector F will contain an additional part:

$$F = F_{ext} + F_{sw} \tag{4.1}$$

where the self-weight part of the force per element is:

$$F_{sw} = V_e \rho_e g \bar{F} = V_e \rho_0 \hat{x} g \bar{F} \tag{4.2}$$

where V_e is the elemental volume, ρ_e the real density of the element, g is the gravity acceleration and ρ_0 is the density of the used material. $\hat{x}$ is the vector with filtered design variables and $\bar{F}$ divides the elemental force in y-direction over the 4 nodes in 2D problems and in z-direction over 8 nodes in 3D problems. A part of F is now depending on the design variable, which is important for the calculation of the sensitivities later on.

4.1.2 Objective functions

Now that the design domain, loads and boundary conditions are known, the *objective* of the optimisation needs to be determined. The objective function is the function that will be minimised. Topology optimisation can be used to find optimal solutions using different objective functions. As mentioned before the standard SIMP method minimises the compliance to obtain a structure which is as stiff as possible for a certain part of the initial volume. In this research an additional objective, the minimisation of the volume or mass, is an option as well.

4.1.2.1 Minimise compliance

A measurement of the overall stiffness of the structure is the compliance c, the inverse of the stiffness. This value needs to be minimised by the algorithm in order to obtain a structure which is as stiff as possible. The objective is described as:

$$minimise \quad c(\hat{x}) = F^T U(\hat{x}) \tag{4.3}$$

where F is the load vector and $U(\tilde{x})$ is the vector with nodal displacements.

4.1.2.2 Minimise volume

Rather than obtaining a maximum stiffness for a certain volume percentage, designers, architects and engineers are more interested in minimising the amount of material necessary without exceeding the limitations of the used material. The objective, the volume fraction, is described as:

$$minimise \quad v(\hat{x}) = \frac{1}{n_e} \sum_{n=1}^{n_e} \hat{x} v_i \tag{4.4}$$

where $v(\tilde{x})$ is the fraction of the initial design domain, n_e the number of design variables, $\hat{x}$ is the vector with filtered design variables and v_i is the volume of an element.

4.1.3 Constraint functions

In optimisation problems the objective is always subjected to a constraint function. Otherwise the optimisation would result in trivial outcomes. For example, in case of compliance minimisation the final result would be a completely solid design space. After all, that would result in the stiffest structure. It is possible to limit the material use, by means of a volume constraint, however it is possible to set up constraints for the strength and stiffness as well.

4.1.3.1 Volume constraint

In standard compliance minimisation usually the volume is limited. Only a certain percentage of the initial design area or volume can be used to create a structure. The volume

constraint is connected to the mass by the well known formula:

$$m = \rho V \tag{4.5}$$

where m is the mass in kg, ρ the density of the used material in kg/m^3 and V the volume in m^3. For that reason a volume constraint can be easily linked to a mass constraint. The volume constraint that is used in this research is:

$$s.t. \quad \sum_{n=1}^{n_e} \hat{x} v_i \leq V_{lim} = v_{lim} n_e \tag{4.6}$$

where $\tilde{x}$ is the vector with design variables, v_{lim} is the prescribed volume limit, v_i is the volume of an element and n_e is the number of considered elements. Rearranging the formula gives the final volume constraint:

$$s.t. \quad g_v = \frac{\sum_{n=1}^{n_e} \hat{x} v_i}{v_{lim} n_e} - 1 \leq 0 \tag{4.7}$$

4.1.3.2 Compliance constraint

A different constraint that can be applied is the compliance constraint. When applying this constraint the overall stiffness of the structure is bounded by a certain limit value for the compliance, c_{lim}. The maximum deformation can be limited by this constraint. The stiffness or compliance constraint is simply described as:

$$s.t. \quad c(\hat{x}) \leq c_{lim} \tag{4.8}$$

where $c(\hat{x})$ is the compliance of the structure and c_{lim} is the maximum allowable value for the compliance. Rearranging these values gives:

$$s.t. \quad g_c = \frac{c(\hat{x})}{c_{lim}} - 1 \leq 0 \tag{4.9}$$

4.1.3.3 Stress constraint

The third type of constraint that can be implemented to the optimisation problem is a stress limit. To obtain a result that contains a certain strength it is possible to limit the maximum stress in the final design:

$$s.t. \quad \sigma(\hat{x}) \leq \sigma_{lim} \tag{4.10}$$

where $\sigma(\hat{x})$ is a certain considered stress in the final structure and σ_{lim} is the strength of the material.

4.1.3.3.1 Introduction in stress-constraint topology optimisation

Many researchers worked on an approach with stress constraints in the last decades and many researchers are working on it at the moment. A straightforward approach is to calculate the stresses at given points in the design space and constrain these values. If the

stress in for example an element is higher than the prescribed limit, more material or a different distribution of the material is needed. For example, Duysinx and Bendsøe (1998), Pereira et al. (2004) and Yang and Chen (1996) applied this so called *local constraint* approach in their works.

A different notation of the stress constraint in Equation 4.10 is:

$$\left(\frac{\sigma}{\sigma_{lim}} - 1\right) \hat{x} \leq 0 \tag{4.11}$$

where σ is the considered stress value and σ_{lim} is the maximum allowable stress. This formula was used as a local stress constraint considered at the centre of the finite elements.

When in this approach the relative density goes towards zero, the so called ***singularity phenomenon*** can occur. This phenomenon was already encountered in truss optimisation, stated in the early works of Sved and Ginos (1968). They found that when the bar area goes to zero, the stress constraints are violated. The considered bar could thus not be removed. A similar problem occurs when design variables (virtual densities) of the considered elements in topology optimisation go to zero. The stresses become really high, while they should actually go to zero, because the elements are removed from the structure. This phenomenon is discussed by many researchers, such as Kirsch (1990), Guo et al. (2001) and Rozvany and Birker (1994). One approach to tackle this problem is the *ϵ-relaxation* suggested by Cheng and Guo (1997). They suggested to relax the stress constraint by an expression. The ϵ starts at a certain positive value and this value of the relaxation is gradually reduced, which leaves the original stress constraint in the end. Duysinx and Bendsøe (1998) used this approach in stress constraint topology optimisation and defined the relaxation as:

$$\left(\frac{\sigma}{\sigma_{lim}} - 1\right) \hat{x} \leq \epsilon \tag{4.12}$$

A different approach to overcome the difficulty of singularity is the *qp-relaxation* (Bruggi 2008). The relaxed stress vector $\tilde{\sigma}$ in this approach is given by:

$$\tilde{\sigma} = \frac{\langle\sigma\rangle}{\hat{x}^q}, \quad where, \quad q \in (0, p) \tag{4.13}$$

In this equation q is the relaxation exponent. The relaxation increases for lower values of q. The considered macroscopic stress $\langle\sigma\rangle$ is calculated using the equation:

$$\langle\sigma\rangle = C_e\epsilon = \hat{x}^p C_0\epsilon \tag{4.14}$$

where C_e is the constitutive or elasticity matrix and ϵ is the strain. Combining these equations gives:

$$\tilde{\sigma} = \hat{x}^{p-q}\sigma_0, \quad where, \quad \sigma_0 = C_0\epsilon \tag{4.15}$$

where σ_0 is the stress vector and C_0 is the constitutive or elasticity matrix for solid elements.

$$\frac{\hat{x}^{p-q}\sigma_0}{\sigma_{lim}} - 1 \leq 0 \tag{4.16}$$

According to Verbart et al. (2011) the design space is highly non-convex and therefore it may be difficult to find an optimal solution. Equation 4.16 holds for every element in the design.

When the design domain is large and consist of many elements, many constraint equations need to be considered. This is a great disadvantage of the local approach. Duysinx and Bendsøe (1998) also concluded that more efficient methods are required to reduce the ***high computational cost*** of of this large-scale optimisation problem.

Duysinx and Sigmund (1998) came up with an idea to bundle all the local stresses in a single *global constraint* to reduce the computational time. The challenge hereby is to find a function σ^G that represents all the local constraints in a proper way. Well known aggregation functions from the literature are the *Kreisselmeier-Steinhauser (KS)* and the *P-norm and P-mean* functions. These functions consider a number of stress evaluation points, n_a. In this research the stress is evaluated in every element, so $n_a = n_e$. An important property of both functions is the fact that they are, in contrast to min/max-operators, differentiable. Gradient information is necessary to solve the optimisation problem.

The Kreisselmeier-Steinhauser (KS) function approximates the maximum stress by giving a lower and and upper bound:

$$\sigma_{KS}^{low} = \frac{1}{P} \ln \left(\frac{1}{n_a} \sum_{i=1}^{n_a} e^{P\tilde{\sigma}} \right) \leq \sigma_{lim} \leq \frac{1}{P} \ln \left(\sum_{i=1}^{n_a} e^{P\tilde{\sigma}} \right) = \sigma_{KS}^{upp} \tag{4.17}$$

In this research the P-norm function is used as the global measure of the stress. This aggregation function is an upper bound on the largest stress in a structure and is defined by:

$$\sigma_{PN} = \left(\sum_{i=1}^{n_a} (\tilde{\sigma})^P \right)^{\frac{1}{P}} \geq \sigma_{lim} \tag{4.18}$$

P-mean is an underestimation of the maximum stress and therefore a lower bound:

$$\sigma_{PM} = \left(\frac{1}{n_a} \sum_{i=1}^{n_a} (\tilde{\sigma})^P \right)^{\frac{1}{P}} \leq \sigma_{lim} \tag{4.19}$$

For both the KS function as well as the P-norm and the P-mean functions it holds that the higher P, the closer the real maximum value is approached. However, the higher this aggregation factor, the higher the non-linearity of the functions. In research papers from Kiyono et al. (2016), Le et al. (2010) and Holmberg et al. (2013) values of P between 6 and 30 are used.

The global stress measure σ_{PN} lacks physical meaning and to actually use it as a stress constraint Le et al. (2010) proposed a *normalised global stress measure*. The global stress measure is scaled by a factor c_{norm}, which is calculated using information from previous iterations.

$$\sigma_{max} \approx c_{norm} \sigma_{PN}, \quad where, \quad c_{norm}^k = \alpha^k \frac{\sigma_{max}^{k-1}}{\sigma_{PN}^{k-1}} + \left(1 - \alpha^k\right) c^{k-1} \tag{4.20}$$

where $0 < \alpha^k \leq 1$ controls the variation between the factor c_{norm}^k and that factor in the previous iteration ($\alpha^k = 1$). k is the iteration number.

Duysinx and Sigmund (1998), Pereira et al. (2004), Fancello Ancello and Pereira (2003) and many others all came up with different versions of global constraint functions. These

functions lower the computational costs, however with the use of global constraints the local stress control is lost, which can lead to final designs that are still not acceptable. To gain more local control and to keep the computational costs low a *clustered* approach is proposed in which several stress evaluation points are grouped. París et al. (2010) came up with the so called block aggregation and Le et al. (2010) defined a regional stress measure. This clustering is not implemented in the research so far.

The constraints that are adopted in the code are listed en explained below.

4.1.3.3.2 Von Mises yield criterion

The most commonly used stress criterion is the von Mises stress. It can be used to constrain the results in a way that the yield stress is not exceeded in any element of the structure.

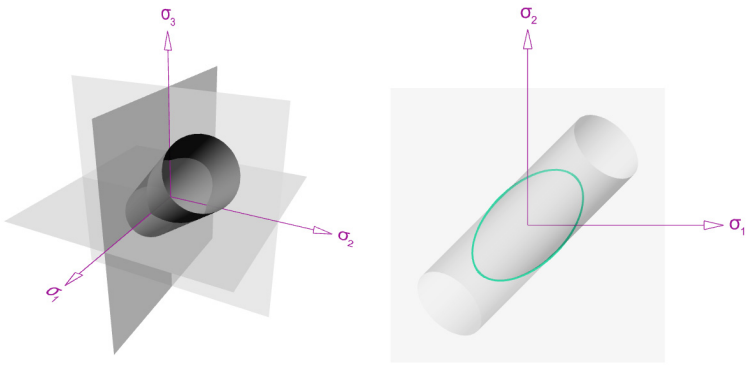


Figure 4.6: 3D and 2D representation of the Von Mises stress criterion

In 2D optimisation problems the von Mises stress can be calculated using:

$$\begin{aligned} \tilde{\sigma}_{vM} &= \sqrt{\tilde{\sigma}_{xx}^2 + \tilde{\sigma}_{yy}^2 + 3\tilde{\sigma}_{xy}^2 - \tilde{\sigma}_{xx}\tilde{\sigma}_{yy}} \\ &= \sqrt{\tilde{\sigma}^T V_{2D} \tilde{\sigma}} \end{aligned} \tag{4.21}$$

where the relaxed stress tensor is defined as:

$$\tilde{\sigma} = \begin{bmatrix} \tilde{\sigma}_{xx} \\ \tilde{\sigma}_{yy} \\ \tilde{\sigma}_{xy} \end{bmatrix} \tag{4.22}$$

and the middle matrix V_{2D} is given by:

$$V_{2D} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \tag{4.23}$$

In 3D problems the stress tensor contains six values and therefore the calculations are

changed to:

$$\begin{aligned}\tilde{\sigma}_{vM} &= \sqrt{\frac{1}{2}[(\tilde{\sigma}_{xx}-\tilde{\sigma}_{yy})^2+(\tilde{\sigma}_{yy}-\tilde{\sigma}_{zz})^2+(\tilde{\sigma}_{zz}-\tilde{\sigma}_{xx})^2]+3(\tilde{\tau}_{xy}^2+\tilde{\tau}_{yz}^2+\tilde{\tau}_{zx}^2)} \\ &= \sqrt{\tilde{\sigma}^T V_{3D}\tilde{\sigma}}\end{aligned} \tag{4.24}$$

where the relaxed stress tensor is defined as:

$$\tilde{\sigma} = \begin{bmatrix}\tilde{\sigma}_{xx}\\ \tilde{\sigma}_{yy}\\ \tilde{\sigma}_{zz}\\ \tilde{\tau}_{yz}\\ \tilde{\tau}_{zx}\\ \tilde{\tau}_{xy}\end{bmatrix} \tag{4.25}$$

and the middle matrix V_{3D} is given by:

$$V_{3D} = \begin{bmatrix}1 & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0\\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0\\ -\frac{1}{2} & -\frac{1}{2} & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 3 & 0 & 0\\ 0 & 0 & 0 & 0 & 3 & 0\\ 0 & 0 & 0 & 0 & 0 & 3\end{bmatrix} \tag{4.26}$$

By using the qp-relaxation approach to avoid singularity and applying the normalised P-norm function to keep the computational time within reasonable limits the first stress constraint is given as:

$$s.t. \quad g_{s,1} = \frac{c_{norm}\sigma_{PN_{vM}}}{\sigma_{vM_{lim}}} - 1 \leq 0 \tag{4.27}$$

where

$$\sigma_{PN_{vM}} = \left(\sum_{i=1}^{n_a}(\tilde{\sigma}_{vM})^P\right)^{\frac{1}{P}} \tag{4.28}$$

4.1.3.3.3 Drucker-Prager yield criterion

In this research the focus is on optimising 3D printed concrete. Because the material is printed in layers and due to the bad performance of concrete in tension the von Mises stress is not a realistic stress criterion for the chosen material and manufacturing procedure. Some researchers, such as Cai (2011) tried to incorporate this unequal behaviour in tension and compression through adapting a non-linear constitutive law for the material. Others like Bruggi and Duysinx (2013) worked on a stress-based approach considering unilateral material.

According to them the Drucker-Prager yield criterion is a smooth strength criterion that is more suited to the used material, than the von Mises criterion. The material considered contains different strengths in tension and in compression, respectively σ_{Lt} and σ_{Lc}. The uni-axial asymmetry ratio s is defined as:

$$s = \frac{\sigma_{Lc}}{\sigma_{Lt}} \tag{4.29}$$

From this formula follows that for a high value of s no-tension material is favoured and for a low value of s no-compression material is preferred. In their research Drucker and Prager (1952) describe that the stress tensor σ is part of the feasible domain if the following equation holds for every element:

$$\sigma^{eq} = \alpha_{d.p.}\sqrt{3J_{2D}} + \beta_{d.p.}J_1 \leq 1 \tag{4.30}$$

where

$$\alpha_{d.p.} = \frac{\sigma_{Lc} + \sigma_{Lt}}{2\sigma_{Lt}\sigma_{Lc}}, \quad and \quad \beta_{d.p.} = \frac{\sigma_{Lc} - \sigma_{Lt}}{2\sigma_{Lt}\sigma_{Lc}} \tag{4.31}$$

where J_1 is the first stress invariant of σ and J_{2D} the second invariant of its deviatoric part (Bruggi and Duysinx 2013). Assuming plane stress conditions the following holds for those values in 2D:

$$\begin{aligned} J_1 &= \sigma_{xx} + \sigma_{yy} \\ 3J_{2D} &= \sigma_{xx}^2 + \sigma_{yy}^2 + 3\sigma_{xy}^2 - \sigma_{xx}\sigma_{yy} \end{aligned} \tag{4.32}$$

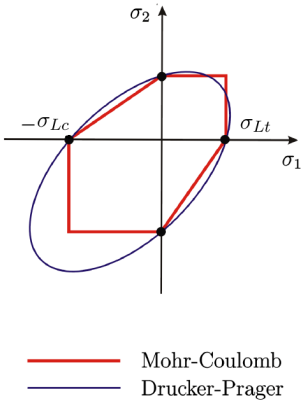


Figure 4.7: 2D representation of the Drucker-Prager stress criterion

In 3D the values are calculated using:

$$\begin{aligned} J_1 &= \sigma_{xx} + \sigma_{yy} + \sigma_{zz} \\ 3J_{2D} &= \frac{1}{2}[(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2] + 3(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2) \end{aligned} \tag{4.33}$$

According to Duysinx et al. (2008) it is possible to rewrite the terms above and the equivalent stress in terms of a "hydrostatic stress matrix" H^0 and a "von Mises stress matrix" M^0.

$$\begin{aligned} J_1 &= \hat{x}^p H^0 U \\ 3J_{2D} &= \hat{x}^{2p} U^T M^0 U \end{aligned} \tag{4.34}$$

the equivalent stress referring to the macroscopic stress of the elements is now described as:

$$\begin{aligned} \langle \sigma^{eq} \rangle &= \hat{x}^p \left(\frac{s+1}{2s}\sqrt{U^T M^0 U} + \frac{s-1}{2s} H^0 U \right) \\ &= \hat{x}^p \sigma^{eq} \end{aligned} \tag{4.35}$$

To use this stress failure criterion as a constraint in the topology optimisation problems, the relaxed stress $\tilde{\sigma}^{eq}$ is determined:

$$\tilde{\sigma}^{eq} = \frac{\langle \sigma^{eq} \rangle}{\hat{x}^q} = \hat{x}^{p-q}\sigma^{eq} \tag{4.36}$$

In this case the P-norm aggregation is used as well for computational efficiency. The second stress constraint, which takes into account the stress in different directions is given as:

$$s.t. \quad g_{s,2} = \frac{\sigma_{PN_{eq}}}{\sigma_{Lt}} - 1 \leq 0 \tag{4.37}$$

where

$$\sigma_{PN_{eq}} = \left(\sum_{i=1}^{n_a} (\tilde{\sigma}_{eq})^P \right)^{\frac{1}{P}} \tag{4.38}$$

4.1.3.3.4 Defined problems

In the previous sections the most important objectives and constraints are defined and explained. However, some combinations of those objectives and constraints are senseless. Below the six most common problems are defined.

Problem A: Compliance minimisation subjected to a volume constraint

$$\begin{aligned} &minimise && c(\hat{x}) = F^T U(\hat{x}) \\ &s.t. && g_v = \frac{\sum_{n=1}^{n_e} \hat{x}}{v_{lim} n_e} - 1 \leq 0 \\ & && 0 < x_{min} \leq x_i \leq 1 \end{aligned} \tag{4.39}$$

Problem B: Volume minimisation subjected to a compliance constraint

$$\begin{aligned} &minimise && v(\hat{x}) = \frac{1}{n_e} \sum_{n=1}^{n_e} \hat{x} \\ &s.t. && c(\hat{x}) \leq c_{lim} \\ & && 0 < x_{min} \leq x_i \leq 1 \end{aligned} \tag{4.40}$$

Problem C: Volume minimisation subjected to a von Mises stress constraint

$$\begin{aligned} &minimise && v(\hat{x}) = \frac{1}{n_e} \sum_{n=1}^{n_e} \hat{x} \\ &s.t. && g_{s,1} = \frac{c_{norm}\sigma_{PN_{vM}}}{\sigma_{vM_{lim}}} - 1 \leq 0 \\ & && 0 < x_{min} \leq x_i \leq 1 \end{aligned} \tag{4.41}$$

Problem D: Volume minimisation subjected to a Drucker-Prager stress constraint

$$\begin{aligned} &minimise && v(\hat{x}) = \frac{1}{n_e} \sum_{n=1}^{n_e} \hat{x} \\ &s.t. && g_{s,2} = \frac{\sigma_{PN_{eq}}}{\sigma_{Lt}} - 1 \leq 0 \\ & && 0 < x_{min} \leq x_i \leq 1 \end{aligned} \tag{4.42}$$

Problem E: Volume minimisation subjected to a compliance and a von Mises stress constraint

$$\begin{aligned} &minimise && v(\hat{x}) = \frac{1}{n_e}\sum_{n=1}^{n_e}\hat{x} \\ &s.t. && c(\hat{x}) \leq c_{lim} \\ & && g_{s,1} = \frac{c_{norm}\sigma_{PN_{vM}}}{\sigma_{vM_{lim}}} - 1 \leq 0 \\ & && 0 < x_{min} \leq x_i \leq 1 \end{aligned} \tag{4.43}$$

Problem F: Volume minimisation subjected to a compliance and a Drucker-Prager stress constraint

$$\begin{aligned} &minimise && v(\hat{x}) = \frac{1}{n_e}\sum_{n=1}^{n_e}\hat{x} \\ &s.t. && c(\hat{x}) \leq c_{lim} \\ & && g_{s,2} = \frac{\sigma_{PN_{eq}}}{\sigma_{Lt}} - 1 \leq 0 \\ & && 0 < x_{min} \leq x_i \leq 1 \end{aligned} \tag{4.44}$$

4.2 Density method and filtering of densities

The SIMP method is a density based method. In this section the method will be explained in more depth and the filters used to obtain manufacturable structures are discussed.

4.2.1 Density based method

As a starting point for the optimisation the previously created elements are all assigned an arbitrary virtual density x of for example 0.5. The binary problem (0 or 1) is relaxed in this method by using a continuous density value between the two values. This method does not make use of different microstructures, in contrast to the homogenisation method, however a *power law interpolation function* between void and solid is used to determine the mechanical properties of the material element. Intermediate densities are penalised by the power law towards a void or a solid, which will lead to a final solution with elements with or without material.

The SIMP method is based on a relation between the virtual elemental density and the element Young's modulus E_i given by:

$$E_i = E_i(x_i) = x_i^p E_0 \tag{4.45}$$

where E_0 is the elastic modulus of the solid material and p is the penalisation power, which is larger than one. However, in this research a modified SIMP approach is used where:

$$E_i = E_i(x_i) = E_{min} + x_i^p(E_0 - E_{min}) \tag{4.46}$$

In both equations the elemental density x_i can vary between 0 and 1. In Equation 4.46 the elastic modulus of the void material, E_{min}, is a little larger than 0 to avoid singularity of the finite element stiffness matrix (Liu and Tovar 2014).

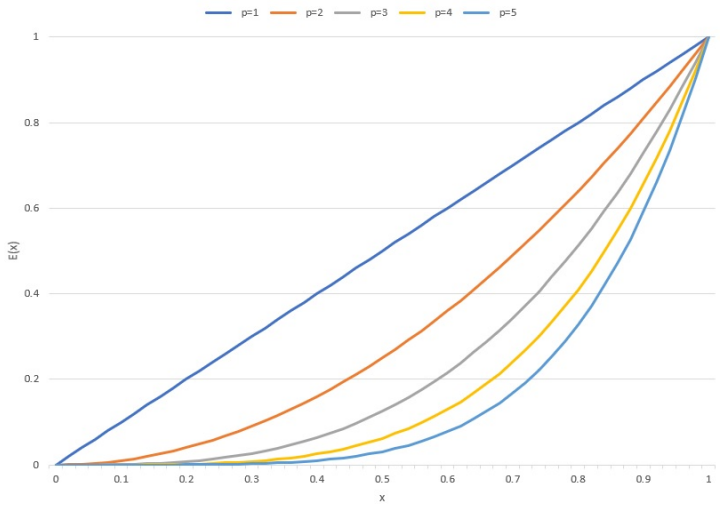


Figure 4.8: SIMP relation between x and $E(x)$

This modified elastic modulus has a number of advantages over the original Young's modulus. One of the advantages is the independency between the minimum value of the material's elastic modulus and the penalisation power (Sigmund 2007).

Despite the use of this modified SIMP method the optimisation can still encounter numerical difficulties. To avoid difficulties like mesh-dependency, checker-board patterns and local minima (Bendsøe and Sigmund 2003) different regularisation or filter techniques have been proposed. A filtered density $\hat{x}_i$ is computed by different regularisation techniques.

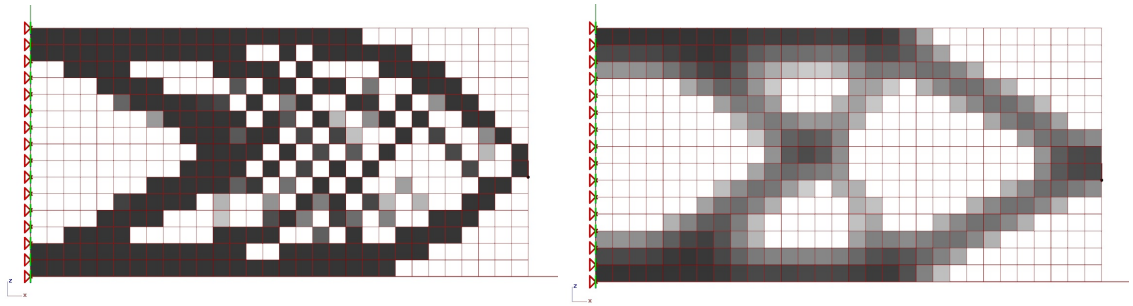


Figure 4.9: Checkerboard patterns can be diminished or eliminated by filtering the virtual densities

After this filtering process the filtered density $\hat{x}_i$ is incorporated in the topology optimisation as:

$$E_i = E_i(\hat{x}_i) = E_{min} + \hat{x}_i^p(E_0 - E_{min}) \tag{4.47}$$

The filters considered in this research are the density filter, the sensitivity filter, the volume preserving Heaviside filter and the robust filter.

4.2.2 Density filter

One method to avoid numerical difficulties is the use of *density filters* (Bruns and Tortorelli 2001). A density filter proposed is defined as:

$$\tilde{x}_i = \frac{\sum_{j \in N_i} H_{ij} v_j x_j}{\sum_{j \in N_i} H_{ij} v_j} \tag{4.48}$$

in which N_i is the neighbourhood of the element i with volume v_i. H_{ij} is the so called weight factor:

$$H_{ij} = R - dist(i,j) \tag{4.49}$$

and

$$N_i = \{j : dist(i,j) \leq R\} \tag{4.50}$$

where $dist(i,j)$ is the distance between the center of element i and the center of element j and R is the filter size defined by the user. The formulas show that elements close to the considered element i contribute more to the filtered density of element i than the the elements further away. Elements outside the neighbourhood do not contribute at all.

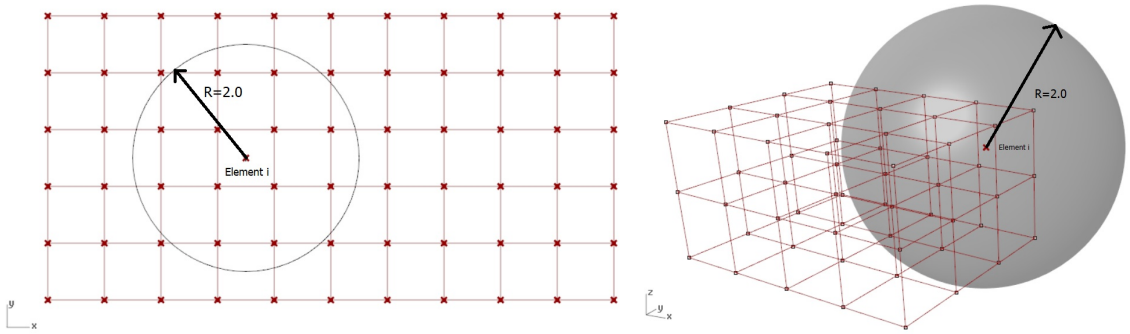


Figure 4.10: Neighbourhood of the ith element in 2D and 3D

In 2D H_{ij} can be further specified as:

$$H_{ij} = \max \left\{ \begin{array}{l} 0 \\ R - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \end{array} \right\} \tag{4.51}$$

In 3D H_{ij} is given by:

$$H_{ij} = \max \left\{ \begin{array}{l} 0 \\ R - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \end{array} \right\} \tag{4.52}$$

The design variables (x) are updated during the search for the minimum value of the objective function f. The sensitivity of the objective function to the design variables is needed and not just the sensitivity to the physical densities ($\tilde{x}$). The right value for the sensitivity can be found by using the chain rule:

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \left\{ \frac{\partial \tilde{x}_j}{\partial x_i} \frac{\partial f}{\partial \tilde{x}_j} \right\} = \sum_{j \in N_i} \left\{ \frac{H_{ij}}{\sum_{j \in N_i} H_{ij}} \frac{\partial f}{\partial \tilde{x}_j} \right\} \tag{4.53}$$

4.2.3 Sensitivity filter

Sensitivity filtering is a different filtering method that was introduced by Sigmund (1997) to avoid numerical instabilities. He replaced the real sensitivities with filtered sensitivities in every iteration to achieve that goal. The form of the sensitivity filter is:

$$\frac{\widehat{\partial f(x)}}{\partial x_i} = \frac{\sum_{j \in N_i} H_{ij} x_j}{max(\gamma, x_i) \sum_{j \in N_i} H_{ij}} \frac{\partial f(x)}{\partial x_j} \tag{4.54}$$

where $\gamma = 10^{-3}$ is a small number to avoid division by zero. The design variables are not filtered in this method.

4.2.4 Heaviside filter

Another filter that can be used is the Heaviside filter proposed by Guest et al. (2004) and Sigmund (2007). This filter is able to achieve a minimum length scale in the optimised design and can steer the design towards a black and white (solid and void) solution. To judge if a design is a black and white design the measure of non-discreteness is given by:

$$M_{nd} = \sum_{i=1}^{n_e} \frac{(\frac{1}{f(1-f)})\hat{x}_e(1-\hat{x}_e)}{n_e} 100\% \tag{4.55}$$

where f is the design volume fraction. 0% means that the design is fully discrete and thus only black and white elements exist. 100% means a fully homogeneous distribution.

The Heaviside function proposed is the continuous function:

$$\bar{x}_i = 1 - e^{-\beta \tilde{x}_i} + \tilde{x}_i e^{-\beta} \tag{4.56}$$

where β doubles every 50 iterations. The factor starts as $\beta = 1$ and stops increasing when $\beta = 512$. It starts with a low value to avoid local optima, since small changes in x result in strong changes in the projected densities for higher values of β. Because of the sudden change in β every 50 iterations there is a discontinuity in the convergence. Grey elements are suddenly steered towards black elements, which leads to changes in objective and constraint functions.

In a similar way as with density filtering the derivative with respect to the design variables need to be determined. This can be done by applying the chain rule twice:

$$\frac{\partial f(\bar{x})}{\partial x_i} = \sum_{j \in N_i} \left\{ \frac{\partial f(\bar{x})}{\partial \bar{x}_i} \frac{\partial \bar{x}_i}{\partial \tilde{x}_i} \frac{\partial \tilde{x}_i}{\partial x_i} \right\} \tag{4.57}$$

where the last term is given in Subsection 4.2.2 and the second term by:

$$\frac{\partial \bar{x}_i}{\partial \tilde{x}_i} = \beta e^{-\beta \tilde{x}_i} + e^{-\beta} \tag{4.58}$$

4.2.5 Robust filter

Another projection filter that can be used is the robust approach proposed by Wang et al. (2011). The Heaviside step function compares the filtered design variables $\tilde{x}$ with a

threshold value η_H. If the densities are lower than this value, they are projected to 0. On the other hand, if they are larger than η_H they are projected to 1.

The robust filter is a similar filter and is defined by:

$$\bar{x}_i = \frac{tanh(\beta\eta) + tanh(\beta(\tilde{x}_i - \eta))}{tanh(\beta\eta) + tanh(\beta(1-\eta))} \quad and \quad \eta = \frac{1}{2} \tag{4.59}$$

where η can have values between 0 and 1. In this approach three different density vectors will be calculated: a dilated ($0 < \eta_d < 0.5$), an intermediate ($\eta_n = 0.5$) and an eroded ($0.5 < \eta_e < 1.0$) design. For compliance minimisation the eroded design is the design with the worst compliance and is therefore the only design to consider.

The earlier mentioned measure of non-discreteness tends to go to zero using this filter. In for example compliance minimisation the compliance is calculated with the eroded design and the volume constraint with the nominal or intermediate design. Grey elements, which lead to an additional volume, will not contribute to the compliance and are therefore removed from the optimal solution.

Similar to the Heaviside filter the factor β is doubled every 50 iterations until a value of $\beta = 512$ is reached. And filtering of the sensitivities is done by applying the chain rule twice as well:

$$\frac{\partial f(\bar{x})}{\partial x_i} = \sum_{j \in N_i} \left\{ \frac{\partial f(\bar{x})}{\partial \bar{x}_i} \frac{\partial \bar{x}_i}{\partial \tilde{x}_i} \frac{\partial \tilde{x}_i}{\partial x_i} \right\} \tag{4.60}$$

where the middle term is different from the Heaviside projection filter and described as:

$$\frac{\partial \bar{x}_i}{\partial \tilde{x}_i} = \frac{\beta(sech(\beta(\tilde{x}_i - \eta)))^2}{tanh(\beta\eta) + tanh(\beta(1-\eta))} \tag{4.61}$$

4.3 Material model and Finite Element Analysis

In order to design an optimal structure in printed concrete the actual properties should be included in the optimisation. The used material in traditional optimisation methods is linear elastic and isotropic. For this kind of material the physical properties are same in all possible directions. Printed concrete behaves in a different way.

4.3.1 Material behaviour of printed concrete

Recent material tests at the Eindhoven University of Technology (Doomen (2016) and Slager (2017)) and other projects (Nerella and Mechtcherine 2016) show that the behaviour of printed concrete is depending on a lot of variables.

Probably the most important factor is the composition of the used material. A cement-like mixture is used in Eindhoven at the moment. The chosen mixture needs to be workable for a reasonable amount of time. The material needs to be transported from the location where the mixture is made, through the hoses of the printer, towards the desired position. The different layers on top of each other or next to each other should connect in a proper way to ensure a certain strength and stiffness, which is another reason for the mixture

not to dry too fast. However, the mixture should not be too wet for a long time either, because a minimum strength is needed to support the next layers. These factors cause challenges in determining the composition of the used mixture.

The strength and stiffness of printed concrete is not merely depending on the composition of the material. The print shape, print speed, pump pressure, temperature and other environmental conditions are aspects which influence the material behaviour as well. For this research the assumption is made that the aspects mentioned above are not varied during the printing process. In reality it is hard to achieve such an optimal and constant material.

Additive manufacturing of concrete is a *layer-based* process in which the layers can be placed next to each other and on top of each other. The direction of printing can be varied in the horizontal plane as well. The mechanical properties of structures created with this process are dependent on the print direction.

Recent tests (Doomen (2016) and Slager (2017)) on printed layers on top of each other show that the behaviour of printed concrete is considerably different than the behaviour of isotropic materials. In Eindhoven both compressive and tensile tests have been performed.

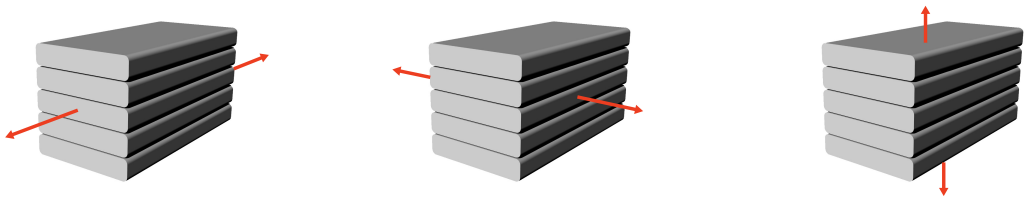


Figure 4.11: Tensile tests in x,y and z-direction performed at the TUe

The results from the tests (Doomen 2016) are listed in the tables below.

Compressive strength (N/mm^2)		Tensile strength (N/mm^2)	
$f_{c,x}$ =	25.43	$f_{t,x}$ =	1.69
$f_{c,y}$ =	19.73	$f_{t,y}$ =	1.59
$f_{c,z}$ =	22.20	$f_{t,z}$ =	1.00

Table 4.1: Strengths in different directions (Doomen 2016)

These results are matching the results in the research done by Slager (2017), which gives a minimum tensile strength of 1.08 N/mm^2 and a minimum compressive strength of 23.16 N/mm^2. In this research the lowest values are chosen for safety reasons: $f_t = 1.00$ N/mm^2 and $f_c = 19.73$ N/mm^2

The tests performed so far were focussed on determining the maximum stresses in the material. In addition values were measured and determined for the Young's moduli and the Poisson's ratios. However, the test set-up and the measuring equipment in the first research (Doomen 2016) were not chosen to determine these values. For that reason the values below should not be considered completely accurate.

Young's modulus (N/mm^2)				Poisson's ratio ($-$)	
$E_c =$	4500	$E_{t,x} =$	27715	$\nu_{t,x} =$	0.12
		$E_{t,y} =$	30376	$\nu_{t,y} =$	0.14
		$E_{t,z} =$	19596	$\nu_{t,z} =$	0.11

Table 4.2: Young's moduli en Poisson's ratios in different directions (Doomen 2016)

After Doomen (2016), Slager (2017) came up with more reliable results for the Young's modulus. A value of 17336 N/mm^2 in compression and 11103 N/mm^2 in tension result from this research. Again it is noted that the results are not exact, due to the set-up of the equipment.

To make safe designs or to use the known material properties in analysis or optimisations, further research is desirable. The Young's moduli and Poisson's ratios should be measured in a more precise way. Another test that has to be performed is a shear test to determine the shear moduli in the different directions.

To design a concrete printed structure using topology optimisation the different properties in the different directions must be taken into account. The tests that are performed so far indicate a clear difference in strength en stiffness in the x- and y-direction in comparison to the z-direction. The material is much weaker in this latter direction.

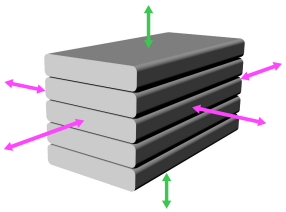


Figure 4.12: Different properties in two directions (two strong directions and one weak direction)

Considering these tests only it can be concluded that the used specimens behave in an orthotropic manner. However, in real designs multiple layers of the concrete-like mixture are printed next to each other as well. Hence, there is a vertical interface between the layers as well. The strength and stiffness in this direction depends on the interval time between the production of the layers and the horizontal distance between the layers.

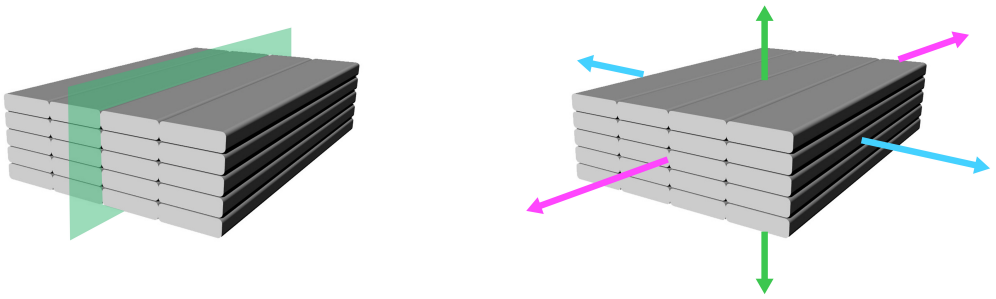


Figure 4.13: An additional (vertical) interface, hence different properties in three directions

At this moment no specimens have been tested with multiple layers next to each other. This will be a difficult task because of the many factors that can influence the result. If the circumstances are ideal and the layers are slightly overlapping, it can be assumed that the strength and stiffness in this y-direction equal those in x-direction. In that case an transverse isotropic material behaviour is assumed. This assumption needs to be incorporated into the algorithm, because this behaviour leads to dissimilar stiffness behaviour in different directions.

Figure 4.14: The real layers (left) and the transverse isotropic simplification (right)

4.3.2 Standard material model in topology optimisation

After the initial design is described in topology optimisation a finite element analysis is done to calculate the displacements of the nodes.

The isotropic material behaviour of standard topology optimisation algorithms can be found in the stiffness matrix. The nodal displacement vector $U(\hat{x})$ can be obtained by solving the equilibrium equation:

$$K(\hat{x})U(\hat{x}) = F \tag{4.62}$$

F is the load vector which is assigned and $K(\hat{x})$ is the global stiffness matrix.

The global stiffness matrix is obtained by an assembly of the elemental stiffness matrices. These elemental stiffness matrices are volume (3D) or surface (2D) integrals of the elements constitutive matrix $C_i(\hat{x}_i)$ an the strain-displacement matrix B.

The constitutive matrix or stiffness matrix $C_i(\hat{x})$ describes the relation between the stresses and the strains according to:

$$\sigma = C_i(\hat{x})\epsilon \tag{4.63}$$

The constitutive matrix or stiffness matrix of an isotropic element is interpolated in the modified SIMP method through:

$$C_i(\hat{x}_i) = E_i(\hat{x}_i)C_i^0 \tag{4.64}$$

the term C_i^0 in this equation is the constitutive matrix for unit Young's modulus and in 3D described as:

$$C_i^0 = \frac{1}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-2\nu)/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (1-2\nu)/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-2\nu)/2 \end{bmatrix} \tag{4.65}$$

In this relation ν is the Poisson's ratio of the isotropic material.

For 2D topology optimisation a simplified matrix can be used:

$$C_i^0 = \begin{bmatrix} \frac{1}{1-\nu^2} & \frac{\nu}{1-\nu^2} & 0 \\ \frac{\nu}{1-\nu^2} & \frac{1}{1-\nu^2} & 0 \\ 0 & 0 & \frac{1}{2(1+\nu)} \end{bmatrix} \tag{4.66}$$

Now the elemental stiffness matrix can be calculated. In 3D this matrix is the *volume* integral of the constitutive matrix and the strain-displacement matrix in the form of:

$$k_i(\hat{x}_i) = \int_{V_e} B^T C_i(\hat{x}_i) B \; dV = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} B^T C_i(\hat{x}_i) B \; det[J] \; d\xi_1 d\xi_2 d\xi_3 \tag{4.67}$$

Where ξ_1, ξ_2 and ξ_3 are the so called natural coordinates as indicated in Figure 4.15 and J is the Jacobian matrix.

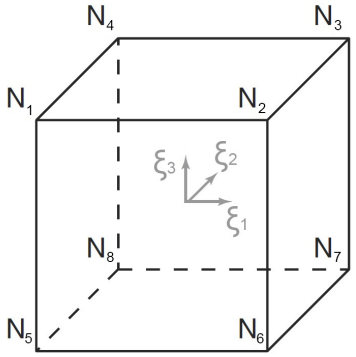


Figure 4.15: Hexahedron with eight nodes and the natural coordinates

The local hexahedron coordinates of the corners are listed in Table 4.3.

Node	ξ_1	ξ_2	ξ_3
1	-1	-1	+1
2	+1	-1	+1
3	+1	+1	+1
4	-1	+1	+1
5	-1	-1	-1
6	+1	-1	-1
7	+1	+1	-1
8	-1	+1	-1

Table 4.3: The conventions for node numbering of the eight-node hexahedral element

The strain-displacement matrix B indicates the relation between strains and displacements as in: $\epsilon = Bu$. For an eight-node hexahedral element the strain-displacement matrix B is given by:

$$B = \begin{bmatrix} \frac{\partial n_1(\xi_e)}{\partial \xi_1} & 0 & 0 & ... & \frac{\partial n_q(\xi_e)}{\partial \xi_1} & 0 & 0 \\ 0 & \frac{\partial n_1(\xi_e)}{\partial \xi_2} & 0 & ... & 0 & \frac{\partial n_q(\xi_e)}{\partial \xi_2} & 0 \\ 0 & 0 & \frac{\partial n_1(\xi_e)}{\partial \xi_3} & ... & 0 & 0 & \frac{\partial n_q(\xi_e)}{\partial \xi_3} \\ \frac{\partial n_1(\xi_e)}{\partial \xi_2} & \frac{\partial n_1(\xi_e)}{\partial \xi_1} & 0 & ... & \frac{\partial n_q(\xi_e)}{\partial \xi_2} & \frac{\partial n_q(\xi_e)}{\partial \xi_1} & 0 \\ 0 & \frac{\partial n_1(\xi_e)}{\partial \xi_3} & \frac{\partial n_1(\xi_e)}{\partial \xi_2} & ... & 0 & \frac{\partial n_q(\xi_e)}{\partial \xi_3} & \frac{\partial n_q(\xi_e)}{\partial \xi_2} \\ \frac{\partial n_1(\xi_e)}{\partial \xi_3} & 0 & \frac{\partial n_1(\xi_e)}{\partial \xi_1} & ... & \frac{\partial n_q(\xi_e)}{\partial \xi_3} & 0 & \frac{\partial n_q(\xi_e)}{\partial \xi_1} \end{bmatrix} \tag{4.68}$$

In which $e = 1, ..., 3$ and $q = 1, ..., 8$. The shape functions n_q for these sort of elements in a natural coordinate system ξ_e are as follows:

$$n_q(\xi_e) = \frac{1}{8} \begin{Bmatrix} (1-\xi_1)(1-\xi_2)(1+\xi_3) \\ (1+\xi_1)(1-\xi_2)(1+\xi_3) \\ (1+\xi_1)(1+\xi_2)(1+\xi_3) \\ (1-\xi_1)(1+\xi_2)(1+\xi_3) \\ (1-\xi_1)(1-\xi_2)(1-\xi_3) \\ (1+\xi_1)(1-\xi_2)(1-\xi_3) \\ (1+\xi_1)(1+\xi_2)(1-\xi_3) \\ (1-\xi_1)(1+\xi_2)(1-\xi_3) \end{Bmatrix} \tag{4.69}$$

The elemental stiffness matrix in 2D is the *surface* integral of the constitutive matrix and the strain-displacement matrix in the form of:

$$k_i(\hat{x}_i) = \int_{A_e} B^T C_i(\hat{x}_i) B \ dA = \int_{-1}^{+1} \int_{-1}^{+1} B^T C_i(\hat{x}_i) B \ det[J] \ d\xi_1 d\xi_2 \tag{4.70}$$

Where ξ_1 and ξ_2 are the natural coordinates as indicated in Figure 4.16 and J is the Jacobian matrix.

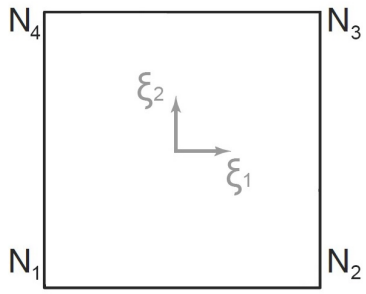


Figure 4.16: Quadrilateral element with four nodes and the natural coordinates

The local coordinates of the corners are listed in Table 4.4.

Node	ξ_1	ξ_2
1	-1	-1
2	+1	-1
3	+1	+1
4	-1	+1

Table 4.4: The conventions for node numbering of the four-node quadrilateral element

The strain-displacement matrix B indicates the relation between strains and displacements as in: $\epsilon = Bu$. For a four-node quadrilateral element the strain-displacement matrix B is given by:

$$B = \begin{bmatrix} \frac{\partial n_1(\xi_e)}{\partial \xi_1} & 0 & ... & \frac{\partial n_q(\xi_e)}{\partial \xi_2} & 0 \\ 0 & \frac{\partial n_1(\xi_e)}{\partial \xi_2} & ... & 0 & \frac{\partial n_q(\xi_e)}{\partial \xi_2} \\ \frac{\partial n_1(\xi_e)}{\partial \xi_2} & \frac{\partial n_1(\xi_e)}{\partial \xi_1} & ... & \frac{\partial n_q(\xi_e)}{\partial \xi_2} & \frac{\partial n_q(\xi_e)}{\partial \xi_1} \end{bmatrix} \tag{4.71}$$

In which $e = 1, 2$ and $q = 1, ..., 4$. The shape functions n_q for these sort of elements in a natural coordinate system ξ_e are as follows:

$$n_q(\xi_e) = \frac{1}{4} \begin{Bmatrix} (1-\xi_1)(1-\xi_2) \\ (1+\xi_1)(1-\xi_2) \\ (1+\xi_1)(1+\xi_2) \\ (1-\xi_1)(1+\xi_2) \end{Bmatrix} \tag{4.72}$$

By making use of the modified SIMP method the element stiffness matrix can be interpolated from void to solid as:

$$k_i(\hat{x}_i) = E_i(\hat{x}_i) k_i^0 \tag{4.73}$$

where the unit constitutive matrix is used to calculate the constant matrix (3D):

$$k_i^0 = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} B^T C^0 B \ \ det[J] \ \ d\xi_1 d\xi_2 d\xi_3 \tag{4.74}$$

And for 2D problems:

$$k_i^0 = \int_{-1}^{+1} \int_{-1}^{+1} B^T C^0 B \ \ det[J] \ \ d\xi_1 d\xi_2 \tag{4.75}$$

Using the global versions of the elemental stiffness matrices (K_i and K_i^0) it is possible to calculate $K(\hat{x})$ using:

$$K(\tilde{x}) = \sum_{i=1}^{n} K_i(\tilde{x}_i) = \sum_{i=1}^{n} E_i(\hat{x}_i) K_i^0 \tag{4.76}$$

In combination with Equation 4.46 this can be rewritten as:

$$K(\hat{x}) = \sum_{i=1}^{n} [E_{min} + x_i^p (E_0 - E_{min})] K_i^0 \tag{4.77}$$

Now the displacements of the nodes can be calculated using the equilibrium in Equation 4.62. In a different form this holds:

$$U(\hat{x}) = K(\hat{x})^{-1} F \tag{4.78}$$

4.3.3 Transverse isotropic material model for topology optimisation

To apply the assumption that the material is not isotropic to the optimisation algorithm the stiffness matrices need to be adapted. In case of anisotropic, orthotropic or transverse isotropic material behaviour different element stiffness matrices need to be created. All elements will be transverse isotropic elastic in the modified method. In 2D there is only a difference in stiffness in the x- and y-direction, so the material can be considered orthotropic. The Solid Orthotropic Material with Penalisation (SOMP) method (Alamo and da Silva 2012) can be implemented.

4.3.3.1 2D material model

The matrix C_i^0 is the stiffness matrix for unit Young's modulus or elasticity matrix derived from Hooke's Law and can be obtained by taking the inverse of the compliance matrix S, which is described in the following equation:

$$\epsilon = S\sigma \tag{4.79}$$

For the determination of the exact matrix S five elastic engineering constants have to be known: the two Young's modulii, the two Poisson's ratios and the shear modulus G_{xy}:

$$\begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_x} & -\frac{\nu_{yx}}{E_y} & 0 \\ -\frac{\nu_{xy}}{E_x} & \frac{1}{E_y} & 0 \\ 0 & 0 & \frac{1}{G_{xy}} \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \tag{4.80}$$

where

$$\frac{\nu_{yx}}{E_y} = \frac{\nu_{xy}}{E_x}, \tag{4.81}$$

As stated above the stiffness matrix C_i^0 is found from the inverse of the compliance matrix ($C_i^0 = S^{-1}$) and the following holds:

$$\sigma = C_i^0 \epsilon \tag{4.82}$$

Again in terms of the five engineering constants:

$$\begin{bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{xy}\end{bmatrix} = \begin{bmatrix}\frac{E_x}{1-\nu_{xy}\nu_{yx}} & \frac{\nu_{xy}E_y}{1-\nu_{xy}\nu_{yx}} & 0\\ \frac{\nu_{yx}E_x}{1-\nu_{xy}\nu_{yx}} & \frac{E_y}{1-\nu_{xy}\nu_{yx}} & 0\\ 0 & 0 & G_{xy}\end{bmatrix}\begin{bmatrix}\epsilon_{xx}\\ \epsilon_{yy}\\ \gamma_{xy}\end{bmatrix} \tag{4.83}$$

The strongest and stiffest direction of printing is the *x-direction* and so E_x is inserted as 1.0. If the Young's modulus in the y-direction is two times smaller than in x-direction, $E_y = 0.5$. So the Young's modulus in y-direction is always a factor of E_x. The real value of the elastic modulus of the concrete is assigned at the interpolation between void and solid.

The strain-displacement matrix B indicates the relation between strains and displacements as in: $\epsilon = Bu$. For a four-node quadrilateral element the strain-displacement matrix B is given before.

Now the elemental stiffness matrix can be calculated. In 2D this matrix is the surface integral of the constitutive matrix and the strain-displacement matrix in the form of:

$$k_i(\hat{x}_i) = \int_{A_e} B^T C_i(\hat{x}_i) B \ dA = \int_{-1}^{+1}\int_{-1}^{+1} B^T C_i(\hat{x}_i) B \ det[J] \ d\xi_1 d\xi_2 \tag{4.84}$$

Where ξ_1 and ξ_2 are the natural coordinates as indicated in Figure 4.16 and J is the Jacobian matrix.

By making use of the modified SOMP method the element stiffness matrix can be interpolated from void to solid as:

$$k_i(\hat{x}_i) = E_i(\hat{x}_i) k_i^0 \tag{4.85}$$

where the unit constitutive matrix is used to calculate the constant matrix:

$$k_i^0 = \int_{-1}^{+1}\int_{-1}^{+1} B^T C^0 B \ det[J] \ d\xi_1 d\xi_2 \tag{4.86}$$

4.3.3.2 3D material model

In 3D a similar approach is followed. However, now the elemental stiffness matrix k_i^0 can be defined as:

$$k_i^0 = \int_{V_e} B^T C_i^0 B \quad dV \tag{4.87}$$

where B is the strain-displacement matrix and the integration is performed over element volume V_e. In case eight-node hexahedrons are used with natural coordinates ξ_1, ξ_2 and ξ_3 Equation 4.87 becomes:

$$k_i^0 = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} B^T C_i^0 B \ det[J] \ d\xi_1 d\xi_2 d\xi_3 \tag{4.88}$$

For an eight-node hexahedral element the strain-displacement matrix B is given before.

C_i^0 is the constitutive matrix or, in this case, the 3D elasticity matrix derived from Hooke's Law and can be obtained by taking the inverse of the compliance matrix S.

For the determination of the exact matrix nine elastic engineering constants have to be known: the three Young's modulii, the three Poisson's ratios and the three shear modulii:

$$\begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_x} & -\frac{\nu_{yx}}{E_y} & -\frac{\nu_{zx}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{xy}}{E_x} & \frac{1}{E_y} & -\frac{\nu_{zy}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{xz}}{E_x} & -\frac{\nu_{yz}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{zx}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{xy}} \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{bmatrix} \tag{4.89}$$

where

$$\frac{\nu_{yz}}{E_y} = \frac{\nu_{zy}}{E_z}, \frac{\nu_{zx}}{E_z} = \frac{\nu_{xz}}{E_x}, \frac{\nu_{xy}}{E_x} = \frac{\nu_{yx}}{E_y} \tag{4.90}$$

As stated above the stiffness matrix C_i^0 is found from the inverse of the compliance matrix ($C_i^0 = S^{-1}$) and the following holds for orthotropic material:

$$\sigma = C_i^0 \epsilon \tag{4.91}$$

Again in terms of the nine engineering constants:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{1-\nu_{yz}\nu_{zy}}{E_y E_z \Delta} & \frac{\nu_{yx}+\nu_{zx}\nu_{yz}}{E_y E_z \Delta} & \frac{\nu_{zx}+\nu_{yx}\nu_{zy}}{E_y E_z \Delta} & 0 & 0 & 0 \\ \frac{\nu_{xy}+\nu_{xz}\nu_{zy}}{E_z E_x \Delta} & \frac{1-\nu_{zx}\nu_{xz}}{E_z E_x \Delta} & \frac{\nu_{zy}+\nu_{zx}\nu_{xy}}{E_z E_x \Delta} & 0 & 0 & 0 \\ \frac{\nu_{xz}+\nu_{xy}\nu_{yz}}{E_x E_y \Delta} & \frac{\nu_{yz}+\nu_{xz}\nu_{yx}}{E_x E_y \Delta} & \frac{1-\nu_{xy}\nu_{yx}}{E_x E_y \Delta} & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{yz} & 0 & 0 \\ 0 & 0 & 0 & 0 & G_{zx} & 0 \\ 0 & 0 & 0 & 0 & 0 & G_{xy} \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \end{bmatrix} \tag{4.92}$$

where

$$\Delta = \frac{1 - \nu_{xy}\nu_{yx} - \nu_{yz}\nu_{zy} - \nu_{zx}\nu_{xz} - 2\nu_{xy}\nu_{yz}\nu_{zx}}{E_x E_y E_z} \tag{4.93}$$

In case of orthotropic material behaviour typically one strong axis is defined and two weaker axes. In case the horizontal layers work well together, as assumed, the material contains two strong axes (x- and y-axis) and one weak axis (z-axis). The material behaves in a *transverse isotropic* manner. In that case only five engineering constraints have to be known. The Young's modulus and Poisson's ratio in the x-y symmetry plane, E_p and ν_p, the Young's modulus and Poison's ratio in the z-direction, E_{pz} and ν_{pz} and the shear modulus in z-direction G_{zp}.

The following now holds for transverse isotropic material:

$$\begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_p} & -\frac{\nu_p}{E_p} & -\frac{\nu_{zp}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_p}{E_p} & \frac{1}{E_p} & -\frac{\nu_{zp}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{pz}}{E_p} & -\frac{\nu_{pz}}{E_p} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{zp}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{zp}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2(1+\nu_p)}{E_p} \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{bmatrix} \tag{4.94}$$

where

$$\frac{\nu_{pz}}{E_p} = \frac{\nu_{zp}}{E_z}, \tag{4.95}$$

By taking the inverse of this compliance matrix the stiffness matrix is obtained, which can be used in the calculation of de elemental stiffness matrix.

It is important to note that the values for the engineering constants are simply parameters in the algorithm and they can be adapted at any moment. For example, when new research is done in determining these values. Another important aspect is that elastic properties and material orientation are now determined and will not change during the optimisation. The constitutive matrix C_i^0 is constant during the optimisation as well. It is only multiplied and thus scaled by the elemental density in the calculation of the compliance.

4.3.3.3 Orthogonal orientation and the orientation of the baseplate

For materials with different material properties in different directions it is important to take into account the print direction and the orientation of the baseplate. Structures can simply be printed from bottom-to-top, however it is also possible to print the design in a different way and rotate it afterwards, if that results in a stiffer or stronger structure, or if material can be saved by rotating the baseplate and print direction.

In both 2D and 3D the direction of the printing path can be adapted in order to find a preferred angle θ. This orientation can possibly become a parameter in the topology optimisation in the future. In this research the angle can be changed, however is not a parameter that is optimised.

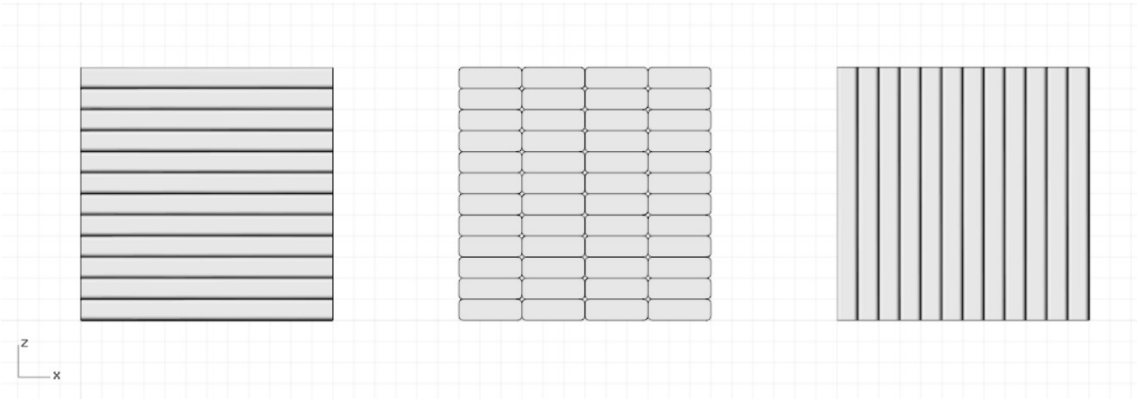


Figure 4.17: Different directions of the orthogonal orientation (Left: no rotation, center: 90 degrees rotation around z-axis, right: 90 degrees rotation around x-axis))

4.3.3.3.1 2D orientation and rotation

In the algorithm for 2D problems the regular orthotropic constitutive or stiffness matrix can be rotated around the z-axis to change the baseplate. The standard baseplate is the

bottom of the design space, where $y = 0$. The standard matrix is defined as:

$$C_1 = \begin{bmatrix} \frac{E_x}{1-\nu_{xy}\nu_{yx}} & \frac{\nu_{xy}E_y}{1-\nu_{xy}\nu_{yx}} & 0 \\ \frac{\nu_{yx}E_x}{1-\nu_{xy}\nu_{yx}} & \frac{E_y}{1-\nu_{xy}\nu_{yx}} & 0 \\ 0 & 0 & G_{xy} \end{bmatrix} \tag{4.96}$$

The rotated matrix C_2 is defined as:

$$C_2 = (\alpha)C_1(\alpha)^T \tag{4.97}$$

where rotation matrix (α) is defined as:

$$(\alpha) = R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.98}$$

In this research the baseplates are limited to the four sides of the design space:

Baseplate	θ
South	0
East	90
North	180
West	270

Table 4.5: Definition of baseplate angles

The baseplates usually can take every value of θ between 0 and 360, however the baseplate will also be of influence on the printing constraints explained in Section 4.5, so in this research the choice is limited to four values.

4.3.3.3.2 3D orientation and rotation

In 3D the stiffness matrix C_1 can be rotated as well. In that case the standard reference plane or baseplate is the plane in which $z = 0$. This plane can be rotated around the x-axis with a rotation θ_1 and around the y-axis with a rotation θ_2. The normal to this plane is the z-axis, which is the axis that is rotated.

The rotation matrix for a rotation around the x-axis is described as:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{bmatrix} \tag{4.99}$$

for the y-axis this matrix is:

$$R_y = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \tag{4.100}$$

The rotation matrix for the transformation is:

$$R = R_x R_y = \begin{bmatrix} \cos\theta_2 & \sin\theta_1 \sin\theta_2 & \cos\theta_1 \sin\theta_2 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ -\sin\theta_2 & \sin\theta_1 \cos\theta_2 & \cos\theta_1 \cos\theta_2 \end{bmatrix} \tag{4.101}$$

According to Newnham (2005) the coordinate transformation matrix (α) can be created from the matrix R. The components a_{ij} are the components from the matrix R.

$$(\alpha) = \begin{bmatrix} a_{11}^2 & a_{12}^2 & a_{13}^2 & 2a_{12}a_{13} & 2a_{11}a_{13} & 2a_{11}a_{12} \\ a_{21}^2 & a_{22}^2 & a_{23}^2 & 2a_{22}a_{23} & 2a_{21}a_{23} & 2a_{21}a_{22} \\ a_{31}^2 & a_{32}^2 & a_{33}^2 & 2a_{32}a_{33} & 2a_{31}a_{33} & 2a_{31}a_{32} \\ a_{21}a_{31} & a_{22}a_{32} & a_{23}a_{33} & (a_{22}a_{33} + a_{23}a_{32}) & (a_{21}a_{33} + a_{23}a_{31}) & (a_{22}a_{31} + a_{21}a_{32}) \\ a_{31}a_{11} & a_{32}a_{12} & a_{33}a_{13} & (a_{12}a_{33} + a_{13}a_{32}) & (a_{13}a_{31} + a_{11}a_{33}) & (a_{11}a_{32} + a_{12}a_{31}) \\ a_{11}a_{21} & a_{12}a_{22} & a_{13}a_{23} & (a_{12}a_{23} + a_{13}a_{22}) & (a_{13}a_{21} + a_{11}a_{23}) & (a_{11}a_{22} + a_{12}a_{21}) \end{bmatrix} \tag{4.102}$$

filling in the values from matrix R this leads to:

$$(\alpha) = \begin{bmatrix} \cos^2\theta_2 & \sin^2\theta_1 \sin^2\theta_2 & \cos^2\theta_1 \sin^2\theta_2 & \sin 2\theta_1 \sin^2\theta_2 & \cos\theta_1 \sin 2\theta_2 & \sin\theta_1 \sin 2\theta_2 \\ 0 & \cos^2\theta_1 & \sin^2\theta_1 & -\sin 2\theta_1 & 0 & 0 \\ \sin^2\theta_2 & \sin^2\theta_1 \cos^2\theta_2 & \cos^2\theta_1 \cos^2\theta_2 & \sin 2\theta_1 \cos^2\theta_2 & -\cos\theta_1 \sin 2\theta_2 & -0.5 \sin\theta_1 \sin 2\theta_2 \\ 0 & 0.5 \sin 2\theta_1 \cos\theta_2 & -0.5 \sin 2\theta_1 \cos\theta_2 & \cos 2\theta_1 \cos\theta_2 & \sin\theta_1 \sin\theta_2 & -\cos\theta_1 \sin\theta_2 \\ -0.5 \sin 2\theta_2 & 0.5 \sin^2\theta_1 \sin 2\theta_2 & 0.5 \cos^2\theta_1 \sin 2\theta_2 & 0.5 \sin 2\theta_1 \sin 2\theta_2 & \cos\theta_1 \cos 2\theta_2 & \sin\theta_1 \cos 2\theta_2 \\ 0 & 0.5 \sin 2\theta_1 \sin\theta_2 & -0.5 \sin 2\theta_1 \sin\theta_2 & \cos 2\theta_1 \sin\theta_2 & -\sin\theta_1 \cos\theta_2 & \cos\theta_1 \cos\theta_2 \end{bmatrix} \tag{4.103}$$

To obtain the rotated stiffness matrix C_2 the transformation matrix (α) and the original matrix C_1 are used to compile:

$$C_2 = (\alpha) C_1 (\alpha)^T \tag{4.104}$$

The baseplates and matching angles θ_1 and θ_2 defined in this research are listed in Table 4.6.

Baseplate	θ_1	θ_2
Bottom	0	0
Top	0	0
Left	90	0
Right	90	0
Front	0	90
Back	0	90

Table 4.6: Definition of baseplate angles in 3D

4.4 Sensitivity analysis and filtering

After the displacements are calculated in the FEA, the values for the objective function and the constraints are determined according to Section 4.1.2 and Section 4.1.3. Before the optimisation can start to find a new design (a new set of x) the next step in topology optimisation is to obtain the *sensitivities* of the objective function and the sensitivities

of the constraint functions. This gradient information is needed to steer the final design towards an optimum. The derivatives of the structural responses with respect to the design variables x are determined in this section.

As mentioned before the sensitivities for a density filter can be calculated by:

$$\frac{\partial f(\tilde{x})}{\partial x_i} = \sum_{j \in N_i} \left\{ \frac{\partial f}{\partial \tilde{x}_j} \frac{\partial \tilde{x}_j}{\partial x_i} \right\} \tag{4.105}$$

in case of a sensitivity filter:

$$\frac{\widehat{\partial f(x)}}{\partial x_i} = \frac{\sum_{j \in N_i} H_{ij} x_j}{max(\gamma, x_i) \sum_{j \in N_i} H_{ij}} \frac{\partial f(x)}{\partial x_j} \tag{4.106}$$

and in case of a Heaviside filter or a robust filter by:

$$\frac{\partial f(\bar{x})}{\partial x_i} = \sum_{j \in N_i} \left\{ \frac{\partial f(\bar{x})}{\partial \bar{x}_i} \frac{\partial \bar{x}_i}{\partial \tilde{x}_i} \frac{\partial \tilde{x}_i}{\partial x_i} \right\} \tag{4.107}$$

the unknown term (in blue) in these expressions is derived in this section of the research and is below denoted as:

$$\frac{\partial f}{\partial \hat{x}_i} \tag{4.108}$$

4.4.1 Sensitivity analysis of the objective functions

The sensitivities of the two introduced objective functions f_0 are derived below.

4.4.1.1 Compliance objective

In case the objective is to minimise the compliance $f_0 = c(\hat{x}) = F^T U(\hat{x})$.

We assume the design variable x_i is continuously changing between 0 and 1. The sensitivity of the objective function with respect to the change in filtered design variable is:

$$\frac{\partial c(\hat{x})}{\partial \hat{x}_i} = \frac{\partial F^T}{\partial \hat{x}_i} U(\hat{x}) + F^T \frac{\partial U(\hat{x})}{\partial \hat{x}_i} \tag{4.109}$$

To determine the sensitivity of the displacement vector we need to know the unknown factor $\frac{\partial U(\tilde{x})}{\partial \hat{x}_i}$ first. To solve this problem a vector of Lagrangian multiplier λ in an extra term $\lambda^T(F - K(\tilde{x})U(\tilde{x}))$ is introduced (Huang and Xie 2010). This term is added to the objective function.

$$c(\hat{x}) = F^T U(\hat{x}) + \lambda^T (F - K(\hat{x})U(\hat{x})) \tag{4.110}$$

This can be done, because regardless of the value of λ the extra term will always be zero ($F = K(\hat{x})U(\hat{x})$ so $F - K(\hat{x})U(\hat{x}) = 0$) . Now the sensitivity of the modified function is:

$$\begin{aligned} \frac{\partial c(\hat{x})}{\partial \hat{x}_i} = \quad & \frac{\partial F^T}{\partial \hat{x}_i} U(\hat{x}) + F^T \frac{\partial U(\hat{x})}{\partial \hat{x}_i} + \frac{\partial \lambda^T}{\partial \hat{x}_i} (F - K(\hat{x})U(\hat{x})) \\ & + \lambda^T \left(\frac{\partial F}{\partial \hat{x}_i} - \frac{\partial K(\hat{x})}{\partial \hat{x}_i} U(\hat{x}) - K(\hat{x}) \frac{\partial U(\hat{x})}{\partial \hat{x}_i} \right) \end{aligned} \tag{4.111}$$

In this equation $F - K(\hat{x})U(\hat{x})$ equals 0 as stated above, so that term can be deleted. $\frac{\partial F}{\partial \hat{x}_i}$ equals zero as well, because the variation of the element density has no effect on the load vector if self-weight is not considered. So the new formula is:

$$\frac{\partial c(\hat{x})}{\partial \hat{x}_i} = (F^T - \lambda^T K(\hat{x}))\frac{\partial U(\hat{x})}{\partial \hat{x}_i} - \lambda^T \frac{\partial K(\hat{x})}{\partial \hat{x}_i} U(\hat{x}) \tag{4.112}$$

To eliminate the unknown $\frac{\partial U(\hat{x})}{\partial \hat{x}_i}$ from the equation, a value for λ needs to be chosen so that:

$$F^T - \lambda^T K(\hat{x}) = 0 \tag{4.113}$$

according to the equilibrium in Equation 4.62 it holds that: $\lambda = U(\hat{x})$

Substituting this value into the latest formula for the sensitivity of the objection function gives:

$$\frac{\partial c(\hat{x})}{\partial \hat{x}_i} = -U(\hat{x})^T \frac{\partial K(\hat{x})}{\partial \hat{x}_i} U(\hat{x}) \tag{4.114}$$

The derivative of K with respect to x_i is:

$$\begin{aligned} \frac{\partial K(\hat{x})}{\partial \hat{x}_i} &= \frac{\partial}{\partial \hat{x}_i} \sum_{i=1}^{n} [E_{min} + \hat{x}_i^p (E_0 - E_{min})] K_i^0 \\ &= p\hat{x}_i^{p-1}(E_0 - E_{min}) K_i^0 \end{aligned} \tag{4.115}$$

The final compliance sensitivity function is given by:

$$\begin{aligned} \frac{\partial c(\hat{x})}{\partial \hat{x}_i} &= -U(\hat{x})^T [p\hat{x}_i^{p-1}(E_0 - E_{min}) K_i^0] U(\hat{x}) \\ &= -p(E_0 - E_{min})\hat{x}_i^{p-1} U(\hat{x})^T K_i^0 U(\hat{x}) \end{aligned} \tag{4.116}$$

If self-weight is considered an additional term is added to this derivative, since a part of F is not independent from $\hat{x}$, according to:

$$F_{sw} = V_e \rho_e g \bar{F} = V_e \rho_0 \hat{x} \bar{F} \tag{4.117}$$

If the same approach is used as above Equation 4.114 changes to:

$$\frac{\partial c(\hat{x})}{\partial \hat{x}_i} = \frac{dF_{sw}(\hat{x})}{d\hat{x}} 2U(\hat{x}) - U(\hat{x})^T \frac{\partial K(\hat{x})}{\partial \hat{x}_i} U(\hat{x}) \tag{4.118}$$

In that way the compliance sensitivity considering self-weight is given by:

$$\frac{\partial c(\hat{x})}{\partial \hat{x}_i} = 2V_e \rho_0 \bar{F} U(\hat{x}) - p(E_0 - E_{min})\hat{x}_i^{p-1} U(\hat{x})^T K_i^0 U(\hat{x}) \tag{4.119}$$

The sensitivity can now have values higher than zero, because of the additional self-weight part. A non-monotonous solver is advised when using this derivative to avoid oscillating behaviour. Without the self-weight addition this is not needed, because the compliance sensitivity will always be negative.

4.4.1.2 Volume objective

In case the objective is to minimise the volume the objective function is:

$$f_0 = v(\hat{x}) = \frac{1}{n_e} \sum_{n=1}^{n_i} \hat{x} v_i. \tag{4.120}$$

The sensitivity of the volume objective is calculated by:

$$\frac{\partial v(\hat{x})}{\partial \hat{x}_i} = \frac{v_i}{n_e} \tag{4.121}$$

4.4.2 Sensitivity analysis of the constraint functions

Not just the sensitivities of the objective function are needed as inputs for the optimisation solvers, the derivatives of the constraint functions need to be determined as well.

4.4.2.1 Volume constraint

The volume constraint is described as:

$$g_v = \frac{\sum_{n=1}^{n_e} \hat{x} v_i}{v_{lim} n_e} - 1 \tag{4.122}$$

The derivative of this volume constraint with respect to the filtered design variable x_i is:

$$\frac{\partial g_v(\hat{x})}{\partial \hat{x}_i} = \frac{v_i}{v_{lim} n_e} \tag{4.123}$$

4.4.2.2 Compliance constraint

In order to adapt a stiffness constraint, the derivative of the following function needs to be determined:

$$g_c = \frac{c(\hat{x})}{c_{lim}} - 1 \tag{4.124}$$

The sensitivity of this constraint towards the change in filtered design variable is:

$$\frac{\partial g_c(\hat{x})}{\partial \hat{x}_i} = \frac{1}{c_{lim}} \frac{\partial c(\hat{x})}{\partial \hat{x}_i} = \frac{-p(E_0 - E_{min}) \hat{x}_i^{p-1} U(\hat{x})^T K_i^0 U(\hat{x})}{c_{lim}} \tag{4.125}$$

or in case self-weight is considered:

$$\frac{\partial g_c(\hat{x})}{\partial \hat{x}_i} = \frac{1}{c_{lim}} \frac{\partial c(\hat{x})}{\partial \hat{x}_i} = \frac{2 V_e \rho_0 \bar{F} U(\hat{x}) - p(E_0 - E_{min}) \hat{x}_i^{p-1} U(\hat{x})^T K_i^0 U(\hat{x})}{c_{lim}} \tag{4.126}$$

4.4.2.3 Von Mises stress constraint

The calculation of the von Mises stress constraint is a bit more challenging. As derived before the constraint is stated as:

$$g_{s,1} = \frac{c_{norm}\sigma_{PN_{vM}}}{\sigma_{vM_{lim}}} - 1 \tag{4.127}$$

The part of the constraint that depends on the filtered design variable is $\sigma_{PN_{vM}}$. Now the adjoint method is applied, similar to in Equation 4.110 an extra term is added:

$$h = \sigma_{PN} + \lambda^T(F - K(\hat{x})U(\hat{x})) \tag{4.128}$$

differentiated with respect to $\hat{x}$ this becomes:

$$\frac{dh}{d\hat{x}_i} = \left(\sum_{a=1}^{n} \frac{\partial\sigma_{PN}}{\partial\sigma_{vM,a,i}}\frac{\partial\sigma_{vM,a,i}}{\partial U(\hat{x})} + \lambda^T K(\hat{x})\right)\frac{\partial U(\hat{x})}{\partial\hat{x}_i} + \lambda^T\frac{\partial K(\hat{x})}{\partial\hat{x}_i}U(\hat{x}) + \frac{\partial\sigma_{PN}}{\partial\hat{x}_i} \tag{4.129}$$

To eliminate the unknown $\frac{\partial U(\hat{x})}{\partial\hat{x}_i}$, which is difficult to determine, from the equation the part between brackets in the first term should equal zero. In that case:

$$\lambda = -\sum_{a=1}^{n}\frac{\partial\sigma_{PN}}{\partial\sigma_{vM,a,i}}\frac{\partial\sigma_{vM,a,i}}{\partial U(\hat{x})}K^{-1} \tag{4.130}$$

Equation 4.129 can now be written as:

$$\frac{\partial h}{\partial\hat{x}_i} = \lambda^T\frac{\partial K(\hat{x})}{\partial\hat{x}_i}U(\hat{x}) + \frac{\partial\sigma_{PN}}{\partial\hat{x}_i} \tag{4.131}$$

the terms that need to be known are described below:

$$\frac{\partial\sigma_{PN}}{\partial\hat{x}_i} = \left(\sum_{i=1}^{n_a}(\tilde{\sigma}_{vM,i})^P\right)^{\frac{1}{P}-1}\frac{\partial\tilde{\sigma}_{vM,i}}{\partial\hat{x}_i}(\tilde{\sigma}_{vM,i})^{P-1} \tag{4.132}$$

$$\frac{\partial\tilde{\sigma}_{vM,i}}{\partial\hat{x}_i} = (p-q)\sigma_{vM,i}\hat{x}_i^{p-q-1} \tag{4.133}$$

where p is the penalty factor and not the aggregation factor P.

$$\frac{\partial\sigma_{PN}}{\partial\tilde{\sigma}_{vM,a,i}} = \left(\sum_{a=1}^{n_a}(\tilde{\sigma}_{vM,a})^P\right)^{\frac{1}{P}-1}(\tilde{\sigma}_{vM,a})^{P-1} \tag{4.134}$$

The von Mises stress was described in matrix and vector notation as:

$$\tilde{\sigma}_{vM,a} = \sqrt{\tilde{\sigma_a}^T V\tilde{\sigma_a}} \tag{4.135}$$

in terms of the displacement U this is:

$$\tilde{\sigma}_{vM,a} = \sqrt{U^T C B^T V B C U} \tag{4.136}$$

Now it is easier to calculate the last needed derivative:

$$\frac{\partial \tilde{\sigma}_{vM,a,i}}{\partial U(\hat{x})} = \frac{\hat{x}_i^{p-q}}{\sigma_{vM,a}}(CB^T VBCU) \tag{4.137}$$

Now the adjoint variable λ can be solved using Equation 4.130. This λ can be inserted in Equation 4.131 to obtain the derivative.

The final von Mises stress sensitivity function is given by:

$$\begin{aligned} \frac{\partial g_{s,1}}{\partial \hat{x}_i} &= \frac{1}{\sigma_{vM_{lim}}}\left(\frac{\partial \sigma_{PN}}{\partial \hat{x}_i} + \lambda^T[p\hat{x}_i^{p-1}(E_0 - E_{min})K_i^0]U(\hat{x})\right) \\ &= \frac{1}{\sigma_{vM_{lim}}}\left(\frac{\partial \sigma_{PN}}{\partial \hat{x}_i} + p(E_0 - E_{min})\hat{x}_i^{p-1}\lambda^T K_i^0 U(\hat{x})\right) \end{aligned} \tag{4.138}$$

4.4.2.4 Drucker-Prager stress constraint

The last sensitivity that is needed to be able to optimise structures is the derivative of the Drucker-Prager stress constraint:

$$g_{s,2} = \frac{\sigma_{PN_{eq}}}{\sigma_{Lt}} - 1 \tag{4.139}$$

In a similar way as the von Mises constraint we use the adjoint method to derive the derivative of the p-norm $\sigma_{PN_{eq}}$, which is the part of the constraint that depends on the filtered design variable $\hat{x}$.

$$h = \sigma_{PN} + \lambda^T(F - K(\hat{x})U(\hat{x})) \tag{4.140}$$

differentiated with respect to $\hat{x}$ this becomes:

$$\frac{dh}{d\hat{x}_i} = \left(\sum_{a=1}^{n} \frac{\partial \sigma_{PN}}{\partial \sigma_{eq,a,i}} \frac{\partial \sigma_{eq,a,i}}{\partial U(\hat{x})} + \lambda^T K(\hat{x})\right) \frac{\partial U(\hat{x})}{\partial \hat{x}_i} + \lambda^T \frac{\partial K(\hat{x})}{\partial \hat{x}_i} U(\hat{x}) + \frac{\partial \sigma_{PN}}{\partial \hat{x}_i} \tag{4.141}$$

To eliminate the unknown $\frac{\partial U(\hat{x})}{\partial \hat{x}_i}$ from the equation the part between brackets in the first term should equal zero. In that case:

$$\lambda = -\sum_{a=1}^{n} \frac{\partial \sigma_{PN}}{\partial \sigma_{eq,a,i}} \frac{\partial \sigma_{eq,a,i}}{\partial U(\hat{x})} K^{-1} \tag{4.142}$$

Equation 4.141 can now be written as:

$$\frac{\partial h}{\partial \hat{x}_i} = \lambda^T \frac{\partial K(\hat{x})}{\partial \hat{x}_i} U(\hat{x}) + \frac{\partial \sigma_{PN}}{\partial \hat{x}_i} \tag{4.143}$$

the terms that need to be known are described below:

$$\frac{\partial \sigma_{PN}}{\partial \hat{x}_i} = \left(\sum_{i=1}^{n_a} (\tilde{\sigma}_{eq,i})^P\right)^{\frac{1}{P}-1} \frac{\partial \tilde{\sigma}_{eq,i}}{\partial \hat{x}_i} (\tilde{\sigma}_{eq,i})^{P-1} \tag{4.144}$$

$$\frac{\partial \tilde{\sigma}_{eq,i}}{\partial \hat{x}_i} = (p-q)\sigma_{eq,i}\hat{x}_i^{p-q-1} \tag{4.145}$$

where p is the penalty factor and not the p-norm factor P.

$$\frac{\partial \sigma_{PN}}{\partial \tilde{\sigma}_{eq,a,i}} = \left(\sum_{a=1}^{n_a}(\tilde{\sigma}_{eq,a})^P\right)^{\frac{1}{P}-1}(\tilde{\sigma}_{eq,a})^{P-1} \tag{4.146}$$

The Drucker-Prager equivalent stress was described in terms of the displacement U as:

$$\tilde{\sigma}_{eq,a} = \hat{x}^{p-q}\left(\frac{s+1}{2s}\sqrt{U^T M^0 U} + \frac{s-1}{2s}H^0 U\right) \tag{4.147}$$

The last derivative can be determined:

$$\frac{\partial \tilde{\sigma}_{eq,a}}{\partial U(\hat{x})} = \hat{x}_i^{p-q}\left(\frac{s+1}{2s}(U^T M^0 U)^{-\frac{1}{2}}M^0 U + \frac{s-1}{2s}H^0\right) \tag{4.148}$$

Now the adjoint variable λ can be solved using Equation 4.142. This λ can be inserted in Equation 4.143 to obtain the derivative.

The final Drucker-Prager equivalent stress sensitivity function is given by:

$$\begin{aligned}\frac{\partial g_{s,2}}{\partial \hat{x}_i} &= \frac{1}{\sigma_{Lt}}\left(\frac{\partial \sigma_{PN}}{\partial \hat{x}} + \lambda^T[p\hat{x}_i^{p-1}(E_0 - E_{min})K_i^0]U(\hat{x})\right) \\ &= \frac{1}{\sigma_{Lt}}\left(\frac{\partial \sigma_{PN}}{\partial \hat{x}} + p(E_0 - E_{min})\hat{x}_i^{p-1}\lambda^T K_i^0 U(\hat{x})\right)\end{aligned} \tag{4.149}$$

4.4.3 Filtering of the sensitivities

Before the problem can be solved, the sensitivities of the objective and the constraint functions need to be filtered according to the chosen filter. The final sensitivities are calculated using the partial derivatives derived in the previous sections, which are subsequently filtered according to Equation 4.53 when using the density filter, to Equation 4.54 when using the sensitivity filter, to Equation 4.57 in case of a Heaviside filter or to Equation 4.60 when the robust approach is chosen.

4.5 Manufacturing constraints

The limitations of a 3D printer are far less than the constraints of the methods used in current practice. Much more is possible with additive manufacturing than, for example, with pouring concrete. Only when costly and labour intensive moulds are created, pouring of concrete can produce forms that are more challenging. Although there are less limitations, these limitations or constraints of the AM process should be taken into account in the optimisation to avoid modifications or post-processing afterwards. That would add costs and reduce the performance.

Many researchers tried to incorporate the additive manufacturing constraints into the optimisation. The aim was to minimise the amount of support structure or to obtain

designs that were completely self-supported. At first, scientists like Brackett et al. (2011) and Leary et al. (2014) tried to solve the problem with a post-processing step. After the optimisation the design was made printable. Other researchers, such as Gaynor and Guest (2016) tried to include the constraint into the optimisation. They used nodal design variables and they have added a projection step after the filtering of the variables. If the density in the nodes is smaller than a certain *average* density in the support region, then the node can be projected as solid, otherwise it will be projected as a void. The support region is determined by an angle, defined by the critical overhang of the additive manufacturing method and material.

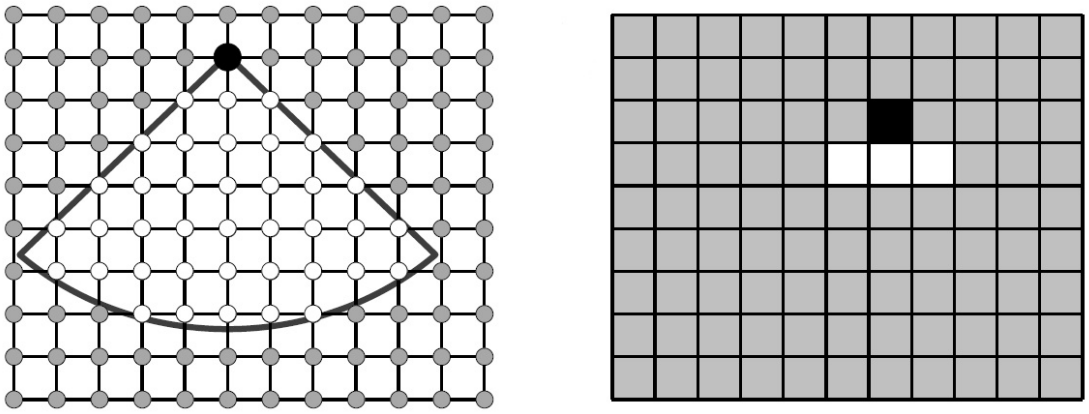


Figure 4.18: Support regions proposed by Gaynor and Guest (2016) (left) and by Langelaar (2017) (right)

Langelaar (2017) came up with a similar approach, however uses elemental densities instead of nodal densities. The support region is also different than the suggested region by Gaynor and Guest (2016) as visible in Figure 4.18. The supporting region is directly tied to the finite elements, which is good for the computational cost, however the overhang angle cannot be varied as easily as in the method of Gaynor and Guest (2016). It is fixed to the finite elements. Another limitation, which holds for both methods, is the fact that a predefined print direction and baseplate need to be chosen. They cannot be variable in the optimisation problem, which will lead to less optimal results.

To take into account the 3D printing constraints in this research the filter method as proposed by Langelaar (2017) is used. The filter considers a certain element i and its supporting elements. In case of a 45 degree angle filter there are three elements supporting a certain element i. The density of this element cannot exceed the maximum density of the three supporting elements, otherwise it is not supported in a proper way. In the subsections below the applied filters will be explained in more depth.

An important assumption that is made in this research, which is not actually correct, is the fact that the printer can start and stop printing at any time. In the future this might become possible with concrete additive manufacturing, however at the moment the printer can only print in continuous printing paths.

4.5.1 Printing constraint AM45

The 45 degree angle printing constraint, called AM45 in this research, is defined on the regular mesh created in the algorithm. The vertical direction or y-direction is the direction of printing, the first elements are printed on the baseplate.

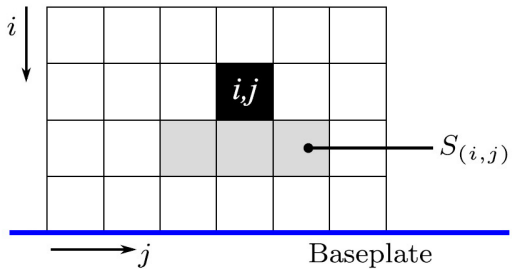


Figure 4.19: Definition of support region $S_{(i,j)}$ for element i, j in case of AM45

Every element in the design space contains a *blueprint* density $x_{(i,j)}$, where i and j are the vertical and horizontal locations of the elements. In this research the top layer is denoted as $i = 0$ and the bottom layer as $i = nely - 1$, where $nely$ is the amount of elements in y-direction. The next step is to express the densities that can be printed, called the *printed* densities $\xi(i, j)$.

The bottom layer can always be printed, because the baseplate supports these elements. For the layers above holds that the elements can have a printed density ξ that is not larger than the maximum printed density in its support region, called Ξ. The following holds for the layers above the baseplate:

$$\xi_{(i,j)} = min(x_{(i,j)}, \Xi_{(i,j)}) \qquad where \tag{4.150}$$

$$\Xi_{(i,j)} = max(\xi_{(i+1,j-1)}, \xi_{(i+1,j)}, \xi_{(i+1,j+1)}) \tag{4.151}$$

At the left and right edge of the domain the elements are only supported by two elements. A problem with the equations stated above is the fact that they are not differentiable, because of the *min* and the *max* operators. A smooth approximation for both equations is proposed by Langelaar (2017):

$$smin(x, \Xi) = \frac{1}{2}(x + \Xi - ((x - \Xi)^2 + \epsilon)^{\frac{1}{2}} + \sqrt{\epsilon}) \tag{4.152}$$

$$smax(\xi_1, \xi_2, \xi_3) = (\sum_{k=1}^{n_s} \xi_k^P)^{\frac{1}{Q}} \tag{4.153}$$

where ϵ and P control the accuracy and the smoothness of the approximated equations. If $\epsilon \to 0$ and $Q \to \infty$ the smoothness is lost, however the real min and max operators are obtained. Q is described as:

$$Q = P + \frac{\log n_s}{\log \xi_0} \tag{4.154}$$

where n_s is the number of supporting elements and $\xi_0 = 0.5$. The effect of changing these parameters is studied by Langelaar (2017), however kept constant in this research.

Not only the blueprint elemental densities should be filtered, the sensitivities of the objective and the constraint function as well. These functions, now called f, depend on the printed design, which depends on the blueprint design, so $f(\xi(x))$. Sensitivities with respect to the blueprint densities x are given by:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial \xi}\frac{\partial \xi}{\partial x} \tag{4.155}$$

To calculate these sensitivities an adjoint formulation is used by Langelaar (2017) and the blueprint and printed densities are combined in:

$$\xi_i = smin(x_i, \xi_{i+1}) = s(x_i, \xi_{i+1}) \tag{4.156}$$

where s is $smin$ and the layers are denoted by the subscripts. Using this relation and some calculation steps that can be consulted (Langelaar 2017) the following equation was derived for the sensitivities:

$$\frac{\partial f}{\partial x_k} = \lambda_k^T \frac{\partial s_k}{\partial x_k} \tag{4.157}$$

where

$$\lambda_k^T = \frac{\partial f}{\partial \xi_k} \tag{4.158}$$

for the top layer, where $k = 0$. For the layers in between the top and the bottom layer holds:

$$\lambda_k^T = \frac{\partial f}{\partial \xi_k} + \lambda_{k-1}^T \frac{\partial s_{k-1}}{\partial \xi_k} \tag{4.159}$$

The calculation of the printed densities was performed from bottom to top, however the calculation of the transformed sensitivities is done from top to bottom (layer $0 \rightarrow n_i$). The partial derivatives in the formulas above are given by:

$$\frac{\partial s_k}{\partial x_k} = \frac{1}{2}(1 - (x - \Xi)((x - \Xi)^2 + \epsilon)^{-\frac{1}{2}}) \tag{4.160}$$

$$\frac{\partial s_k}{\partial \xi_k} = \frac{\partial s_k}{\partial \Xi_k}\frac{\partial \Xi_k}{\partial \xi_k} \tag{4.161}$$

$$\frac{\partial s_k}{\partial \Xi_k} = \frac{1}{2}(1 + (x - \Xi)((x - \Xi)^2 + \epsilon)^{-\frac{1}{2}}) \tag{4.162}$$

$$\frac{\partial \Xi_k}{\partial \xi_k} = \frac{P\xi_i^{P-1}}{Q}(\sum_{k=1}^{n_s} \xi_k^P)^{\frac{1}{Q-1}} \tag{4.163}$$

Using these values the transformed sensitivities can be calculated at every iteration and used in the optimisation.

4.5.2 Printing constraint AM0

In case of a zero degree angle filter element i can not have a density value higher than the element below. The support region now consists of only one element instead of three, so $n_s = 1$.

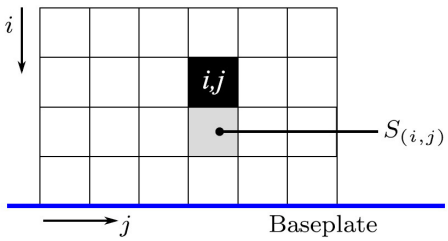


Figure 4.20: Definition of support region $S_{(i,j)}$ for element i, j in case of AM0

This printing constraint matches the current possibilities of the printer at the university in Eindhoven. The printer can print layers on top of each other, however the layers can not cantilever yet: a zero degree angle.

The method used for a 45 degree angle can be used for AM0 as well, with the difference that the elements now only have one supporting element.

4.5.3 Orientation of the baseplate

The orientation of the baseplate as defined in Tables 4.5 and 4.6 is not just of importance for the material properties, however the baseplate is also influencing the manufacturing constraints. When, for example, the baseplate is changed from the bottom to the top, different solutions will come out of the optimisation. In the algorithm the grid is rotated to make it possible to use the printing constraints in four (2D) or six (3D) directions. Because of the rectangular grid and the square (2D) or box-shaped (3D) elements the amount of possible baseplates is limited. The print direction can only be rotated with steps of 90 degrees.

Results "printed" in other directions can lead to better or stiffer results than results with the standard baseplate. It is therefore rewarding to perform the optimisation in all possible directions, to see which way of manufacturing comes up with the most optimal structure. This is assuming that the final result can be rotated after printing. In Figure 4.21 the result printed on baseplate W, which is rotated after printing, has a lower compliance value and is therefore a stiffer design than the design printed on the regular baseplate S.

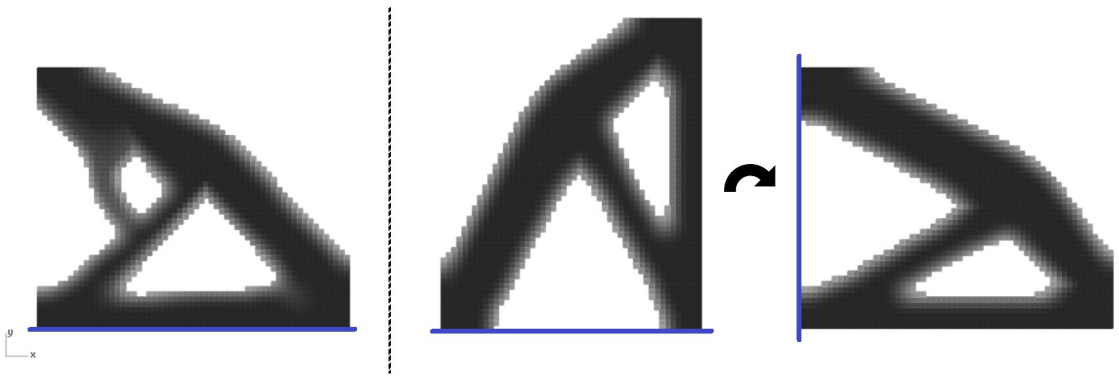


Figure 4.21: Optimised result with standard baseplate S (left) and result with baseplate W (middle) which is rotated after manufacturing (right)

4.6 Solving the optimisation problem

Now that the filtered sensitivities are known a solver can be chosen to obtain an optimal solution for the chosen objective function. The problem is non-linear and can be solved using *sequential convex approximations* such as sequential quadratic programming (SQP) and the method of moving asymptotes (MMA). These approximisation methods consider an initial design x^k, try to find a convex approximation of the original non-linear problem and use that to obtain an improved design x^{k+1}. The nature of the approximation is depending on the used method. A special case of the sequential convex approximation approach is the optimality criterion (OC) method. This method is used often in topology optimisation because it is simple to implement and because of its numerical efficiency. The method is described in this section, however it is not available in the created algorithm because of its limitations. The method can only optimise a single objective considering a single constraint. Furthermore, the SQP method is ignored in this research because a second order derivative has to be calculated for every objective function and constraint function to be investigated.

The available methods in the algorithm used to solve the non-linear problem are the MMA and the GCMMA, the globally convergent version of the MMA.

4.6.1 Optimality criterion method

The optimality criterion method is an older method than the SQP and the MMA method and was first applied to the compliance minimisation problem in 1995 by Bendsøe and Sigmund (1995).

The OC method updates the physical densities of the elements in every iteration. Convergence is achieved when the Karush-Kuhn-Tucker (KKT) condition is satisfied:

$$\frac{\partial c(\tilde{x})}{\partial x_e} + \lambda \frac{\partial v(\tilde{x})}{\partial x_e} = 0 \tag{4.164}$$

where λ is the Lagrange multiplier associated with the volume constraint $v(\tilde{x})$. This condition can be written as $B_e = 1$, where:

$$B_e = -\frac{\partial c(\tilde{x})}{\partial x_e}(\lambda \frac{\partial v(\tilde{x})}{\partial x_e})^{-1} \tag{4.165}$$

To update the densities the following OC updating scheme is proposed by Bendsoe.

$$x_e^{new} = \begin{cases} max(0, x_e - m), if\ x_e B_e^{\eta} \leq max(0, x_e - m), \\ min(1, x_e + m), if\ x_e B_e^{\eta} \geq min(1, x_e - m), \\ x_e B_e^{\eta}, \qquad\qquad otherwise, \end{cases} \tag{4.166}$$

where m is the positive move-limit and η is the numerical damping coefficient. For minimum compliance problems the choice for $m = 0.2$ and $\eta = 0.5$ is recommended by Bendsøe and Sigmund (2003).

The only unknown is now the value of the Lagrange multiplier λ for which holds that:

$$v(\tilde{x}(x_{new}(\lambda))) = 0 \tag{4.167}$$

λ can be found using the bisection method, a root-finding algorithm. It is assumed λ is between the values $l_1 = 0$ and $l_2 = 10^9$. In every step of this bisection method the central value in a domain is taken. This value is taken to create a new value for x, x^{new}. This new obtained value is subsequently used to find new values for l_1 or l_2 to come closer to the true value of λ. Once the final value is found, the new x value can be determined.

This value is then compared to the x value in the previous iteration using:

$$\|x^{new} - x\|_\infty \leq \epsilon \tag{4.168}$$

where ϵ is the tolerance which is a relatively small value, for example 0.01.

When the largest change in elemental density is smaller than this tolerance or when the maximum amount of iterations is reached, the optimisation stops and the final design is determined.

Algorithm 1 OC Method

Define initial design $x^{(k)}$, set $k \leftarrow 0$
while $\left\|x^{(k+1)} - x^{(k)}\right\|_\infty \leq \epsilon$ or $k \leq$ *maxloop* **do**
    Calculate nodal displacements **U** by performing a FEA
    Compute objective function (compliance **c**)
    Compute constraint function (volume **v**)
    Perform sensitivity analysis
    Apply the chosen filtering technique on the sensitivities and/or the design variables
    Define boundaries l_1 and l_2
    Define positive move-limit m and numerical damping coefficient η
    while $(l_2 - l_1)/(l_1 + l_2) > 10^{-3}$ **and** $l_2 > 10^{-3}$ **do**
        Compute a value for the Lagrange multiplier using the bisection method
        Update design variables (x^{new}) using the updating scheme in Equation 4.166
        Compute new boundaries l_1 and l_2
        Set $x^{(k+1)} \leftarrow x^{new}$
    Set $x^{(k)} \leftarrow x^{(k+1)}$ and $k \leftarrow k + 1$

Unfortunately, the OC method is not a very flexible method. If objectives or constraints change, the method needs to be changed. In case of multiple constraints the method cannot be used at all. For that reason two other solvers are applied in this research, the MMA and the GCMMA.

4.6.2 Method of moving asymptotes

A different method that can be used to solve the optimisation problem is the method of moving asymptotes (Svanberg 1987). The MMA is based on the convex linearisation method (CONLIN) introduced by Fleury (1989). In every iteration a convex approximation subproblem is generated and solved. This subproblem is controlled by the *moving asymptotes*. These asymptotes stabilise and fasten the convergence of the process.

The original problem is described as:

$$\begin{array}{lll} \textit{minimise} & g_0(x) & \\ s.t. & g_i \leq 0 & i = 1, ..., m \\ & x_{min} \leq x_j \leq x_{max} & j = 1, ..., n \end{array} \tag{4.169}$$

where $g_0(x)$ is the objective function and $g_i(x)$ indicates the constraint functions. These functions g are assumed to be smooth, non-linear and non-convex. Variable m is the amount of constraints and n is the number of design elements.

Non-linear problems as stated above are hard to solve directly. An approach to tackle this problem is to generate a series of sub-problems and solve these. This approach is used in the MMA. The original functions g are approximated by other functions f that are convex. The convex functions used in the MMA are described as:

$$f_i(x) = \sum_{j=1}^{n} \left(\frac{p_{ij}}{U_j^{(k)} - x_j} + \frac{q_{ij}}{x_j - L_j^{(k)}}\right) + r_i \tag{4.170}$$

The *moving asymptotes* ($U^{(k)}$ and $L^{(k)}$) and the coefficients p_{ij}, q_{ij} and r_i are updated at each iteration based on the design variables $x^{(k)}$, the real function values for that design ($g_0(x^{(k)})$ and $g_i(x^{(k)})$) and the sensitivities of these functions ($\partial g_0(x^{(k)})/\partial x^{(k)}$ and $\partial g_i(x^{(k)})/\partial x^{(k)}$). After 2 iterations the design history is used in the updating process as well.

The MMA method uses several additional variables to ensure a feasible solution, described by $y_i \geq 0$. Another variable that is described is z, which can be used to solve non-smooth problems, for example min-max problems. Equation 4.169 can now be transformed in a standard MMA sub-problem and can be described as:

$$\begin{array}{lll} \textit{minimise} & f_0(x) + z + \frac{1}{2}z^2 + \sum_{i=1}^{m}(y_i c_i + \frac{1}{2}y_i^2) & \\ s.t. & f_i(x) - a_i z - y_i \leq 0 & i = 1, ..., m \\ & \alpha_j \leq x_j \leq \beta_j & j = 1, ..., n \\ & y_i \geq 0 & i = 1, ..., m \\ & z \geq 0 & \end{array} \tag{4.171}$$

The coefficients $p_{ij}^{(k)}$, $q_{ij}^{(k)}$ and $r_i^{(k)}$ described in Equation 4.170 are expressed by Svanberg (1987) as:

$$p_{ij}^{(k)} = (U_j^{(k)} - x_j^{(k)})^2 \left(1.001 \left(\frac{\partial g_i}{\partial x_j}(x^{(k)})\right)^+ + 0.001 \left(\frac{\partial g_i}{\partial x_j}(x^{(k)})\right)^- + \frac{10^{-5}}{x_j^{max} - x_j^{min}}\right) \tag{4.172}$$

$$q_{ij}^{(k)} = (x_j^{(k)} - L_j^{(k)})^2 \left(0.001 \left(\frac{\partial g_i}{\partial x_j}(x^{(k)})\right)^+ + 1.001 \left(\frac{\partial g_i}{\partial x_j}(x^{(k)})\right)^- + \frac{10^{-5}}{x_j^{max} - x_j^{min}}\right) \tag{4.173}$$

$$r_i^{(k)} = g_i(x^{(k)}) - \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k)}}{x_j^{(k)} - L_j^{(k)}}\right) \tag{4.174}$$

in which the term $\left(\frac{\partial g_i}{\partial x_j}(x^{(k)})\right)^+$ is the largest value of 0 and $\frac{\partial g_i}{\partial x_j}(x^{(k)})$ and the term $\left(\frac{\partial g_i}{\partial x_j}(x^{(k)})\right)^-$ the largest value of 0 and $-\frac{\partial g_i}{\partial x_j}(x^{(k)})$.

$L^{(k)}$ and $U^{(k)}$ are updated iteratively to avoid large oscillation and to have a better convergence. In the CONLIN method the asymptotes are $L^{(k)} = 0$ and $U^{(k)} \to \infty$. In the MMA the asymptotes are updated following the rules below proposed by Svanberg (1987). For $k = 1$ or $k = 2$:

$$U^{(k)} + L^{(k)} = 2x^{(k)}, U^{(k)} - L^{(k)} = 1 \tag{4.175}$$

And for $k \geq 3$:

$$U^{(k)} + L^{(k)} = 2x^{(k)}, U^{(k)} - L^{(k)} = \gamma^{(k)} \tag{4.176}$$

where:

$$\gamma_i^{(k)} = \begin{cases} 0.7 & (x_j^{(k)} - x_j^{(k-1)})(x_j^{(k-1)} - x_j^{(k-2)}) < 0 \\ 1.2 & (x_j^{(k)} - x_j^{(k-1)})(x_j^{(k-1)} - x_j^{(k-2)}) > 0 \\ 1.0 & (x_j^{(k)} - x_j^{(k-1)})(x_j^{(k-1)} - x_j^{(k-2)}) = 0 \end{cases} \tag{4.177}$$

This last equation indicates that if x_i oscillates, which can be derived from the (opposite) signs of the successive iterations, that the asymptotes will be brought closer to $x_i^{(k)}$. If x_i does not oscillate, and the signs are thus the same, the asymptotes are moved away from $x_i^{(k)}$.

The asymptotes should now satisfy the following four inequalities. If this is not the case, the right hand side of the violated equation is the new value for the corresponding asymptote.

$$\begin{aligned} L_j^{(k)} &\leq x_j^{(k)} - 0.01(x_j^{max} - x_j^{min}) \\ L_j^{(k)} &\geq x_j^{(k)} - 10(x_j^{max} - x_j^{min}) \\ U_j^{(k)} &\geq x_j^{(k)} + 0.01(x_j^{max} - x_j^{min}) \\ U_j^{(k)} &\leq x_j^{(k)} + 10(x_j^{max} - x_j^{min}) \end{aligned} \tag{4.178}$$

The bounds α_j and β_j are given by:

$$\alpha_j^{(k)} = max\{x_j^{min}, L_j^{(k)} + 0.1(x_j^{(k)} - L_j^{(k)}, x_j^{(k)} - m(x_j^{max} - x_j^{min})\} \tag{4.179}$$

$$\beta_j^{(k)} = min\{x_j^{max}, U_j^{(k)} - 0.1(U_j^{(k)} - x_j^{(k)}, x_j^{(k)} + m(x_j^{max} - x_j^{min})\} \tag{4.180}$$

where m is the move-limit.

The now created regular convex sub-problem can be solved using a primal-dual method (Svanberg 1987).

Algorithm 2 Method of Moving Asymptotes

Define initial design $x^{(k)}$, set $k \leftarrow 0$
while $\left\|x^{(k+1)} - x^{(k)}\right\|_\infty \leq \epsilon$ or $k \leq maxloop$ **do**
    Calculate nodal displacements **U** by performing a FEA
    Compute objective function g_0
    Compute constraint function(s) g_i
    Perform sensitivity analysis to compute $(\partial g_0(x^{(k)})/\partial x^{(k)}$ and $\partial g_i(x^{(k)})/\partial x^{(k)})$
    Apply the chosen filtering technique on the sensitivities and/or the design variables
    Define move-limit m
    if $k = 1$ or $k = 2$ **then**
        Update $L_j^{(k)}$ and $U_j^{(k)}$ according to Equations 4.175
    else
        Update $L_j^{(k)}$ and $U_j^{(k)}$ according to Equations 4.176 and 4.177
    Check the asymptotes using Equation 4.178
    Compute the bounds α and β
    Solve the MMA-subproblem with a primal-dual method to obtain $x^{(k+1)}$
    Set $x^{(k-2)} \leftarrow x^{(k-1)}$
    Set $x^{(k-1)} \leftarrow x^{(k)}$
    Set $x^{(k)} \leftarrow x^{(k+1)}$
    $k \leftarrow k + 1$

4.6.3 Globally convergent method of moving asymptotes

The Globally Convergent Method of Moving Asymptotes or GCMMA (Svanberg 1995) is an adapted version of the MMA. While the MMA approximations are *monotonous*, the GCMMA approximations are *non monotonous*.

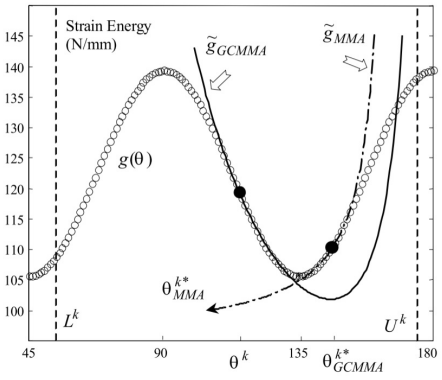


Figure 4.22: MMA and GCMMA approximations Bruyneel et al. (2002)

In a similar way as in the MMA the objective and constraint functions (g) are approximated by a function f as stated in Equation 4.170. The difference is that p_{ij} and q_{ij} are now both non-zero. Which means that both asymptotes (U_j and L_j) are used at the same time. In the MMA only one of the values is non-zero. The asymptote that is used in the MMA depends on the sign of the first order derivative $(\partial g(x^{(k)})/\partial x^{(k)})$. The use of only one asymptote leads to a monotonous approximation.

In addition to the use of two asymptotes a non-monotonous parameter $\rho_i^{(k)}$ is introduced in the calculation of the coefficients p_{ij}, q_{ij} and $r_i^{(k)}$.

$$p_{ij}^{(k,\nu)} = (U_j^{(k)} - x_j^{(k)})^2 \left(1.001 \left(\frac{\partial g_i}{\partial x_j}(x^{(k)}) \right)^+ + 0.001 \left(\frac{\partial g_i}{\partial x_j}(x^{(k)}) \right)^- + \frac{\rho_i^{k,\nu}}{x_j^{max} - x_j^{min}} \right) \tag{4.181}$$

$$q_{ij}^{(k,\nu)} = (x_j^{(k)} - L_j^{(k)})^2 \left(0.001 \left(\frac{\partial g_i}{\partial x_j}(x^{(k)}) \right)^+ + 1.001 \left(\frac{\partial g_i}{\partial x_j}(x^{(k)}) \right)^- + \frac{\rho_i^{k,\nu}}{x_j^{max} - x_j^{min}} \right) \tag{4.182}$$

$$r_i^{(k,\nu)} = g_i(x^{(k)}) - \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k,\nu)}}{U_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k,\nu)}}{x_j^{(k)} - L_j^{(k)}} \right) \tag{4.183}$$

The GCMMA method consists of *outer* iterations k and *inner* iterations ν. The process of updating the asymptotes is the same as in the MMA. The non-monotonous parameters $\rho_i^{(k)}$ are strictly positive and the starting value at a certain outer iteration k is calculated according to:

$$\rho_i^{(k,0)} = \frac{0.1}{n} \sum_{j=1}^{n} \left| \frac{\partial g_i}{\partial x_j}(x_j^{(k)}) \right| (x_j^{max} - x_j^{min}) \tag{4.184}$$

If this value is lower than 10^{-6}, $\rho_i^{(k,0)}$ is set to 10^{-6}. This value is updated in several inner iterations according to a scheme proposed by Svanberg (1995) and used to calculate an improved design.

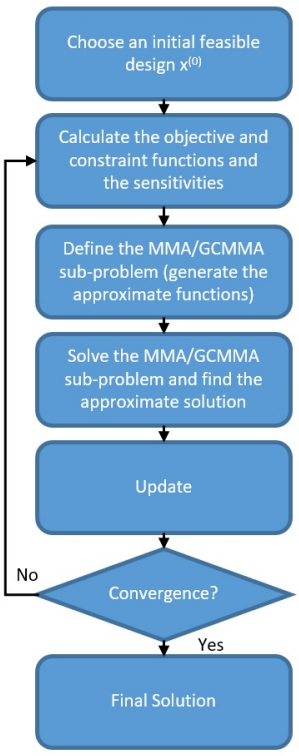


Figure 4.23: Iterative scheme of optimisation using MMA and GCMMA approximations

5 Implementation

In the previous chapters the theoretical background of the created algorithm has been given. In this chapter the implementation of this theory in the created algorithm is explained. The algorithm is a collaboration between two programs, Rhino (McNeel and Associates 2017) and Python (Python Software Foundation 2018). The problem is stated in Rhino, steered by its parametric plug-in called Grasshopper. When the problem is defined, the parameters are imported in Python in which the optimisation is being performed. The results are exported from Python and imported in Rhino again to create an appealing visualisation. The resulting models can be adapted, saved and exported to other software programs if needed.

Figure 5.1: Flowchart of the implementation

5.1 Defining the problem in Rhino/Grasshopper

At first the empty design space or domain in which the final result will fit, can be defined. The design space is created by assigning an amount of elements in the x-direction ($nelx$), in y-direction ($nely$) and in the 3D algorithm in z-direction ($nelz$). The total amount of elements is $nele$. The size of the elements (in mm) can be assigned here. A rectangular grid is now visualised in Rhino. The coordinates of the corner points, the coordinates of

the centre points and the node numbers are defined and exported. In the final step of the creation of the design space, the nodes are ”baked” in Rhino, which makes it possible to ”physically” grab some points and assign loads and boundary conditions to these nodes.

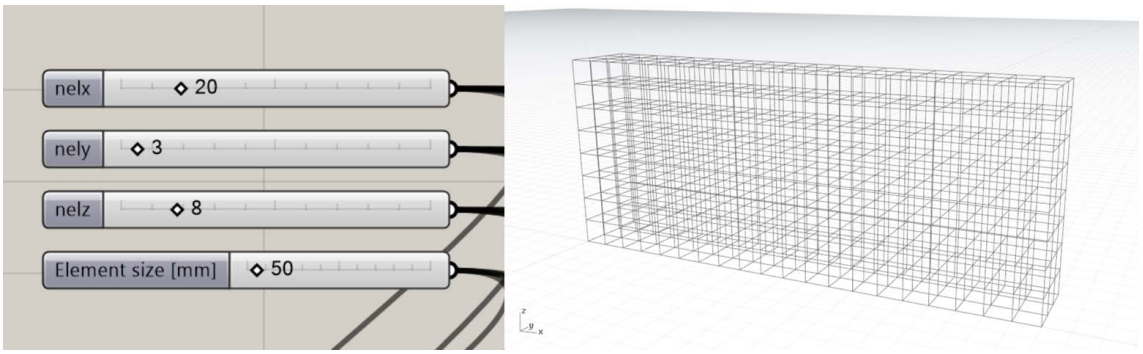


Figure 5.2: Define the 2D or 3D design space and bake the nodes in Rhino

Now that the elements and its corner points are known, the *element connectivity matrix* can be created. In this matrix all the degrees of freedom per element are stored. Every row contains the degrees of freedom of one element. The order in which the degrees of freedom are stored is based on the local node numbers. This matrix is needed to calculate the displacements.

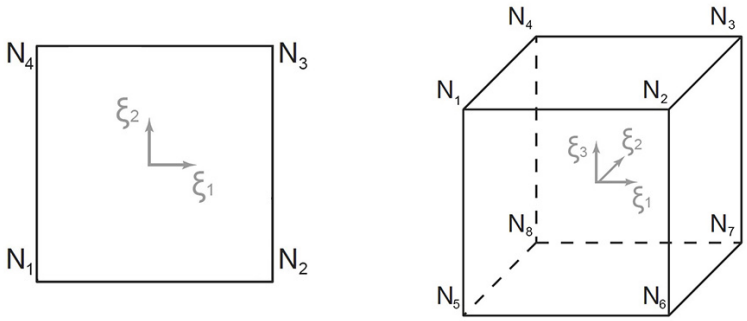


Figure 5.3: Local node numbers within a square or a cubic element

Every row of the 2D element connectivity matrix will contain 8 degrees of freedom (2 degrees of freedom per node and 4 nodes per element) and the same number of rows as there are elements. For example: if $nelx = 10$ and $nely = 6$ the matrix will look like:

$$\underbrace{\begin{bmatrix} 2 & 3 & 16 & 17 & 14 & 15 & 0 & 1 \\ 4 & 5 & 18 & 19 & 16 & 17 & 2 & 3 \\ .. & .. & .. & .. & .. & .. & .. & .. \\ 136 & 137 & 150 & 151 & 148 & 149 & 134 & 135 \\ 138 & 139 & 152 & 153 & 150 & 151 & 136 & 137 \end{bmatrix}}_{\begin{array}{cccc} Local\ Node\ 1 & Local\ Node\ 2 & Local\ Node\ 3 & Local\ Node\ 4 \end{array}} \begin{array}{l} \leftarrow element\ 0 \\ \leftarrow element\ 1 \\ \\ \leftarrow element\ 58 \\ \leftarrow element\ 59 \end{array}$$

A similar approach holds for the 3D elements, however these elements have 8 nodes per element and 3 degrees of freedom per node, so the rows will have a length of 24 instead of 8.

The elements and global nodes in 2D are ordered from top to bottom and from left to right (starting at $x = 0$, $y = y_{max}$). In 3D they are ordered from top to bottom, from left to right and from back to front (starting at $x = 0$, $y = y_{max}$ and $z = z_{max}$).

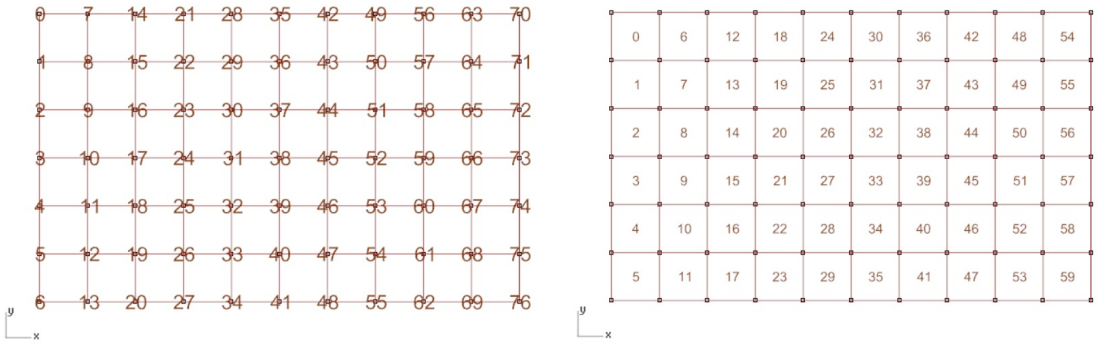


Figure 5.4: Global node numbers and element numbers in a 2D space

In Grasshopper it is possible to create passive en solid elements. The density of these elements will remain 0 or 1 during the optimisation. Solid elements are a part of the structure. Passive elements restrict the design space.

The next step in the preparation of the optimisation is the assignment of the loads and boundary conditions. In Grasshopper the forces can be assigned to the nodes as vectors. The values are stored in a force vector which is exported to be used in Python. The degrees of freedom that are either fixed or free are also stored in a vector and exported.

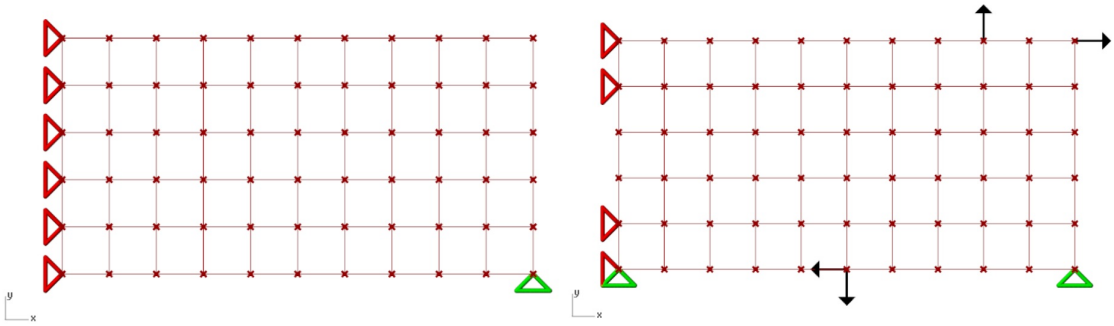


Figure 5.5: Boundary conditions and forces in Rhino

Now the elements, loads and boundary conditions are defined and can be imported in Python to run the actual optimisation.

The next step in defining the optimisation problem is the definition of the problem to be solved. It is possible to choose one of the objective/constraint-combinations described in Chapter 4.1.

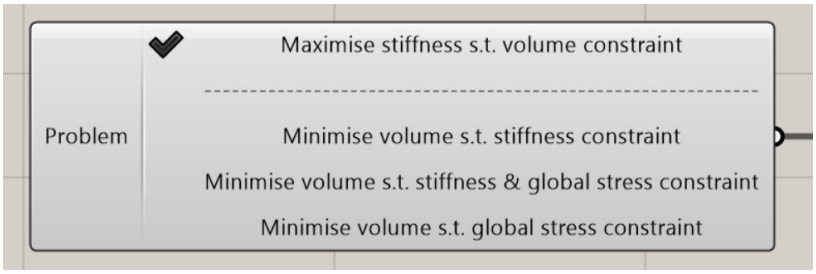


Figure 5.6: Choose the problem in Grashopper/Rhino

If a stress constraint is considered, the stress failure criterion needs to be determined.

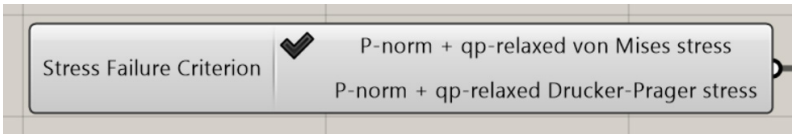


Figure 5.7: Choose the stress failure criterion to be considered in Grashopper/Rhino

Then the limit values of the stresses, compliance and/or the volume should be assigned.

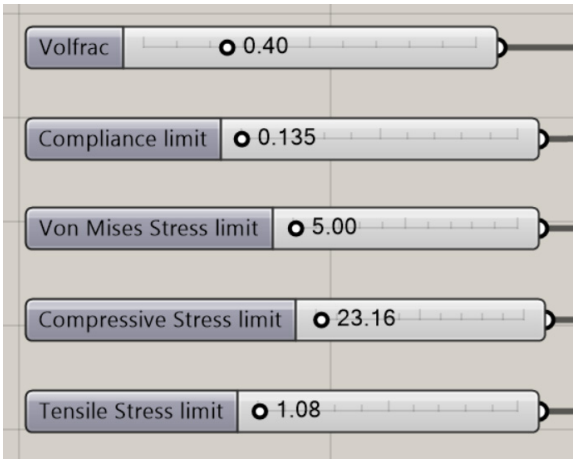


Figure 5.8: Define the limits in Grashopper/Rhino

Other important inputs for the optimisation that need to be defined, are the standard topology parameters, such as the penalisation power p, the stress relaxation factor q and the filter radius. The loop parameters that determine when the optimisation is finished are described as well.

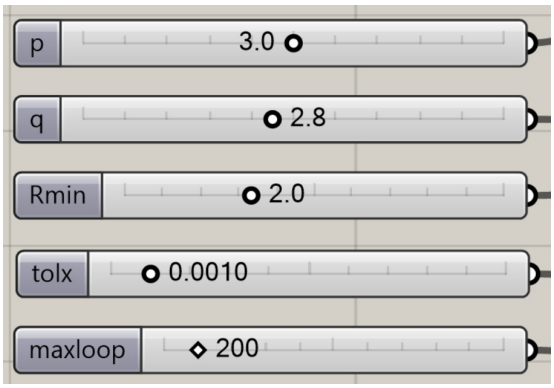


Figure 5.9: Standard topology parameters and loop parameters in Grashopper/Rhino

The preferred filter can be chosen to avoid numerical difficulties such as checker-board patterns. The solver is defined in this stage as well. Users can choose between the MMA or GCMMA.

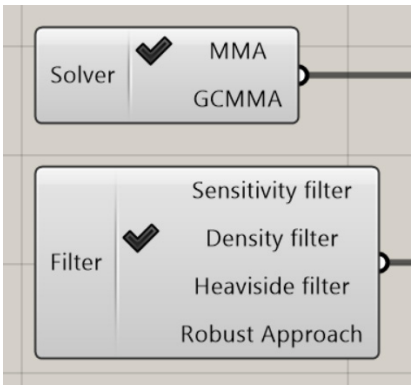


Figure 5.10: Choose the solver and the filter method

One of the last parameters that need to be determined, are the used material model and the baseplate.

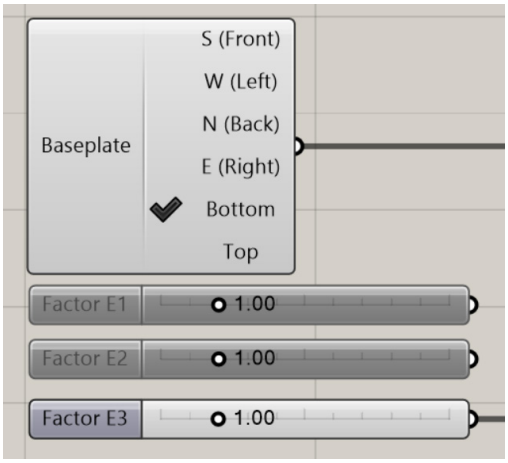


Figure 5.11: Choose the baseplate and the used material model

The baseplate is important for the material model and influences the printing constraints, so needs to be chosen carefully. If isotropic material is assumed and no manufacturing constraint is used, the baseplate is not important. The factor $E_2 = E_y$ (or $E_3 = E_z$ in 3D problems) can be changed to a value between 0 and 1 to create a lower stiffness in the weaker direction. This value is thus a factor between the stiff and the less stiff direction. The standard value for the Young's modulus is not a parameter and therefore described in Python. If the code is used for other materials this value can thus be adapted in Python.

The last choices that need to be made are the manufacturing filter and whether self-weight is considered in the optimisation.

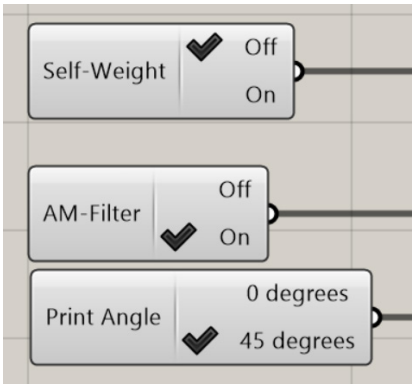


Figure 5.12: Choose the additive manufacturing filter and if self-weight is considered

In a flowchart all the steps in the definition of the design problem are summarised.

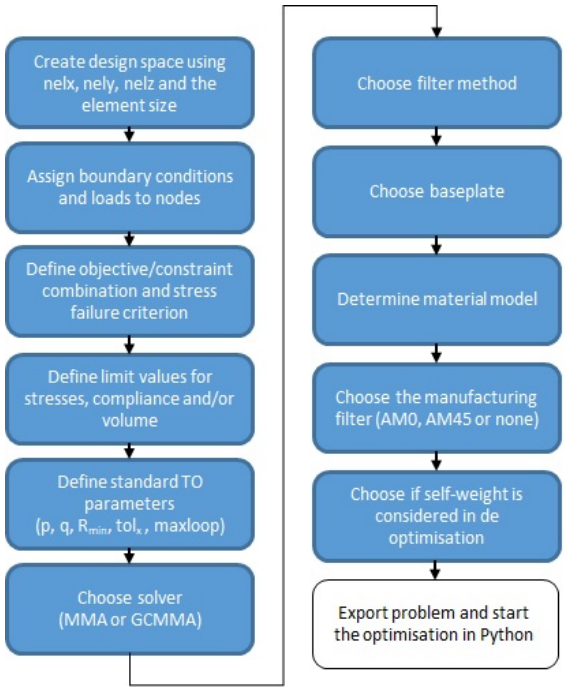


Figure 5.13: Flowchart of the definition of the design problem

5.2 Optimisation in Python

Before importing the problem defined in Rhino/Grasshopper some packages must be loaded in Python to be able to solve the problem. *NumPy* (NumPy Developers 2017) and *SciPy* (SciPy Developers 2018) are packages used for scientific computing in Python and *Sympy* (SymPy Development Team 2018) is loaded for symbolic computation. Furthermore, the *math* module is loaded for using mathematical functions and *time* is imported to keep track of the total duration of the optimisation.

Now the problem as defined in Rhino/Grasshopper is imported in Python.

```
##########################################################################################################################################
# Import Design Space, Loads and Boundary Conditions from Rhino/Grasshopper
##########################################################################################################################################
nelx            = int(np.loadtxt('nelx (GH).csv',delimiter=','))                    # Number of elements in X-direction
nely            = int(np.loadtxt('nely (GH).csv',delimiter=','))                    # Number of elements in Y-direction
nelz            = int(np.loadtxt('nelz (GH).csv',delimiter=','))                    # Number of elements in Z-direction
nele            = int(nelx*nely*nelz)                                               # Total number of elements
ele_list        = np.arange(nele)                                                   # List containing all element numbers
nnod            = (nelx+1)*(nely+1)*(nelz+1)                                        # Total number of nodes
ndof            = nnod*3                                                            # Total number of Degrees of Freedom
DOFsX           = np.loadtxt('Dofs X (GH).csv',delimiter=',')                       # (Selfweight) DOF's X
DOFsY           = np.loadtxt('Dofs Y (GH).csv',delimiter=',')                       # (Selfweight) DOF's Y
DOFsZ           = np.loadtxt('Dofs Z (GH).csv',delimiter=',')                       # (Selfweight) DOF's Z
dofsperel       = int(24)                                                           # Degrees of freedom per element (H8 elements)
IPperElement    = int(8)                                                            # Integration points per element (H8 elements)
Fex1            = np.loadtxt('Force Vector (GH).csv',delimiter=',').reshape(ndof,1) # External Load Vector
Fex2            = np.loadtxt('Force Vector 2 (GH).csv',delimiter=',').reshape(ndof,1)# External Load Vector 2
bcdof_vector    = np.loadtxt('BC Vector (GH).csv',delimiter=',')                    # Vector with DOF's that are constraint
bcdofs          = np.asarray([i for i,x in enumerate(bcdof_vector) if x == 1])      # Constraint Degrees of Freedom
freedofs        = np.asarray([i for i,x in enumerate(bcdof_vector) if x == 0])      # Free Degrees of Freedom
##########################################################################################################################################
# Import Topology Optimisation Parameters from Rhino/Grasshopper
##########################################################################################################################################
problem         = np.loadtxt('Problem (GH).csv',delimiter=',').astype(int)          # Import objective function and constraints
volfrac         = np.loadtxt('Volfrac (GH).csv',delimiter=',')                      # Import initial volume fraction
penal           = np.loadtxt('Penal (GH).csv',delimiter=',')                        # Import penalisation power
qenal           = np.loadtxt('Qenal (GH).csv',delimiter=',')                        # Import stress relaxation parameter q
ft              = np.loadtxt('Filter (GH).csv',delimiter=',').astype(int)           # Import filter
rmin            = np.loadtxt('Rmin (GH).csv',delimiter=',')                         # Import filter radius
solver          = np.loadtxt('Solver (GH).csv',delimiter=',').astype(int)           # Import solver
maxloop         = np.loadtxt('maxloop (GH).csv',delimiter=',').astype(int)          # Import maximum amount of iterations
tolx            = np.loadtxt('tolx (GH).csv',delimiter=',')                         # Import tolerance
##########################################################################################################################################
# Import Void/Solid Elements from Rhino/Grasshopper
##########################################################################################################################################
from pathlib import Path                                                            # Import Path to check if files exist
solele = Path('solidelem (GH).csv')                                                 # Path for solid elements
if solele.is_file():
    solidelements   = np.loadtxt('solidelem (GH).csv',delimiter=',').astype(int)
else:
    solidelements   = []
voiele = Path('voidelem (GH).csv')                                                  # Path for void elements
if voiele.is_file():
    voidelements    = np.loadtxt('voidelem (GH).csv',delimiter=',').astype(int)
else:
    voidelements    = []
elelist_minvoid     = np.delete(ele_list,   voidelements)                           # List with element numbers (without void elements)
nele_minvoid        = len(elelist_minvoid)                                          # Number of elements (without void elements)
##########################################################################################################################################
# Material Model, Baseplate, Manufacturing Constraints, Self-Weight & Design Limit Values
##########################################################################################################################################
E0              = float(17.336e9)   #steel = float(210e9)                           # Young's modulus of the material [N/(m^2)]
Emin            = float(1e-9*E0)                                                    # Young's modulus of zero-density elements
nu              = float(0.12)       #steel = float(0.3)                             # Unit Poisson's ratio of the material
E0_1            = float(1.0)                                                        # Young's modulus strong direction (1)
E0_2            = float(1.0)                                                        # Young's modulus strong direction (2)
E0_3            = np.loadtxt('E3 (GH).csv',delimiter=',')                           # Young's modulus weak direction (3)
##########################################################################################################################################
baseplate       = np.loadtxt('Baseplate (GH).csv',delimiter=',').astype(int)        # Import baseplate
##########################################################################################################################################
AM              = np.loadtxt('AM (GH).csv',delimiter=',').astype(int)               # Import AM filter
print_angle     = np.loadtxt('Print Angle (GH).csv',delimiter=',').astype(int)      # Import AM angle
if AM == 1 and print_angle == 0:
    eta_cont        = 0.0                                                           # Continuation
if AM == 1 and print_angle == 45:
    eta_cont        = 0.5
if AM == 0:
    eta_cont = 1.0
##########################################################################################################################################
rho0            = float(2029) #steel = float(7850)                                  # Material density[kg/m^3]
elem_size       = np.loadtxt('Element Size (GH).csv',delimiter=',')                 # Element size [m]
V_ele           = (elem_size**3)            # [m^3]                                 # Volume of an element
g               = float(9.81)               # [Nm^2/(kg^2)]                         # Gravity acceleration
selfweight      = int(np.loadtxt('Self-Weight (GH).csv',delimiter=','))             # Self-Weight considered?
max_sw_ele      = float(V_ele*rho0*g)                                               # Maximum weight per element
##########################################################################################################################################
stressmeasure   = np.loadtxt('Stress Measure (GH).csv',delimiter=',')               # Import stress measure
Aggr            = int(6)                                                            # Stress aggregation parameter P
vmslim          = np.loadtxt('Von Mises Stress Limit (GH).csv',delimiter=',')       # Von Mises Stress limit [N/(mm^2)]
vmslim          = vmslim*1e6                                                        # Von Mises Stress limit [N/(m^2)]
comslim         = np.loadtxt('Compressive Stress Limit (GH).csv',delimiter=',')     # Compressive Stress limit [N/(mm^2)]
comslim         = comslim*1e6                                                       # Compressive Stress limit [N/(m^2)]
tenslim         = np.loadtxt('Tensile Stress Limit (GH).csv',delimiter=',')         # Tensile Stress limit [N/(mm^2)]
tenslim         = tenslim*1e6                                                       # Tensile Stress limit [N/(m^2)]
clim            = np.loadtxt('Compliance Limit (GH).csv',delimiter=',')*1           # Compliance limit [Nm]
##########################################################################################################################################
```

Figure 5.14: Importing from Rhino/Grasshopper

Now that the problem is defined the iterations can be initialised. An empty displacement vector U is defined, the element connectivity matrix is imported as *edofMat* and the starting value for the densities is created: a vector x in which all the elemental densities are 1.0. Some of these variable should go to 0, some of them should stay 1.

This vector with the design variables is filtered according to the chosen filter and, if manufacturing constraints are used, filtered by the described AM filter from Section 4.5 as well. Before the iterations can start the MMA/GCMMA parameters are defined, as well as the starting value of the iteration number (0) and the starting value of the change (1.0). The empty vectors with sensitivities *dc* and *dv* and the empty vector with elemental compliances *ce* are the last vectors defined.

Now the global stiffness matrix is created based on the filtered densities $\hat{x}$ to calculate the nodal displacements. The objective value and the constraint values are now calculated. The derivatives are then filtered by the chosen filter and eventually by the AM-filter. Now depending on the chosen solver the optimisation is done.

The final step in the Python algorithm consists of exporting the calculated values. The nodal displacements, the elemental stresses and most importantly the design variables.

In the pseudo code below the algorithm in Python is summarised.

Algorithm 3 Created Topology Optimisation Algorithm

```
Import design problem from Rhino/Grasshopper
Create elemental stiffness (K_0), strain-displacement (B) and stiffness (C_0) matrices
Calculate the weight-factor (H) according to the formula in Section 4.2
Initialise the iterations
while change > tolx and k < maxloop do
    Calculate displacements using FEA
    Calculate objective function and its sensitivities
    Calculate constraint function(s) and the sensitivities
    Filter sensitivities
    Solve the optimisation problem
    Filter the design variables
Export final design, displacements and stresses
```

5.3 Visualisation in Rhino/Grasshopper

The calculated values can now be imported in Grasshopper. Depending on what the designer wants to see the results of the optimisation can be visualised. It is possible to view the virtual densities in a black and white scale, as well as in a coloured scale. It is also possible to view the stresses in the structure in all the directions.

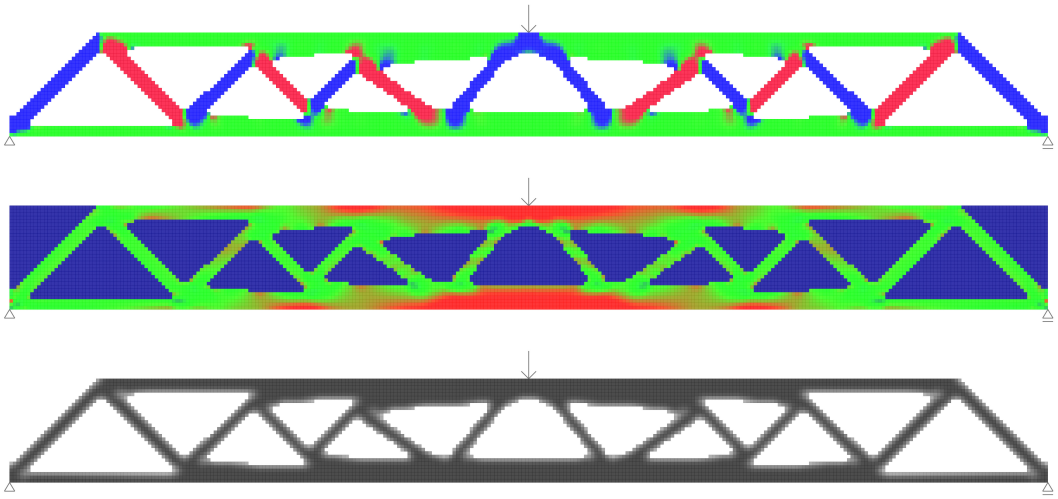


Figure 5.15: Example of the different visualisation options. Density in black and white scale (bottom), Von Mises stress in coloured scale (centre) and the stress in y-direction (top)

3D results can be smoothed, by creating an isosurface, based on the densities, and exported as 3D objects. These objects can be sliced into layers and printed.

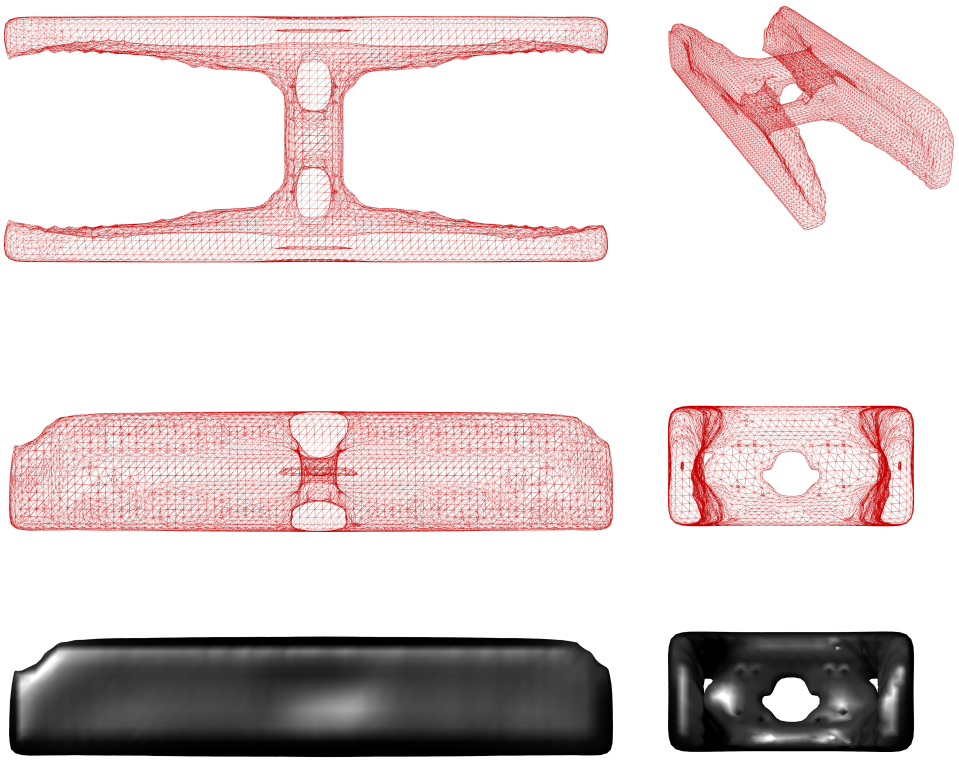


Figure 5.16: Example of a 3D object resulting from an optimisation

6

Results and verification

In this chapter the outcomes of the optimisations are presented. Some 2D results, as well as some 3D outcomes are presented and the effect of the different input parameters is discussed.

6.1 MBB-beam

The different results for different parameters are visualised and explained. The results are compared using well known examples in topology optimisation, such as the MBB-beam and the cantilever problem. Half the beam is modelled and mirrored in the y-axis (2D) or the yz-plane (3D) to obtain a design space with double the size for the same computational costs.

Figure 6.1: Half MBB beam used to compare the results of changing parameters

The nodes on the left side of the domain are all fixed in x-direction. A load is distributed over the three top nodes on the left side of the design space. The three lowest nodes on

the right side are fixed in y-direction. The forces and boundary conditions are distributed over multiple nodes to avoid stress concentrations.

6.1.1 Volume fraction

An important parameter in compliance minimisation is the maximum volume fraction or volume limit. In topology optimisation a predefined value of, for example 0.50 can be chosen for the volume fraction. This means that the algorithm tries to find an optimal topology for 50% of the initial volume, in case of compliance minimisation. In Figure 6.2 the optimal topologies are presented for changing volume fractions.

Parameter	Value
Objective function	Maximising stiffness
Constraint function	Volume
Volume limit	**0.20, 0.40, 0.60 and 0.80**
Compliance limit	-
σ_{vM} limit	-
Penalisation power p	3
Penalisation power q	2.8
Filter radius	2.5
Filter	Density
Solver	MMA
Material model	Isotropic
Manufacturing constraint	-

Table 6.1: Input parameters for comparison of different maximum volume fractions

The design is modelled using 100 elements in x-direction and 40 elements in y-direction. The design is mirrored in the y-axis, resulting in a design space consisting of 200 times 40 elements with a length of 50 mm each. The total design space is therefore 10 m x 2 m and the thickness of the design is 50 mm. The acting load is a point load of 20 kN at midspan.

The numerical results are listed in Table 6.2.

Result	Value 1	Value 2	Value 3	Value 4
Volume [-]	0.20	0.40	0.60	0.80
Compliance $[Nm]$	54.74	19.46	12.69	9.82
σ_{vM} $[N/mm^2]$	10.55	3.92	3.24	3.25
σ_{xx} $[N/mm^2]$	8.21	4.44	3.31	2.79
σ_{yy} $[N/mm^2]$	3.31	2.77	2.91	2.95
σ_{xy} $[N/mm^2]$	2.83	1.12	1.08	1.08
Mnd [-]	0.37	0.21	0.23	0.26

Table 6.2: Optimisation results for different volume fractions

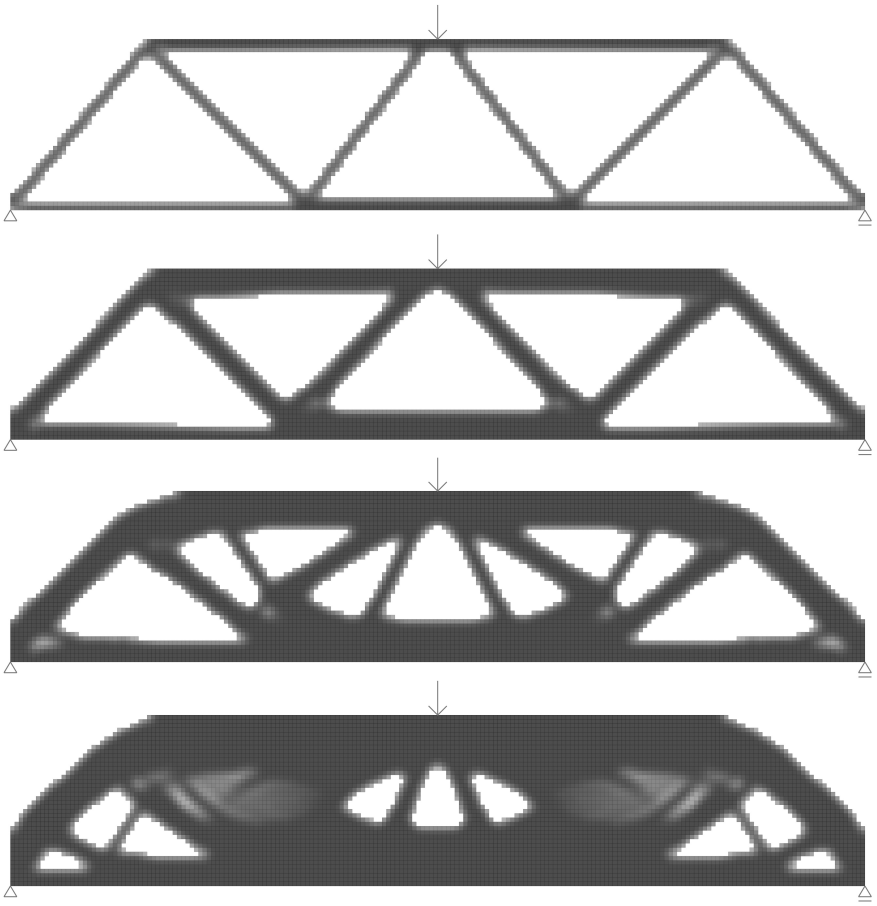


Figure 6.2: Effect of a changing volume fraction (0.20, 0.40, 0.60 and 0.80, from top to bottom)

Logically, the design becomes stiffer as the maximum volume fraction increases. Most maximum stresses decrease as well as the volume limit increases. To use this approach in an valuable manner, the designer should actually calculate the required stiffness or compliance first. Then the volume fraction should be chosen that matches this value of compliance. Obviously this is not a very efficient way, because multiple optimisations need to be performed. In the created method it is therefore possible to minimise the volume fraction given a certain maximum compliance.

In 3D the resulting structures are heavily influenced by the maximum volume as well. In the example below a design space is created with 50 elements in x-direction, 6 in y-direction and 10 in z-direction. In both x-, and y-direction symmetry is used. For clarity the colour scale is changed from black/grey to dark blue (solid elements) and light blue (intermediate elements). Elements with a virtual density below 0.5 are not visualised.

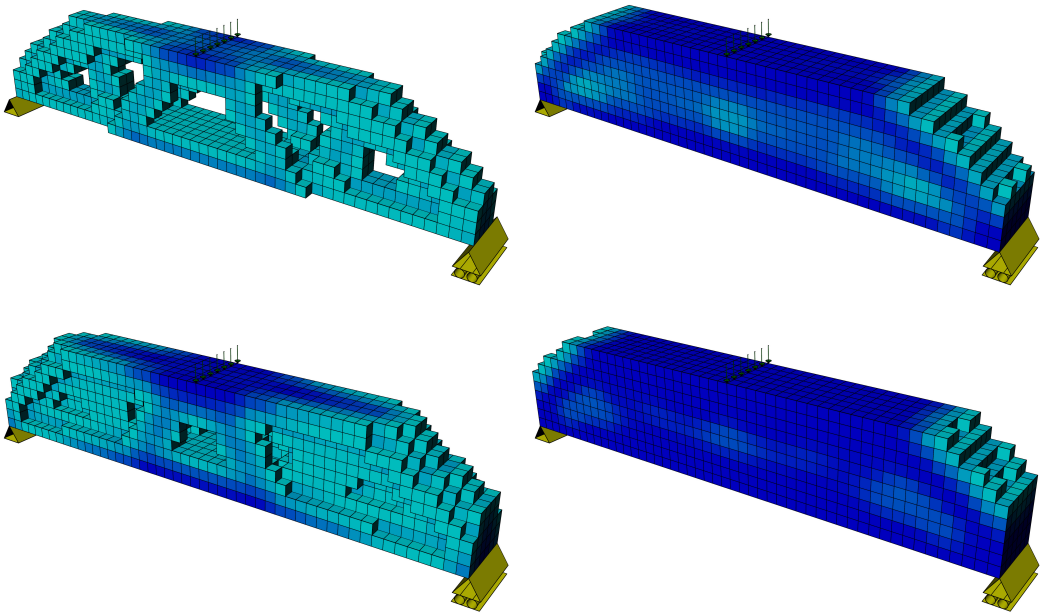


Figure 6.3: Effect of a changing volume fraction in 3D (0.2, 0.4, 0.6 and 0.8)

The 3D numerical results are listed in Table 6.3.

Result	Value 1	Value 2	Value 3	Value 4
Volume fraction [-]	0.20	0.40	0.60	0.80
Compliance $[Nm]$	60.774	12.107	5.343	3.489
$\sigma_{vM_{max}}$ $[N/mm^2]$	8.778	1.727	0.445	0.320
$\sigma_{vM_{min}}$ $[N/mm^2]$	0.007	0.006	0.002	0.007
$\sigma_{xx_{max}}$ $[N/mm^2]$	3.006	0.609	0.270	0.224
$\sigma_{xx_{min}}$ $[N/mm^2]$	-2.264	-0.561	-0.377	-0.300
$\sigma_{yy_{max}}$ $[N/mm^2]$	0.155	0.063	0.036	0.013
$\sigma_{yy_{min}}$ $[N/mm^2]$	-0.317	-0.060	-0.050	-0.025
$\sigma_{zz_{max}}$ $[N/mm^2]$	0.611	0.266	0.084	0.032
$\sigma_{zz_{min}}$ $[N/mm^2]$	-1.919	-0.562	-0.264	-0.190
$\tau_{yz_{max}}$ $[N/mm^2]$	0.450	0.087	0.033	0.014
$\tau_{yz_{min}}$ $[N/mm^2]$	-0.453	-0.092	-0.041	-0.011
$\tau_{zx_{max}}$ $[N/mm^2]$	0.187	0.058	0.042	0.037
$\tau_{zx_{min}}$ $[N/mm^2]$	-0.337	-0.066	-0.018	-0.023
$\tau_{xy_{max}}$ $[N/mm^2]$	0.919	0.425	0.203	0.107
$\tau_{xy_{min}}$ $[N/mm^2]$	-0.069	-0.016	-0.001	-0.009
$U_{z_{max}}$ $[mm]$	0.74	0.15	0.07	0.05
$U_{z_{min}}$ $[mm]$	-3.06	-0.61	-0.27	-0.18
Mnd [-]	0.77	0.69	0.60	0.58

Table 6.3: 3D optimisation results for different volume fractions

6.1.2 Penalisation power

As discussed in Chapter 4 and adapted in the method via Equation 4.46, the penalisation power is introduced to obtain more realistic designs. The penalisation factor helps to steer densities towards voids or solids and helps to avoid intermediate densities. Normally a p value of 3 is chosen in topology optimisation. The factor can vary between 1 and 4 for the best results.

Parameter	Value
Objective function	Maximising stiffness
Constraint function	Volume
Volume limit	0.50
Compliance limit	-
σ_{vM} limit	-
Penalisation power p	**1.0, 2.0, 3.0 and 4.0**
Penalisation power q	**0.8, 1.8, 2.8 and 3.8**
Filter radius	2.5
Filter	Density
Solver	MMA
Material model	Isotropic
Manufacturing constraint	-

Table 6.4: Input parameters for comparison of different values for the penalisation power

Again the design is modelled using 100 elements in x-direction and 40 elements in y-direction. The design is mirrored, resulting in a design space consisting of 200 times 40 elements with a length of 50 mm each. The total design space is therefore 10 m x 2 m and the thickness of the design is 50 mm. The acting load is a point load of 20 kN at midspan. The volume fraction is now set to 0.50.

The numerical results of a changing penalisation power are listed in Table 6.5 and the effect on the topology can be seen in Figure 6.4.

Result	p=1.0	p=2.0	p=3.0	p=4.0
Volume [-]	0.50	0.50	0.50	0.50
Compliance $[Nm]$	11.70	14.43	15.52	16.07
σ_{vM} $[N/mm^2]$	3.34	3.24	3.39	3.60
σ_{xx} $[N/mm^2]$	2.95	3.51	3.84	3.83
σ_{yy} $[N/mm^2]$	3.07	2.89	2.87	2.86
σ_{xy} $[N/mm^2]$	1.04	1.09	1.10	1.10
Mnd [-]	0.57	0.28	0.24	0.22

Table 6.5: Optimisation results for different penalisation factors

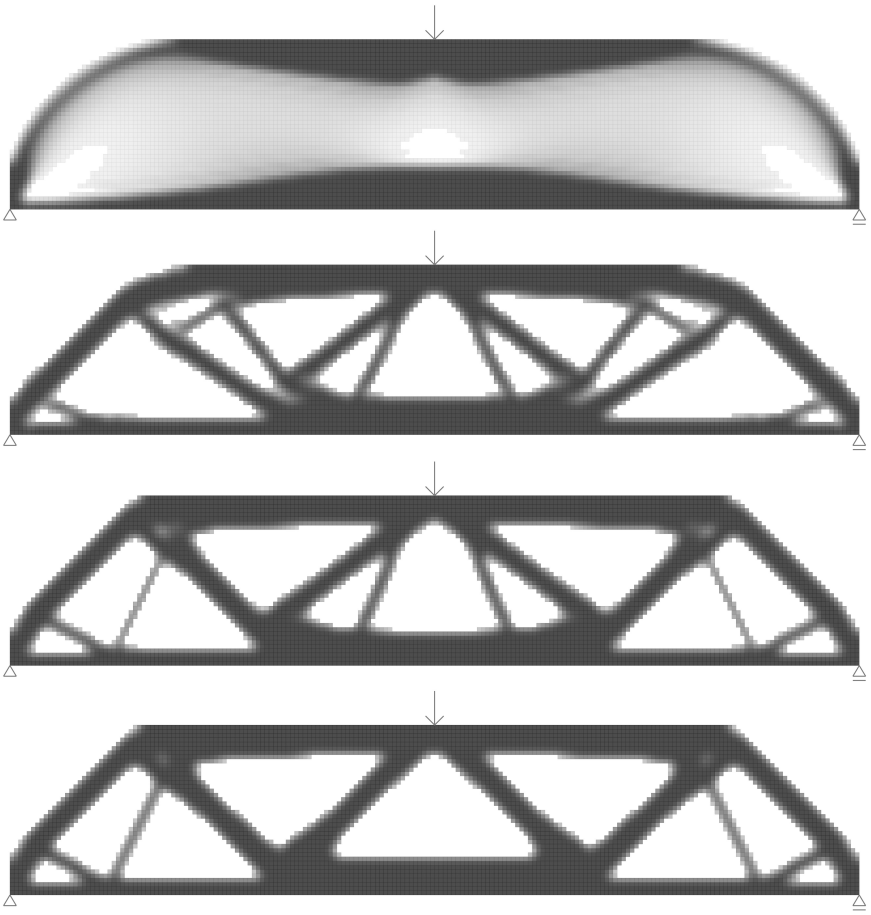


Figure 6.4: Effect of a changing penalisation power (1.0, 2.0, 3.0 and 4.0, from top to bottom)

A clear effect of the changing penalisation power is that the compliance becomes higher, when the penalisation power becomes larger. This is a result of the rejection of intermediate densities that contribute to the overall stiffness of the structure, but cannot be manufactured. The rejection of intermediate results can clearly been seen in Table 6.5, where the measure of non-discreteness is decreasing when the penalisation power increases. The variation in the stresses is minimal.

In 3D a similar result is visible. A large amount of intermediate densities (light blue) in the final result are present in case of a low penalisation power. The higher the penalisation power, the less intermediate values and the lower the stiffness.

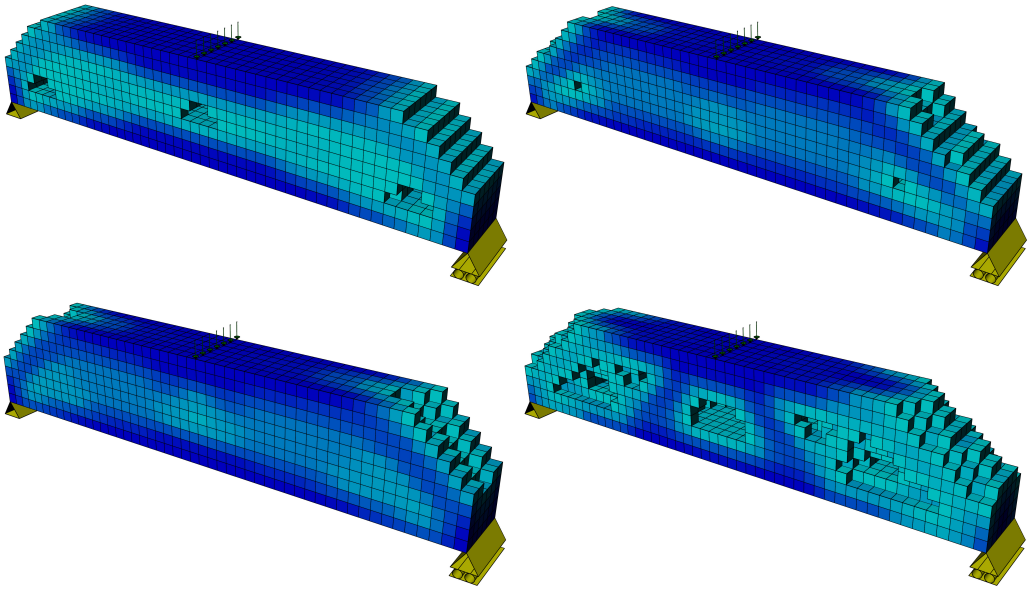


Figure 6.5: Effect of a changing penalisation power (1.0 (left top), 2.0 (left bottom), 3.0 (right top) and 4.0 (right bottom))

The 3D numerical results are listed in Table 6.6.

Result	p=1.0	p=2.0	p=3.0	p=4.0
Volume fraction [-]	0.50	0.50	0.50	0.50
Compliance $[Nm]$	4.026	5.571	7.524	9.945
$\sigma_{vM_{max}}$ $[N/mm^2]$	0.338	0.378	0.944	1.612
$\sigma_{vM_{min}}$ $[N/mm^2]$	0.012	0.010	0.002	0.003
$\sigma_{xx_{max}}$ $[N/mm^2]$	0.243	0.289	0.376	0.485
$\sigma_{xx_{min}}$ $[N/mm^2]$	-0.325	-0.375	-0.436	-0.521
$\sigma_{yy_{max}}$ $[N/mm^2]$	0.011	0.028	0.107	0.094
$\sigma_{yy_{min}}$ $[N/mm^2]$	-0.019	-0.049	-0.074	-0.059
$\sigma_{zz_{max}}$ $[N/mm^2]$	0.017	0.045	0.171	0.356
$\sigma_{zz_{min}}$ $[N/mm^2]$	-0.191	-0.221	-0.322	-0.452
$\tau_{yz_{max}}$ $[N/mm^2]$	0.005	0.027	0.047	0.093
$\tau_{yz_{min}}$ $[N/mm^2]$	-0.011	-0.029	-0.060	-0.079
$\tau_{zx_{max}}$ $[N/mm^2]$	0.036	0.041	0.050	0.047
$\tau_{zx_{min}}$ $[N/mm^2]$	-0.024	-0.023	-0.041	-0.128
$\tau_{xy_{max}}$ $[N/mm^2]$	0.109	0.158	0.301	0.395
$\tau_{xy_{min}}$ $[N/mm^2]$	-0.016	-0.014	-0.002	-0.015
$U_{z_{max}}$ $[mm]$	0.06	0.07	0.09	0.12
$U_{z_{min}}$ $[mm]$	-0.20	-0.28	-0.38	-0.50
Mnd [-]	0.66	0.65	0.64	0.64

Table 6.6: 3D optimisation results for different penalisation factors

6.1.3 Filter radius

In topology optimisation every element is considered individually, and every virtual elemental density can have a value between 0 and 1. Filters like the density filter in Equation 4.48 are used to link elements with their neighbouring elements to avoid alternating densities that lead to so called checker board patterns. The amount of elements the function considers, depends on the size of the filter radius, R_{min}. The filter radius is measured in element length.

Parameter	Value
Objective function	Maximising stiffness
Constraint function	Volume
Volume limit	0.50
Compliance limit	-
σ_{vM} limit	-
Penalisation power p	3
Penalisation power q	2.8
Filter radius	**0.5, 1.5, 2.5 and 3.5**
Filter	Density
Solver	MMA
Material model	Isotropic
Manufacturing constraint	-

The numerical results of a changing minimum radius are listed in Table 6.7.

Result	R=0.5	R=1.5	R=2.5	R=3.5
Volume [-]	0.50	0.50	0.50	0.50
Compliance $[Nm]$	13.70	14.60	15.52	15.92
σ_{vM} $[N/mm^2]$	3.40	3.28	3.39	3.36
σ_{xx} $[N/mm^2]$	3.69	3.71	3.84	3.80
σ_{yy} $[N/mm^2]$	3.10	2.91	2.87	2.86
σ_{xy} $[N/mm^2]$	1.03	1.09	1.10	1.10
Mnd [-]	0.03	0.18	0.24	0.25

Table 6.7: Optimisation results for different minimum radii

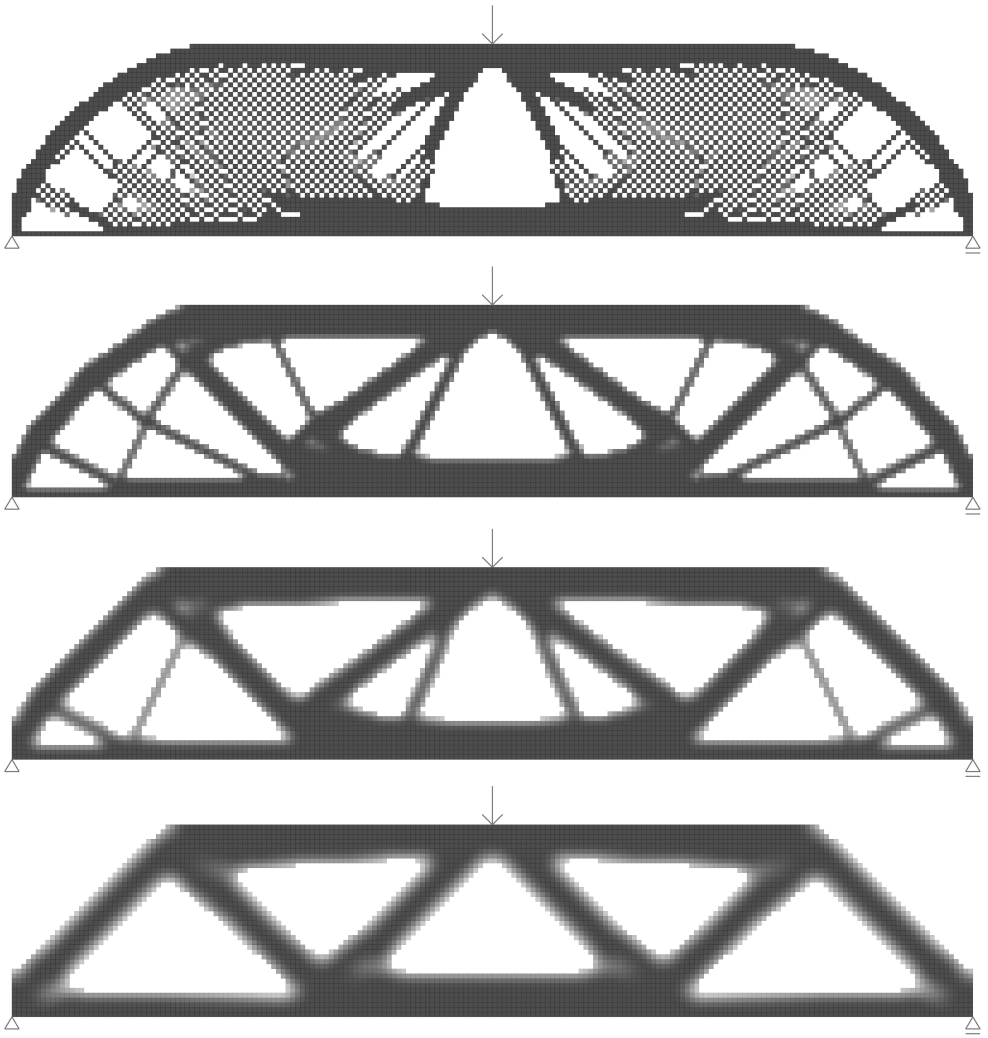


Figure 6.6: Effect of a changing filter radius (0.5, 1.5, 2.5 and 3.5, from top to bottom)

The compliance becomes lower, when the radius gets smaller, however the design becomes harder to manufacture. When the radius is too small (0.5 element lengths) the optimisation does not take into account neighbouring elements so an undesired pattern arises, like in the top design in Figure 6.6. It is important to take a certain neighbourhood into consideration, using a filter, however that area should not be too large. The larger the radius, and thus the neighbourhood, the higher the computational costs.

In 3D the effect is visible as well. Clearly the result with a filter radius of 0.5 does not take into account neighbouring elements. The result is an alternating pattern which cannot be printed by a 3D concrete printer.

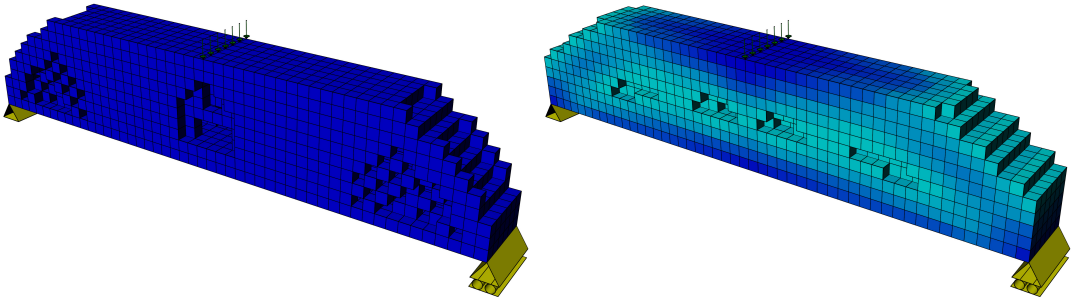


Figure 6.7: Effect of a changing filter radius (0.5 and 3.5)

The 3D numerical results are listed in Table 6.8.

Result	R=0.5	R=3.5
Volume fraction [-]	0.50	0.50
Compliance $[Nm]$	4.273	9.509
$\sigma_{vM_{max}}$ $[N/mm^2]$	0.380	1.127
$\sigma_{vM_{min}}$ $[N/mm^2]$	0.000	0.007
$\sigma_{xx_{max}}$ $[N/mm^2]$	0.256	0.504
$\sigma_{xx_{min}}$ $[N/mm^2]$	-0.368	-0.466
$\sigma_{yy_{max}}$ $[N/mm^2]$	0.030	0.018
$\sigma_{yy_{min}}$ $[N/mm^2]$	-0.032	-0.026
$\sigma_{zz_{max}}$ $[N/mm^2]$	0.041	0.131
$\sigma_{zz_{min}}$ $[N/mm^2]$	-0.186	-0.267
$\tau_{yz_{max}}$ $[N/mm^2]$	0.031	0.035
$\tau_{yz_{min}}$ $[N/mm^2]$	-0.032	-0.028
$\tau_{zx_{max}}$ $[N/mm^2]$	0.030	0.035
$\tau_{zx_{min}}$ $[N/mm^2]$	-0.023	-0.026
$\tau_{xy_{max}}$ $[N/mm^2]$	0.122	0.305
$\tau_{xy_{min}}$ $[N/mm^2]$	-0.008	-0.001
$U_{z_{max}}$ $[mm]$	0.06	0.12
$U_{z_{min}}$ $[mm]$	-0.22	-0.48
Mnd [-]	0.00	0.77

Table 6.8: 3D optimisation results for a changing filter radius

6.1.4 Material model

In this research the assumed transverse isotropic behaviour of printed concrete layers on top of each other is modelled. By changing the Young's moduli in the different directions different optimised results can be obtained. The isotropic case is visualised in the top design of Figure 6.8, the cases with a lower Young's modulus in the y-direction (or z-direction in 3D) are presented below.

Parameter	Value 1	Value 2,3 and 4
Objective function	Maximising stiffness	Maximising stiffness
Constraint function	Volume	Volume
Volume limit	0.50	0.50
Compliance limit	-	-
σ_{vM} limit	-	-
Penalisation power p	3	3
Penalisation power q	2.8	2.8
Filter radius	2.5	2.5
Filter	Density	Density
Solver	MMA	MMA
Material model	**Isotropic**	**Transverse isotropic**
E_y/E_x	**1**	**0.5, 0.1 and 0.01**
Manufacturing constraint	-	-

Table 6.9: Input parameters for comparison of different material models

The numerical results of a changing material model are listed in Table 6.10.

Result	E_y/E_x=1	E_y/E_x=0.5	E_y/E_x=0.1	E_y/E_x=0.01
Volume [-]	0.50	0.50	0.50	0.50
Compliance $[Nm]$	15.52	16.56	21.45	42.63
σ_{vM} $[N/mm^2]$	3.39	3.64	5.06	7.89
σ_{xx} $[N/mm^2]$	3.84	4.10	5.57	8.25
σ_{yy} $[N/mm^2]$	2.87	2.55	1.87	1.09
σ_{xy} $[N/mm^2]$	1.10	1.22	1.43	1.51
Mnd [-]	0.24	0.24	0.21	0.15

Table 6.10: Optimisation results for different material models

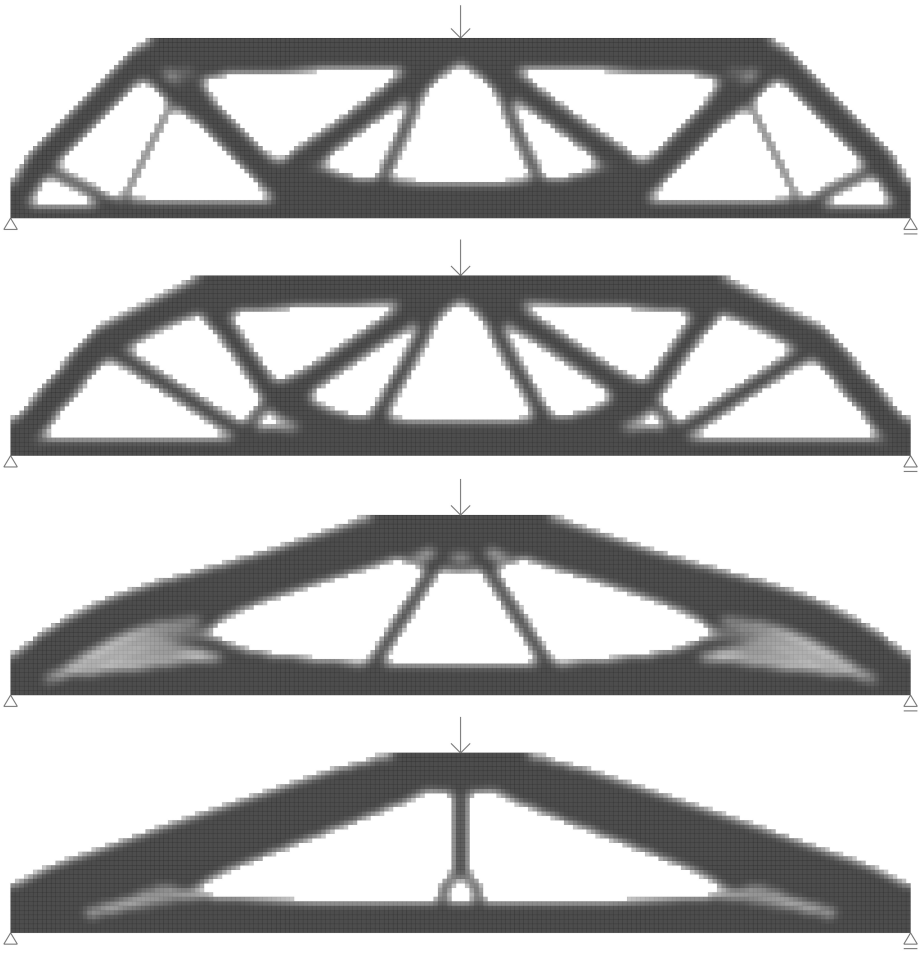


Figure 6.8: Effect of a changing material model (E_y=E_x, E_y=0.5E_x, E_y=0.1E_x and E_y=0.01E_x, from top to bottom)

A clear effect of the changing material model is that the weaker the material gets in the y-direction (or z-direction in 3D), the more the material is positioned in the stronger x-direction. The stress in x-direction becomes much larger and the compliance increases when the Young's modulus in y-direction decreases. The stress in y-direction is decreasing.

In 3D the transverse isotropic behaviour of printed concrete layers on top of each other is modelled as well. By changing the Young's moduli in the z-direction different optimised results are obtained.

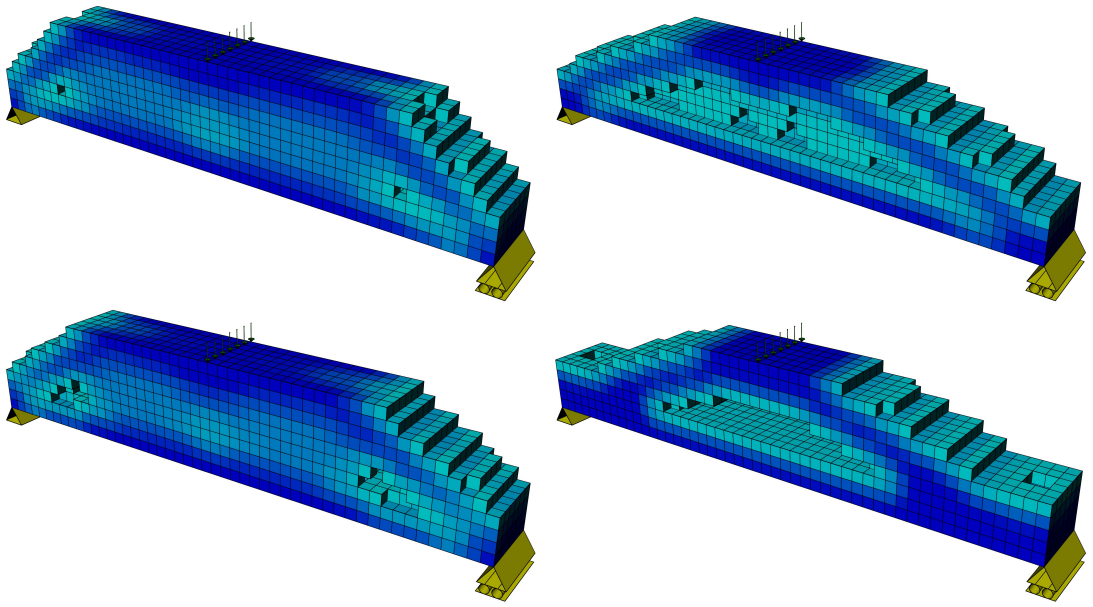


Figure 6.9: Effect of a changing material model (E_z=$E_{x,y}$ (left top), E_z=0.5$E_{x,y}$ (left bottom), E_z=0.1$E_{x,y}$ (right top) and E_z=0.01$E_{x,y}$ (right bottom))

The 3D numerical results are listed in Table 6.11.

Result	E_y/E_x=1	E_y/E_x=0.5	E_y/E_x=0.1	E_y/E_x=0.01
Volume fraction [-]	0.50	0.50	0.50	0.50
Compliance $[Nm]$	7.524	7.981	9.558	38.900
$\sigma_{vM_{max}}$ $[N/mm^2]$	0.944	1.270	0.801	1.737
$\sigma_{vM_{min}}$ $[N/mm^2]$	0.002	0.001	0.000	0.013
$\sigma_{xx_{max}}$ $[N/mm^2]$	0.376	0.364	0.333	0.000
$\sigma_{xx_{min}}$ $[N/mm^2]$	-0.436	-0.461	-0.572	-1.537
$\sigma_{yy_{max}}$ $[N/mm^2]$	0.107	0.083	0.040	0.000
$\sigma_{yy_{min}}$ $[N/mm^2]$	-0.074	-0.084	-0.114	-1.679
$\sigma_{zz_{max}}$ $[N/mm^2]$	0.171	0.182	0.017	-0.222
$\sigma_{zz_{min}}$ $[N/mm^2]$	-0.322	-0.256	-0.135	0.000
$\tau_{yz_{max}}$ $[N/mm^2]$	0.047	0.057	0.056	0.022
$\tau_{yz_{min}}$ $[N/mm^2]$	-0.060	-0.064	-0.020	-0.038
$\tau_{zx_{max}}$ $[N/mm^2]$	0.050	0.055	0.046	0.053
$\tau_{zx_{min}}$ $[N/mm^2]$	-0.041	-0.039	-0.034	-0.032
$\tau_{xy_{max}}$ $[N/mm^2]$	0.301	0.349	0.230	0.188
$\tau_{xy_{min}}$ $[N/mm^2]$	-0.002	-0.002	-0.004	-0.009
$U_{z_{max}}$ $[mm]$	0.09	0.10	0.11	0.37
$U_{z_{min}}$ $[mm]$	-0.38	-0.40	-0.48	-1.95
Mnd [-]	0.64	0.63	0.58	0.41

Table 6.11: 3D optimisation results for a changing material model

6.2 Cantilever beam

For the following parameters a different problem is solved to give a clearer comparison: the cantilever problem. The nodes on the left side of the design are all fixed in all possible directions. The load of 10 kN is assigned to the node (2D) or nodes (3D) on the bottom right. For 2D problems 100 elements are used in x-direction and 40 elements in y-direction. The element size is 50 mm.

Figure 6.10: Cantilever problem used to compare the results of changing parameters

6.2.1 Manufacturing constraints

When taking into account manufacturing constraints the optimised designs are mainly dependent on the defined baseplate. In the examples below it is clear that the optimised results are very different for the different baseplates. An optimal design without considering manufacturing constraints looks as follows:

Figure 6.11: Optimised result without considering additive manufacturing constraints

In case the print filter AM45 is applied this result will change depending on the chosen baseplate.

Parameter	Value
Objective function	Maximising stiffness
Constraint function	Volume
Volume limit	0.25
Compliance limit	-
σ_{vM} limit	-
Penalisation power p	3
Penalisation power q	2.8
Filter radius	2.5
Filter	Density
Solver	MMA
Material model	Isotropic
Manufacturing constraint	**AM45**
Baseplate	**S, W, N and E**

The numerical results of a changing baseplate are listed in Table 6.12.

Result	No AM filter	S	W	N	E
Volume [-]	0.50	0.50	0.50	0.50	0.50
Compliance $[Nm]$	14.99	15.90	14.80	17.15	15.32
σ_{vM} $[N/mm^2]$	5.27	5.27	5.24	6.04	5.27
σ_{xx} $[N/mm^2]$	3.21	3.03	3.23	3.90	3.69
σ_{yy} $[N/mm^2]$	4.35	4.35	4.32	4.34	4.36
σ_{xy} $[N/mm^2]$	1.33	1.33	1.32	1.31	1.32
Mnd [-]	0.24	0.26	0.22	0.29	0.25

Table 6.12: Optimisation results for different baseplates for AM45

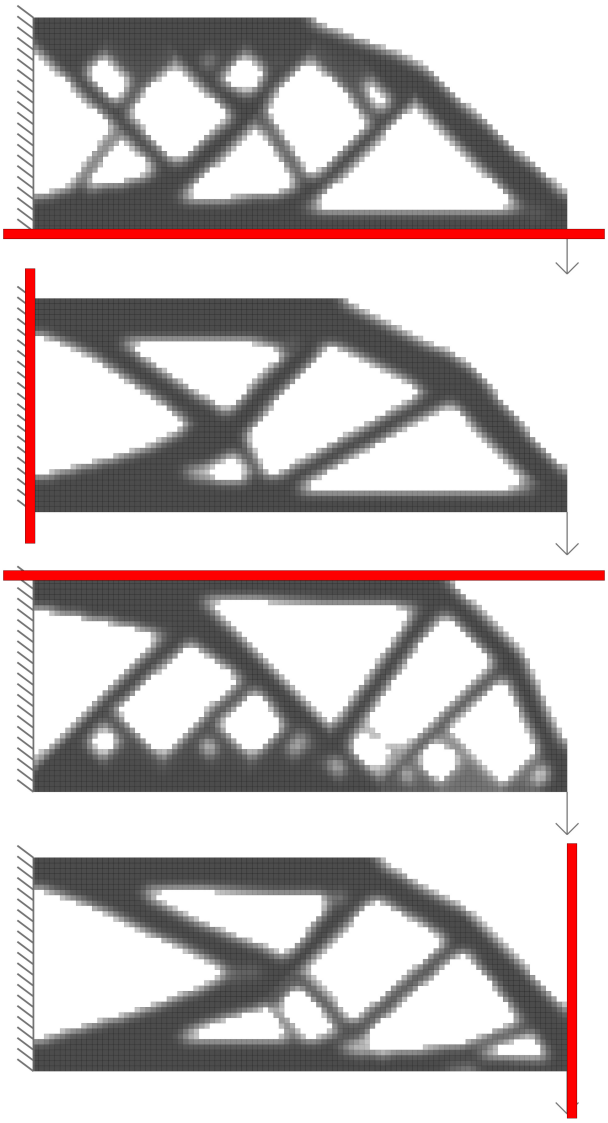
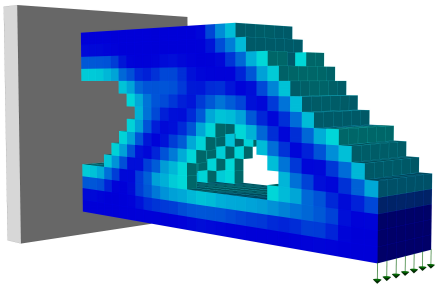


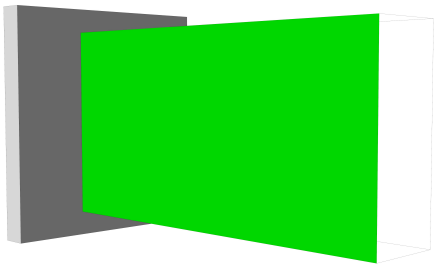
Figure 6.12: Effect of the changing baseplate for AM45 from S, W, N, E (from top to bottom)

The compliances differ significantly depending on the baseplate. In this particular case the maximum stiffness is obtained when printing from left to right, so with baseplate W. Notable is that the compliance is even smaller than the compliance without a manufacturing filter. Except for the compliances and the topologies, the results are not significantly different.

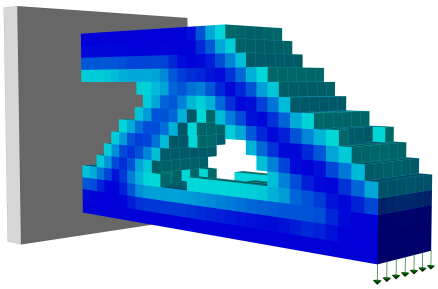
In 3D there a two additional baseplates, the top surface and the bottom surface of the design space. All these six sides of the design space can be chosen as the baseplate for the print process. After construction the design is rotated (except when the bottom surface is chosen), installed and used.



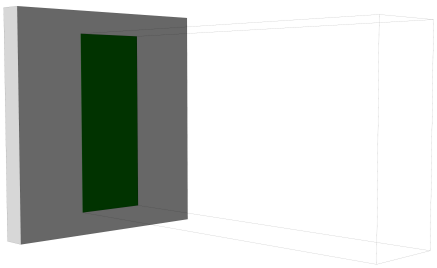
(a) Resulting topology with baseplate S



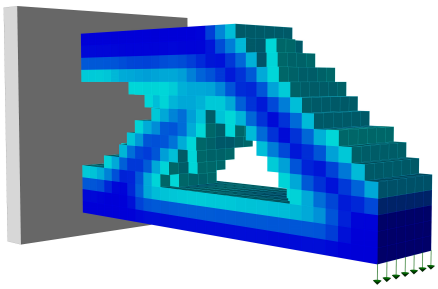
(b) Baseplate S



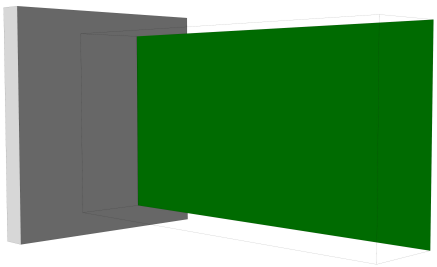
(c) Resulting topology with baseplate W



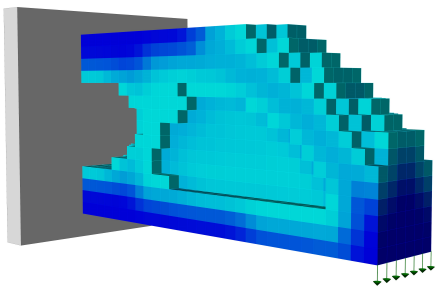
(d) Baseplate W



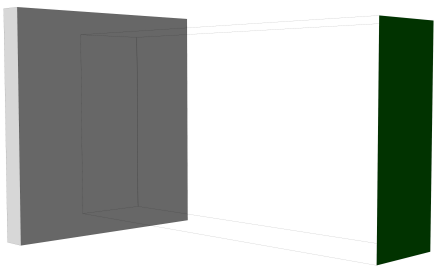
(e) Resulting topology with baseplate N



(f) Baseplate N

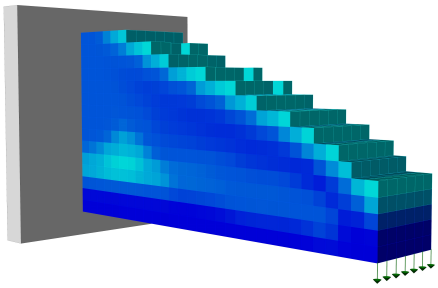


(g) Resulting topology with baseplate E

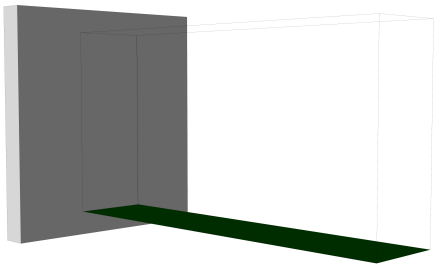


(h) Baseplate E

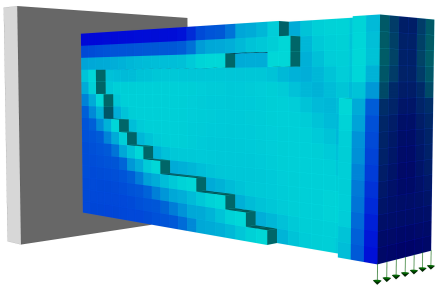
Figure 6.13: Effect of a changing baseplate for AM45 (S, W, N, E)



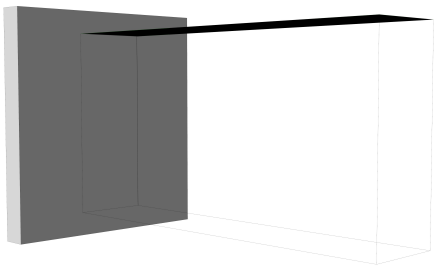
(a) Resulting topology with baseplate B



(b) Baseplate B



(c) Resulting topology with baseplate T



(d) Baseplate T

Figure 6.14: Effect of a changing baseplate for AM45 (B and T)

Switching the baseplates results in very different optimal designs. Because the design space is relatively small in y-direction not all results are considerably different.

6.2.2 Filtering method

As explained in Section 4.2 there are different methods of filtering the densities and sensitivities used in the optimisation. This filtering is needed to avoid numerical difficulties such as mesh-dependency, checker-board patterns and local minima (Bendsøe and Sigmund 2003). The methods used in this research are the density filter, the sensitivity filter, the volume preserving Heaviside filter and the robust filter.

Parameter	Value
Objective function	Maximising stiffness
Constraint function	Volume
Volume limit	0.50
Compliance limit	-
σ_{vM} limit	-
Penalisation power p	3
Penalisation power q	2.8
Filter radius	2.5
Filter	**Sensitivity, Density, Heaviside, Robust**
Solver	MMA
Material model	Isotropic
Manufacturing constraint	-
Baseplate	-

The numerical results of a changing baseplate are listed in Table 6.13.

Result	Sensitivity	Density	Heaviside	Robust
Volume [-]	0.50	0.50	0.50	0.50
Compliance $[Nm]$	14.31	14.99	13.25	13.98
σ_{vM} $[N/mm^2]$	5.27	5.27	5.26	5.27
σ_{xx} $[N/mm^2]$	2.88	3.21	2.83	3.12
σ_{yy} $[N/mm^2]$	4.35	4.35	4.37	4.35
σ_{xy} $[N/mm^2]$	1.33	1.33	1.31	1.33
Mnd [-]	0.18	0.24	0.01	0.05

Table 6.13: Optimisation results for different filtering techniques

Different filtering methods result in different designs. Clearly visible in Figure 6.15 is that the Heaviside and robust filtering methods result in a more black and white structure than the sensitivity and the density filter. This is desirable if the design needs to be printed afterwards. The *measure of non-discreteness* is a value that indicates the degree to which a design is black-and-white (Hofmeyer et al. 2017).

$$M_{nd} = \sum_{e=1}^{n} \frac{(\frac{1}{f(1-f)})\tilde{x}_e(1-\tilde{x}_e)}{n} 100\% \tag{6.1}$$

where f is the design volume fraction. 0% means that the design is fully discrete and thus only black and white elements exist. 100% means a fully homogeneous distribution. By

comparing this value for the four different filtering techniques it can be proved that the Heaviside and robust filter come up with designs that are more black-and-white than the sensitivity and density filters.

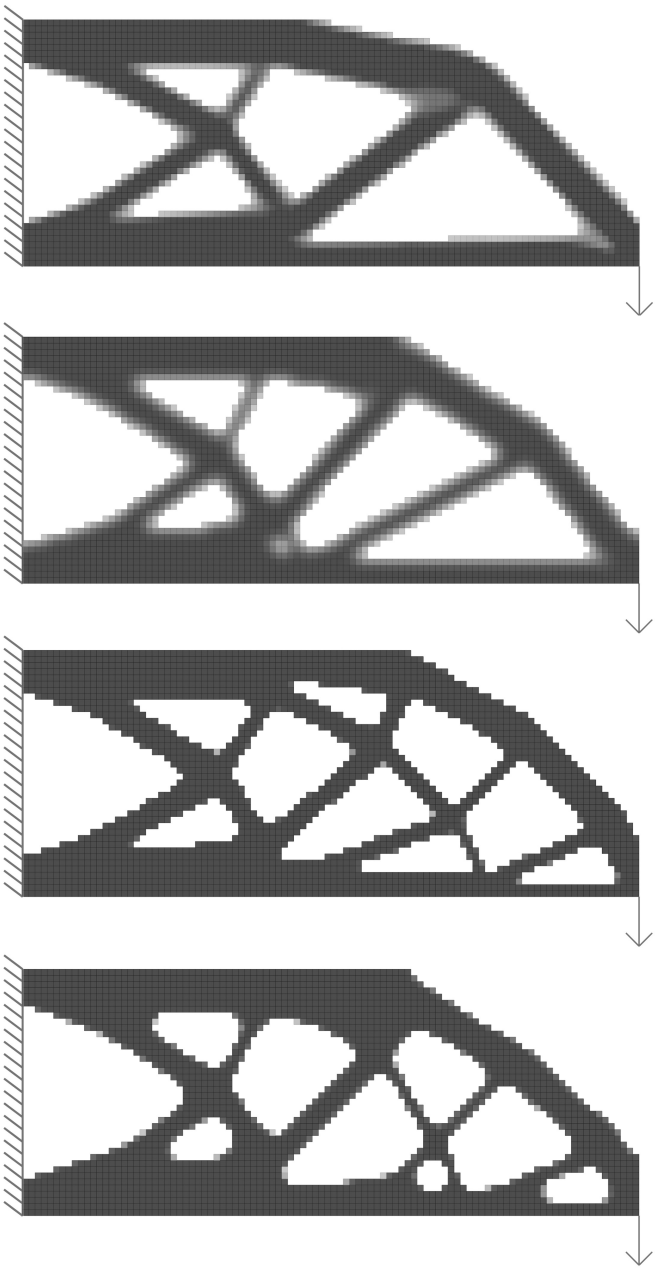
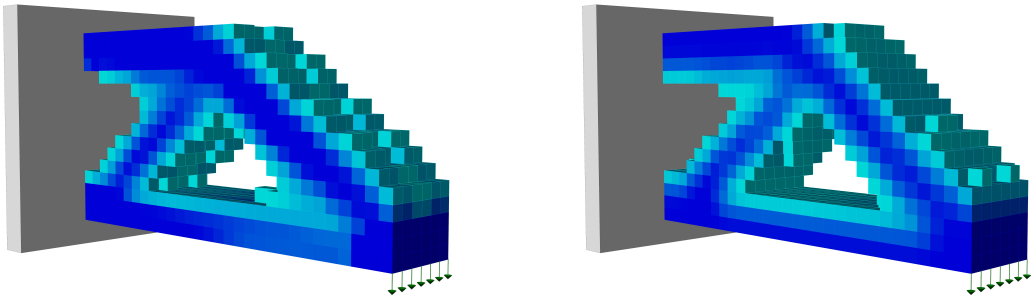


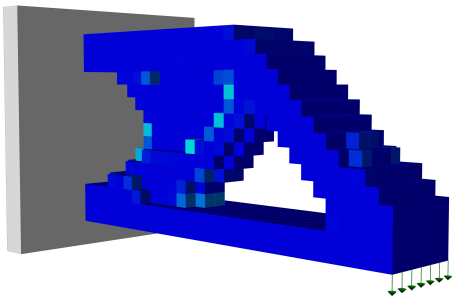
Figure 6.15: Different topologies due to the different filtering methods used (sensitivity, density, Heaviside and robust filtering, from top to bottom

In 3D the filters can be applied as well. Elements with a density near 1.0 are visualised in dark blue, the intermediate results are light blue and the elements below 0.5 are removed from the design. Again it is very clear that the use of the Heaviside and robust filter lead to more black-and-white solutions.

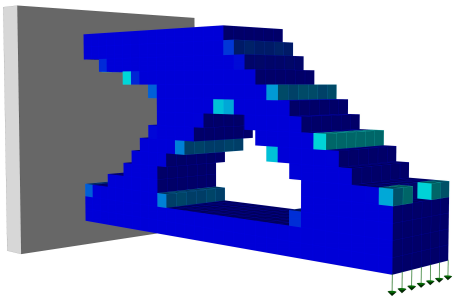


(a) Sensitivity filter in 3D

(b) Density filter in 3D



(c) Heaviside filter in 3D



(d) Robust filter in 3D

Figure 6.16: Effect of a different filtering techniques in 3D

6.2.3 Problem definition

When changing the objective function and the constraints, the optimised results are different. To make a fair comparison first the volume minimisation with stress constraint is executed. The resulting volume is than inserted as the constraint for the compliance minimisation.

Parameter	Value 1	Value 2
Objective function	**Minimising compliance**	**Minimising volume**
Constraint function	**Volume**	**Von Mises stress limit**
Volume limit	**0.26**	-
Compliance limit	-	-
σ_{vM} limit	-	**5.50**
Penalisation power p	3	3
Penalisation power q	2.8	2.8
Filter radius	2.5	2.5
Filter	Density	Density
Solver	MMA	MMA
Material model	Isotropic	Isotropic
Manufacturing constraint	-	-

The numerical results of a changing problem definition are listed in Table 6.14.

Result	Minimum compliance	Minimum volume
Volume [-]	0.26	0.26
Compliance $[Nm]$	33.39	38.41
σ_{vM} $[N/mm^2]$	8.31	5.50
σ_{xx} $[N/mm^2]$	6.26	5.51
σ_{yy} $[N/mm^2]$	4.29	4.30
σ_{xy} $[N/mm^2]$	1.64	1.87
Mnd [-]	0.29	0.39

Table 6.14: Optimisation results for different objectives and constraints

Clearly visible in the stress plots (top figures) in Figure 6.17 is that if the stress constraint is applied almost all members are fully stressed. The volume fraction of both designs is about 26%, however de left design is stiffer and the right design contains a lower maximum stress.

An observed problem is that the code sometimes is not able to find an optimum in case of the stress constraint. The stress limit should therefore be chosen carefully. Another noteworthy aspect that needs to be mentioned is that the stress constrained optimisation mostly needs more iterations to reach an optimum. This leads to a more calculations and therefore a larger computational time.

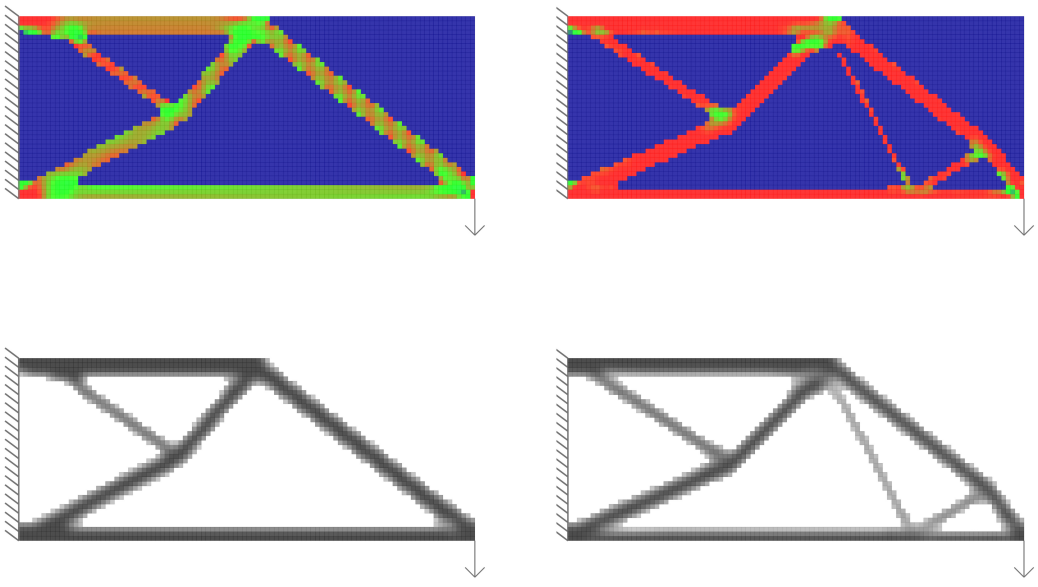


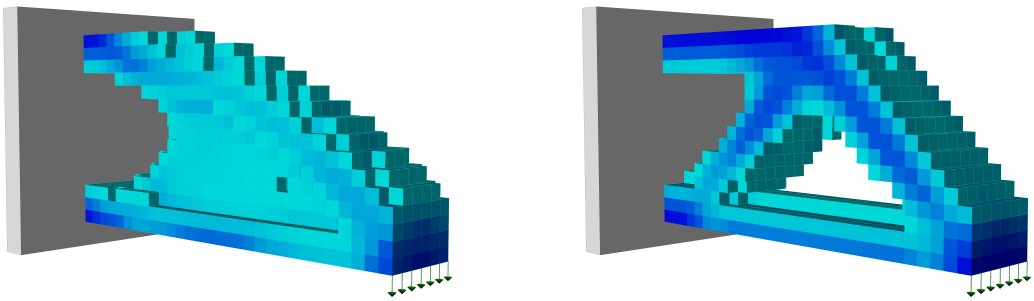
Figure 6.17: Difference in minimising the compliance with a volume constraint (left) and minimising the volume with a von Mises stress constraint (right)

In 3D the choice for an objective function and the constraint function(s) is even better visible. To compare the results, again first the minimising volume problem is conducted.

The volume fraction resulting from this problem (31 %) is used as the limit for the compliance minimisation.

Parameter	Value 1	Value 2
Objective function	**Minimising compliance**	**Minimising volume**
Constraint function	**Volume**	**Von Mises stress limit**
Volume limit	0.31	-
Compliance limit	-	-
σ_{vM} limit	-	5.50
Penalisation power p	3	3
Penalisation power q	2.8	2.8
Filter radius	1.5	1.5
Filter	Density	Density
Solver	MMA	MMA
Material model	Isotropic	Isotropic
Manufacturing constraint	-	-

If the problem (a combination of objective and constraints) is different it can be seen that the results of the optimisation are different as well. Figure 6.18 shows two completely different material distributions.



(a) Minimising compliance in 3D *(b) Minimising volume in 3D*

Figure 6.18: Effect of different problem definitions in 3D

6.3 Tension and/or compression structure

For the last verification a square (2D) or box shaped (3D) design space is chosen with a downward pointing load at the centre of this design space. The corners of the design space are fixed in all possible directions. The design space in 2D is 2 m x 2 m and consists of 40 x 40 = 1600 elements with a length of 50 mm. The load is 20 kN.

6.3.1 Stress failure criterion

To avoid large tensile forces in the final structure the stress failure criterion to be considered can be changed from the von Mises stress to the *Drucker-Prager equivalent stress*.

Parameter	Value 1	Value 2
Objective function	Minimising volume	Minimising volume
Constraint function	**Von Mises stress limit**	**Drucker-Prager stress limit**
Volume limit	-	-
Compliance limit	-	-
σ_{vM} limit	**3.00**	-
σ_{Lt} limit	-	**1.00**
σ_{Lc} limit	-	**5.00**
Penalisation power p	3	3
Penalisation power q	2.8	2.8
Filter radius	2.5	2.5
Filter	Density	Robust
Solver	MMA	MMA
Material model	Isotropic	Isotropic
Manufacturing constraint	-	-
Baseplate	-	-

The numerical results of a changing stress constraint are listed in Table 6.15.

Result	Von Mises stress	Drucker-Prager stress
Volume [-]	0.13	0.11
Compliance $[Nm]$	12.82	5.92
σ_{vM} $[N/mm^2]$	6.04	6.04
σ_{xx} $[N/mm^2]$	3.48	3.76
σ_{yy} $[N/mm^2]$	3.68	3.61
σ_{xy} $[N/mm^2]$	3.26	2.77
Mnd [-]	-	-
Time [s]	15	528

Table 6.15: Optimisation results for different stress constraints

To compare, the densities, the von Mises stress and the stresses in the weaker y-direction are visualised in Figure 6.19.

In case a different stress criterion is chosen as constraint for the volume minimisation, the topology changes. If possible a structure is found that only contains compressive internal forces. If not, the tensile stresses are diminished by adding more material in the tensile areas. In this case it was possible to reduce the stresses in y-direction to a minimum. Only compressive stresses are present in the final result (left figures in Figure 6.19).

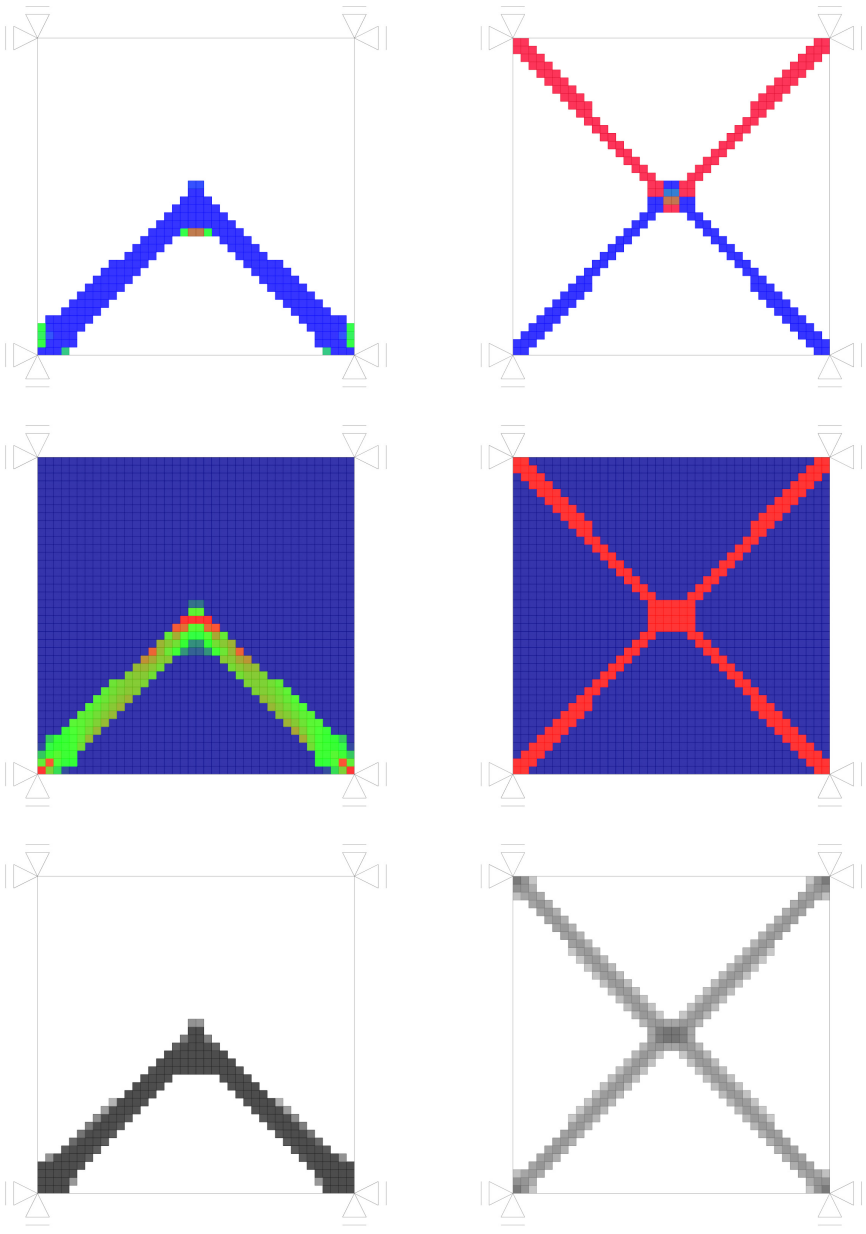


Figure 6.19: Results for the Drucker-Prager stress (left) and the von Mises stress criterion (right) (σ_{yy}, von Mises stress and densities, from top to bottom)

An important difference of stress constraint optimisation (and especially using the Drucker-Prager stress) compared to compliance or volume constraint optimisation is the enormous increase in computational time. The results above are calculated in *15* seconds (von Mises stress constraint) and *528* seconds (Equivalent Drucker-Prager stress

constraint). These values are both larger than the *5* seconds needed to solve the same problem subjected to a compliance constraint.

In 3D the same problem is created, however now with the eight corner points constrained instead of four. The design space is box, instead of a 2D square. The optimisations took several hours.

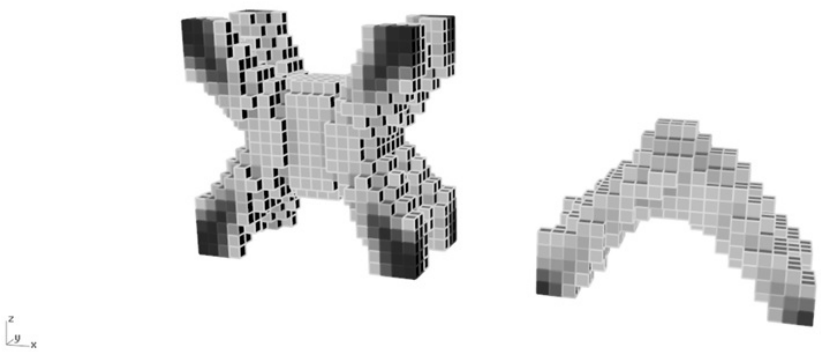


Figure 6.20: 3D example of the difference in stress failure constraint (left: von Mises stress criterion, right: Drucker-Prager equivalent stress)

7

Case Study

In the previous chapters the theory, the implementation of that theory and the results of the created algorithm are described in detail. To investigate what the possible benefits of the method can be for construction a case study is performed in this chapter.

7.1 Floor slab design

An example of a structural part that is often created in concrete is a floor. Floors are often made of poured concrete in a rectangular mould due to the simplicity. The top surface is subjected to a distributed load. An idea for optimisation is to print the floor upside down and rotate it afterwards. This idea results from the fact that the top layer must be a part of the structure, because that is where the distributed load is acting on.

To compare the results of the optimisation with real values, the theory about thin floor slabs is used and briefly explained below. A slab is a plate loaded perpendicular to its plane. We can assume a thin floor because the thickness to width ratio is (lower than or) equal to 1/10. We consider the floor to be thin, so we can assume that the shear deformation is negligibly small. According to the Kirchhoff–Love theory (Love 1888), an extension of the Euler-Bernouilli beam theory, flat cross sections remain flat after loading. And deflections are not depending on the z-coordinate.

The considered floor has a length of 4.0 meters in x-direction and 3.0 meters in y-direction. The thickness of the slab is $t = 300\ mm$. And we consider a distributed load $p = 5\ kN/m^2$. The self-weight is added to this value: $p_{total} = p + p_{self}$. In this formula p_{self} depends on the thickness, the gravity acceleration ($9.81\ m^3s^{-2}kg^{-1}$) and the density ($2029\ kg/m^2$). The formula for the self-weight part becomes $p_{self} = \rho tg$. The total load is now $p_{total} = 10.97\ kN/m^2$. The Young's modulus is $E = 17336\ N/mm^2$ according to Section 4.3 and the Poisson's ratio $\nu = 0.12$.

In slabs different relations can be described. The relations between displacements, deformations, stresses/moments and loads are called the kinematic, constitutive and equilibrium equations.

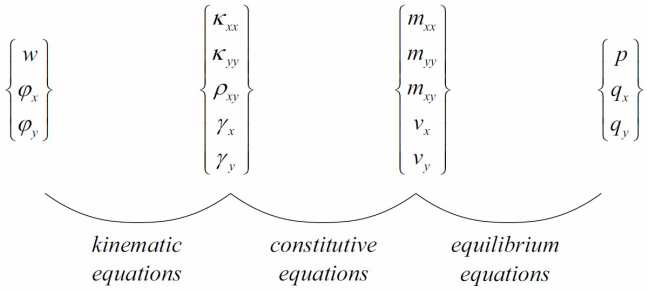


Figure 7.1: Relation scheme plates

Because the shear deformations (γ_x and γ_y) are considered zero in thin plates, the relation scheme can be altered.

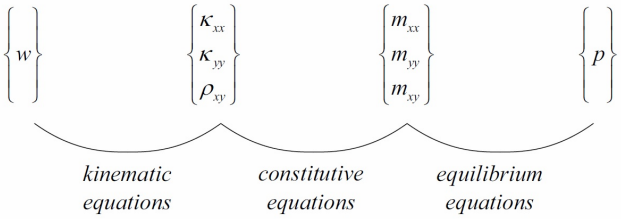


Figure 7.2: Relation scheme thin plates

In Figure 7.3 the relations between stresses/moments and deformations are visualised.

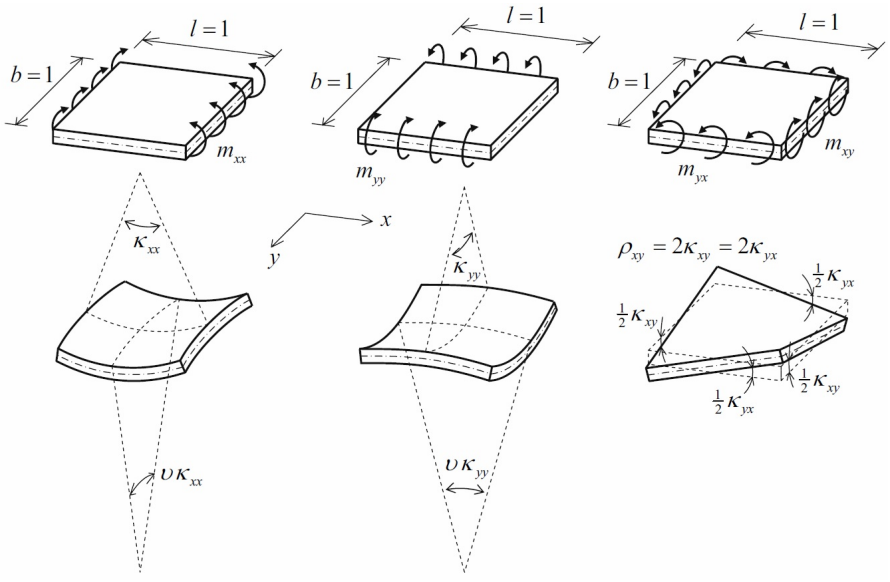


Figure 7.3: Stress resultants and deformations due to bending in a plate with lateral contraction

According to the theory the kinematic equations, the relations between the displacements and the deformations, are given by:

$$\begin{aligned} \kappa_{xx} &= -\frac{\delta^2 w}{\delta x^2} \\ \kappa_{yy} &= -\frac{\delta^2 w}{\delta y^2} \\ \rho_{xy} &= 2\kappa_{xy} = -2\frac{\delta^2 w}{\delta x \delta y} \end{aligned} \tag{7.1}$$

The constitutive equations, relating the deformations and the stresses are given by:

$$\begin{aligned} m_{xx} &= D(\kappa_{xx} + \nu\kappa_{yy}) \\ m_{yy} &= D(\nu\kappa_{xx} + \kappa_{yy}) \\ m_{xy} &= \frac{1}{2}(1-\nu)D\rho_{xy} \end{aligned} \tag{7.2}$$

Where $D = \frac{Et^3}{12(1-\nu^2)}$. The equilibrium equation, which is the relation between the stresses and the external distributed load p can be derived from Figure 7.4.

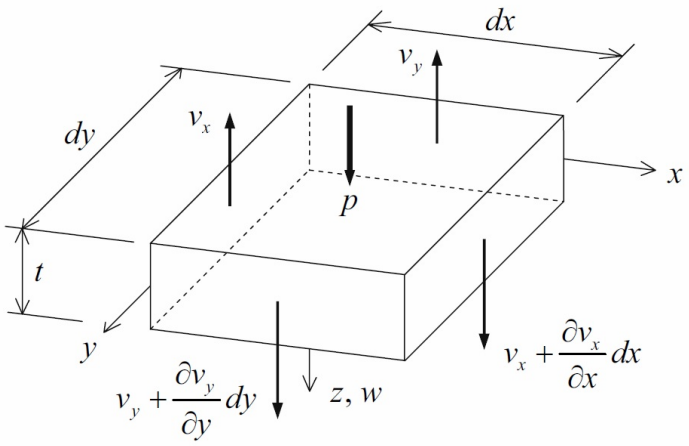


Figure 7.4: Plate equilibrium in z-direction

Coming from the equilibriums in x- and y-direction it can be concluded that:

$$v_x = \frac{\partial m_{xx}}{\partial x} + \frac{\partial m_{yx}}{\partial y} \quad ; \quad v_y = \frac{\partial m_{yy}}{\partial y} + \frac{\partial m_{xy}}{\partial x} \tag{7.3}$$

Therefore, the equilibrium in z-direction can be written as:

$$-\left(\frac{\partial^2 m_{xx}}{\partial x^2} + 2\frac{\partial^2 m_{xy}}{\partial x \partial y} + \frac{\partial^2 m_{yy}}{\partial y^2}\right) = p \tag{7.4}$$

Substitution of Equations 7.1, 7.2 and 7.4 delivers a partial differential equation in terms of w:

$$\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{p}{D} \tag{7.5}$$

This differential equation is known as the biharmonic plate equation. To solve this equation a displacement field $w(x, y)$ must be found that matches the boundary conditions and the plate equation.

To compare several floors and to come up with different topologies, different boundary conditions are imposed. Design 1A is a simply supported floor on all outer edges. Design 1B is supported on the two short edges.

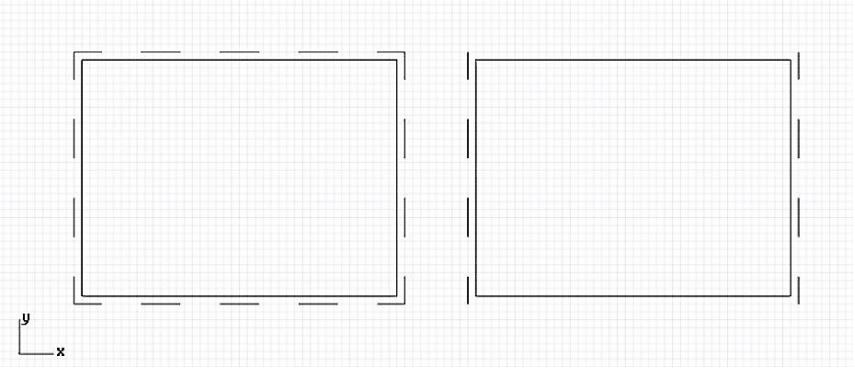


Figure 7.5: Different boundary conditions for designs 1A (left) and 1B (right)

7.1.1 Design 1A - Uniformly loaded plate, simply supported on all edges

Levy (1899) suggested to use the following equation to find the solution to the differential equation from Equation 7.5:

$$w = \sum_{m=1}^{\infty} Y_m \sin\left(\frac{m\pi x}{a}\right) \tag{7.6}$$

where Y_m is a function of y, a is the length of the plate in x-direction and m is an integer that ensures the boundary conditions at $x = 0$ and at $x = a$ are satisfied. Because the deflection and the moments are equal to 0 at these positions it holds (for $x = 0$ and $x = a$) that:

$$\begin{aligned} w &= 0 \\ \frac{\partial^2 w}{\partial x^2} &= 0 \end{aligned} \tag{7.7}$$

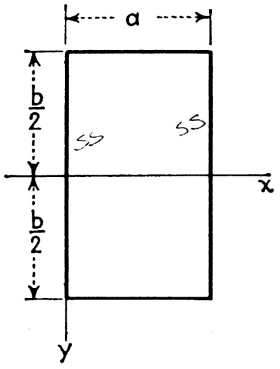


Figure 7.6: Sides a and b in relation to the x- and y-axis (Timoshenko and Woinowsky-Krieger 1959)

Now Y_m needs to be determined in a way that the boundary conditions at $y = -b/2$ and $y = b/2$ are satisfied, as well as the biharmonic plate equation. Nádai (1925) came up with the idea to divide the solution w in two parts:

$$w = w_1 + w_2 \tag{7.8}$$

where

$$w_1 = \frac{p}{24D}(x^4 - 2ax^3 + a^3x) \tag{7.9}$$

this part of the solution represents the deflection of a uniformly loaded strip parallel to the x-axis (Timoshenko and Woinowsky-Krieger 1959). It satisfies Equation 7.5 and the boundary conditions at $x = 0$ and $x = a$.

The part w_2 now has to satisfy:

$$\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = 0 \tag{7.10}$$

and needs to be chosen such that Equation 7.8 holds. w_2 is taken in the form of Equation 7.6 and substituted in Equation 7.10:

$$\sum_{m=1}^{\infty} \left(Y_m^{IV} - 2\frac{m^2\pi^2}{a^2} Y_m^{II} + \frac{m^4\pi^4}{a^4} Y_m \right) \sin\frac{m\pi x}{a} = 0 \tag{7.11}$$

this can only be true for every value of x if:

$$Y_m^{IV} - 2\frac{m^2\pi^2}{a^2} Y_m^{II} + \frac{m^4\pi^4}{a^4} Y_m = 0 \tag{7.12}$$

the general integral of this equation is:

$$Y_m = \frac{pa^4}{D}\left(A_m \cosh\frac{m\pi y}{a} + B_m \frac{m\pi y}{a}\sinh\frac{m\pi y}{a} + C_m \sinh\frac{m\pi y}{a} + D_m \frac{m\pi y}{a}\cosh\frac{m\pi y}{a} \right) \tag{7.13}$$

Because of the symmetrical deflection with respect to the x-axis the constants C_m and D_m can be taken 0. Equation 7.8 can now be rewritten as:

$$w = \frac{p}{24D}(x^4 - 2ax^3 + a^3x) + \frac{pa^4}{D}\left(A_m \cosh\frac{m\pi y}{a} + B_m \frac{m\pi y}{a}\sinh\frac{m\pi y}{a} \right) \sin\frac{m\pi y}{a} \tag{7.14}$$

the first part of this equation (w_1) can be written as:

$$\frac{p}{24D}(x^4 - 2ax^3 + a^3x) = \frac{4pa^4}{\pi^5 D}\sum_{m=1}^{\infty}\frac{1}{m^5}\sin\frac{m\pi x}{a} \tag{7.15}$$

so Equation 7.14 can be rewritten as:

$$w = \frac{pa^4}{D}\sum_{m=1}^{\infty}\left(\frac{4}{\pi^5 m^5} + A_m \cosh\frac{m\pi y}{a} + B_m \frac{m\pi y}{a}\sinh\frac{m\pi y}{a} \right)\sin\frac{m\pi x}{a} \tag{7.16}$$

The boundary conditions in Equation 7.7 also hold for the sides $y = -b/2$ and $y = b/2$ in case of a plate that is simply supported on all edges. By substitution of w in these

boundary conditions and by taking $\alpha_m = \frac{m\pi b}{2a}$ it is possible to determine the constants (A_m and B_m):

$$\begin{aligned} \frac{4}{\pi^5 m^5} + A_m \cosh\alpha_m + \alpha_m B_m \sinh\alpha_m = 0 \\ (A_m + 2B_m)\cosh\alpha_m + \alpha_m B_m \sinh\alpha_m = 0 \end{aligned} \tag{7.17}$$

so the constants are:

$$\begin{aligned} A_m &= -\frac{2(\alpha_m \tanh\alpha_m + 2)}{\pi^5 m^5 \cosh\alpha_m} \\ B_m &= \frac{2}{\pi^5 m^5 \cosh\alpha_m} \end{aligned} \tag{7.18}$$

so the final equation for the displacement field for design 1A is:

$$w = \frac{4pa^4}{\pi^5 D} \sum_{m=1,3,5,\ldots}^{\infty} \frac{1}{m^5}\left(1 - \frac{\alpha_m \tanh\alpha_m + 2}{2\cosh\alpha_m}\cosh 2\alpha_m yb + \frac{\alpha_m}{2\cosh\alpha_m}\frac{2y}{b}\sinh\frac{2\alpha_m y}{b}\right)\sin\frac{m\pi x}{a} \tag{7.19}$$

Using this equation the bending moments and shear forces in the plate can be calculated. The bending moments are given by Equation 7.2 and the shear forces by:

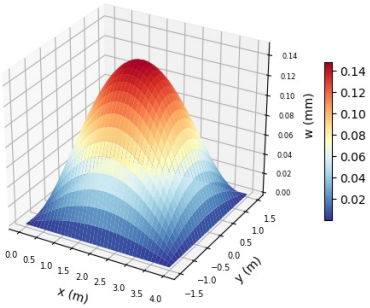
$$\begin{aligned} v_x &= D\left(\frac{\partial^3 w}{\partial x^3} + \frac{\partial^3 w}{\partial y^2 \partial x}\right) \\ v_y &= D\left(\frac{\partial^3 w}{\partial y^3} + \frac{\partial^3 w}{\partial x^2 \partial y}\right) \end{aligned} \tag{7.20}$$

The results are visualised in Figure 7.7 and listed in Table 7.1. The maximum deflections and maximum bending moments are found at $x = a/2$ and $y = 0$. The maximum and minimum twisting moments are found at the corner points. The maximum and minimum shear forces can be found at midspan of the edges. The maximum stresses are calculated using:

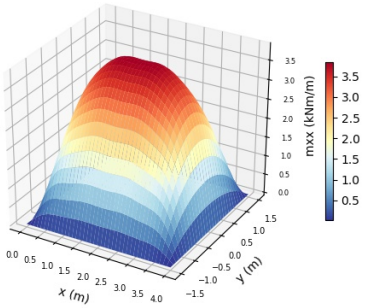
$$\begin{aligned} \sigma_x &= \frac{12z}{h^3} m_{xx_{max}} &&= \pm\frac{6}{h^2} m_{xx_{max}} \\ \sigma_y &= \frac{12z}{h^3} m_{yy_{max}} &&= \pm\frac{6}{h^2} m_{yy_{max}} \\ \tau_{xy} &= \frac{12z}{h^3} m_{xy_{max}} &&= \pm\frac{6}{h^2} m_{xy_{max}} \end{aligned} \tag{7.21}$$

Result		Minimum	Maximum
w	$[mm]$	0.000	0.149
m_{xx}	$[kNm/m]$	0.000	3.858
m_{yy}	$[kNm/m]$	0.000	6.494
m_{xy}	$[kNm/m]$	-4.998	4.998
v_x	$[N/mm]$	-10.71	10.71
v_y	$[N/mm]$	-13.11	13.11
σ_x	$[N/mm^2]$	-0.257	0.257
σ_y	$[N/mm^2]$	-0.433	0.433
τ_{xy}	$[N/mm^2]$	-0.333	0.333

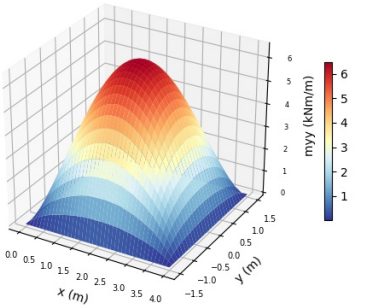
Table 7.1: Results plate theory calculations for design 1A (1A-P-100)



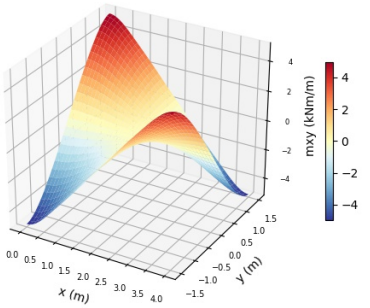
(a) Deflections



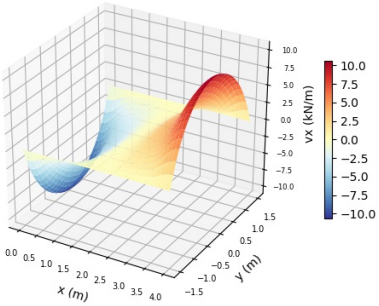
(b) Bending moments m_{xx}



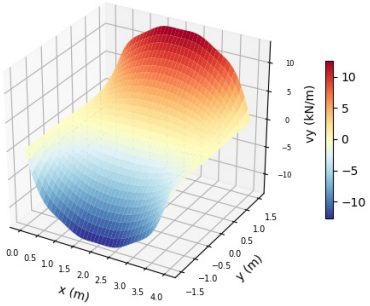
(c) Bending moments m_{yy}



(d) Twisting moments m_{xy}



(e) Shear forces v_x



(f) Shear forces v_y

Figure 7.7: Results floor plate calculations (1A-P-100)

7.1.2 Design 1B - Uniformly loaded plate, simply supported on two short edges

The second design is the uniformly loaded plate, simply supported on the two short edges. So the edges $x = 0$ and $x = a$ are simply supported and the other edges are free. Again we assume that the load is uniformly distributed and the deflection will be symmetrical with respect to the x-axis, so we only have to consider the boundary conditions along the edge $y = b/2$.

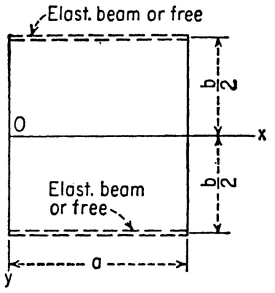


Figure 7.8: Sides a (free) and b (simply supported) in relation to the x- and y-axis (Timoshenko and Woinowsky-Krieger 1959)

According to Timoshenko and Woinowsky-Krieger (1959) we first have to assume that the free edges are elastically supported. They only resist bending in vertical planes and do not resist torsion, therefore the boundary conditions are:

$$\begin{aligned} \left(\frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2}\right)_{y=b/2} &= 0 \\ D\left[\frac{\partial^3 w}{\partial y^3} + (2-\nu)\frac{\partial^3 w}{\partial x^2 \partial y}\right]_{y=b/2} &= \left(EI\frac{\partial^4 w}{\partial x^4}\right)_{y=b/2} \end{aligned} \tag{7.22}$$

where EI is the flexural rigidity of the elastically supporting beams. In a similar way as in Section 7.1.1 a deflection field w is assumed:

$$w = w_1 + w_2 \tag{7.23}$$

where

$$w_1 = \frac{4pa^4}{\pi^5 D}\sum_{m=1,3,5,\ldots}^{\infty}\frac{1}{m^5}\sin\frac{m\pi x}{a} \tag{7.24}$$

and

$$w_2 = \sum_{m=1,3,5,\ldots}^{\infty} Y_m \sin\frac{m\pi x}{a} \tag{7.25}$$

Again we assume the constants $C_m = D_m = 0$, because of symmetry in the x-axis. So,

$$Y_m = \frac{pa^4}{D}\left(A_m \cosh\frac{m\pi y}{a} + B_m\frac{m\pi y}{a}\sinh\frac{m\pi y}{a}\right) \tag{7.26}$$

Now we have the same unknown constants as in Section 7.1.1, however the boundary conditions are different. Filling in the new boundary conditions in Equation 7.22 and

using the notations $\frac{m\pi b}{2a} = \alpha_m$ and $\frac{EI}{aD} = \lambda$ we get:

$$\begin{aligned} A_m(1-\nu)\cosh\alpha_m + B_m[2\cosh\alpha_m + (1-\nu)\alpha_m\sinh\alpha_m] &= \frac{4\nu}{m^5\pi^5} \\ -A_m[(1-\nu)\sinh\alpha_m + m\pi\lambda\cosh\alpha_m] & \\ +B_m[(1+\nu)\sinh\alpha_m - (1-\nu)\alpha_m\cosh\alpha_m - m\pi\lambda\alpha_m] &= \frac{4\lambda}{m^4\pi^4} \end{aligned} \tag{7.27}$$

Solving these equations, we find:

$$\begin{aligned} A_m &= \frac{4}{m^5\pi^5}\frac{\nu(1+\nu)\sinh\alpha_m - \nu(1-\nu)\alpha_m\cosh\alpha_m - m\pi\lambda(2\cosh\alpha_m + \alpha_m\sinh\alpha_m)}{(3+\nu)(1-\nu)(\sinh\alpha_m\cosh\alpha_m - (1-\nu)^2\alpha_m + 2m\pi\lambda\cosh^2\alpha_m} \\ B_m &= \frac{4}{m^5\pi^5}\frac{\nu(1-\nu)\sinh\alpha_m + m\pi\lambda\cosh\alpha_m}{(3+\nu)(1-\nu)\sinh\alpha_m\cosh\alpha_m - (1-\nu)^2\alpha_m + 2m\pi\lambda\cosh^2\alpha_m} \end{aligned} \tag{7.28}$$

the final deflection is found by filling in these constants in:

$$w = w_1 + w_2 = \frac{pa^4}{D}\sum_{m=1,3,5,\ldots}^{\infty}\left(\frac{4}{\pi^5 m^5} + A_m\cosh\frac{m\pi y}{a} + B_m\frac{m\pi y}{a}\sinh\frac{m\pi y}{a}\right)\sin\frac{m\pi x}{a} \tag{7.29}$$

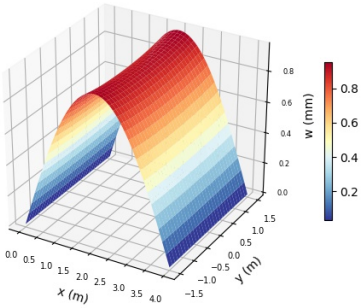
In the considered case two edges are free, instead of elastically supported, so we can fill in $\lambda = 0$ (if we would fill in $\lambda = \infty$ we would obtain the same result as in Section 7.1.1.

Now that w can be calculated, the bending moments, twisting moments, shear forces and the maximum stresses are calculated again using Equations 7.2, 7.20 and 7.21.

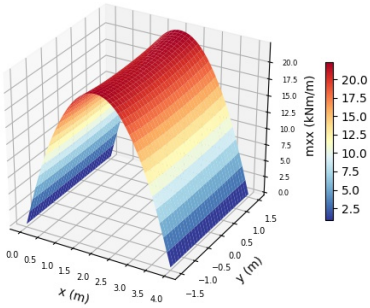
Result		Minimum	Maximum
w	$[mm]$	0.000	0.957
m_{xx}	$[kNm/m]$	0.000	22.38
m_{yy}	$[kNm/m]$	0.000	1.583
m_{xy}	$[kNm/m]$	-1.734	1.734
v_x	$[N/mm]$	-19.95	19.95
v_y	$[N/mm]$	-1.171	1.171
σ_x	$[N/mm^2]$	-1.492	1.492
σ_y	$[N/mm^2]$	-0.106	0.106
τ_{xy}	$[N/mm^2]$	-0.116	0.116

Table 7.2: Results plate theory calculations for design 1B (1B-P-100)

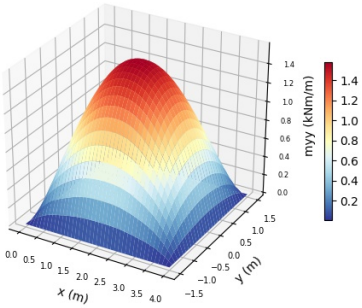
The results are visualised in Figure 7.9.



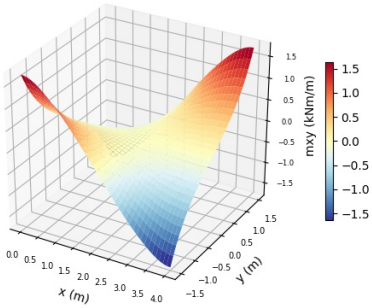
(a) Deflections



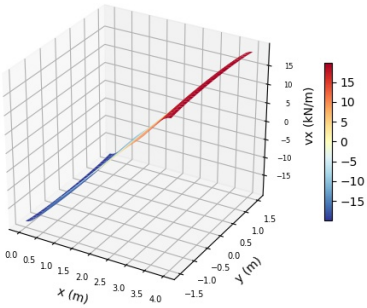
(b) Bending moments m_{xx}



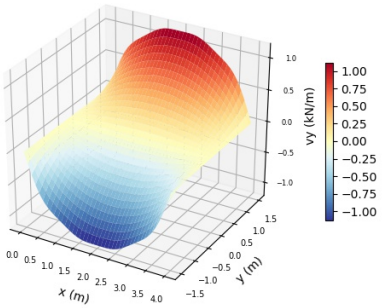
(c) Bending moments m_{yy}



(d) Twisting moments m_{xy}



(e) Shear forces v_x



(f) Shear forces v_y

Figure 7.9: Results floor plate calculations (1B-P-100)

7.1.3 Design 1A and 1B - 60% of the total volume

In Section 7.2 floors will be optimised using a volume constraint of 60%. To make a clear comparison afterwards, first the two flat floors with a volume of 60% are calculated. The height of the floor is now 180 mm instead of 300 mm. The results are listed in the tables below.

Result		Minimum	Maximum
w	$[mm]$	0.000	0.539
m_{xx}	$[kNm/m]$	0.000	3.018
m_{yy}	$[kNm/m]$	0.000	5.080
m_{xy}	$[kNm/m]$	-3.910	3.910
v_x	$[N/mm]$	-8.382	8.382
v_y	$[N/mm]$	-10.25	10.25
σ_x	$[N/mm^2]$	-0.559	0.559
σ_y	$[N/mm^2]$	-0.941	0.941
τ_{xy}	$[N/mm^2]$	-0.724	0.724

Table 7.3: Results plate theory calculations for design 1A (1A-P-60)

Result		Minimum	Maximum
w	$[mm]$	0.000	3.468
m_{xx}	$[kNm/m]$	0.000	17.51
m_{yy}	$[kNm/m]$	0.000	1.238
m_{xy}	$[kNm/m]$	-1.356	1.356
v_x	$[N/mm]$	-15.60	15.60
v_y	$[N/mm]$	-0.916	0.916
σ_x	$[N/mm^2]$	-3.242	3.242
σ_y	$[N/mm^2]$	-0.229	0.229
τ_{xy}	$[N/mm^2]$	-0.251	0.251

Table 7.4: Results plate theory calculations for design 1B (1B-P-60)

7.2 Topology optimised floor

To properly model and optimise a floor the aim is to use a design space which is as large as possible. The larger the amount of elements, the higher the computational costs. By choosing different numbers of elements and keeping track of the increase in time, a design space of 40 x 30 x 6 elements has been chosen for this case study. A compliance minimisation subjected to a volume constraint takes about three quarters of an hour. Doubling the thickness (from 6 to 12 elements) leads to an exponential growth in the total time of about 3 hours.

Because of symmetry in the yz-plane and in the xz-plane a floor of 80 x 60 x 6 elements can be optimised. A total amount of 28800 elements is thus considered. All elements have

a length of 50 mm and therefore the design space has a length of 4 m, a width of 3 m and a thickness or height of 300 mm.

According to Section 4.3 the Young's modulus in the strong x- and y-direction is $E_{x,y} = 17336\ N/mm^2$. In the weaker direction the modulus is about 65% of that value. So the material to be used is transverse isotropic and the factor $E_3 = 0.65$ for the weak z-direction. The density of the concrete mixture is, according to mentioned section, equal to 2029 kg/m^3.

As stated above the top layer cannot be removed from the design, because a distributed load is acting on it. The top 1200 (or actually 4800) elements are therefore set to solid. No elements are set to zero density (passive) in the design space, because there is no restriction to place material anywhere.

The load acting on the top surface of the floor slab is set to 5.0 kN/m^2. The modelled part of the floor has an area of 2.0 m × 1.5 m = 3.0 m^2. So a total force of 5.0 kN/m^2 × 3.0 m^2 = 15 kN is divided over the nodes on the top surface.

Furthermore, the optimisation will be a compliance minimisation with a volume constraint of 60%. The additive manufacturing filter without a possible overhang (AM0) is used and the top surface is taken as the baseplate. The floor will thus be printed upside down and be rotated afterwards.

All parameters and assumptions are listed in Table 7.5 for a clear overview.

Parameter	Value
Elements in x-direction	40 (80 using symmetry)
Elements in y-direction	30 (60 using symmetry)
Elements in z-direction	6
Element size $[mm]$	50
Length (x-direction) $[mm]$	2000 (4000 using symmetry)
Width (y-direction) $[mm]$	1500 (3000 using symmetry)
Height (z-direction) $[mm]$	300
Loads $[N]$	15000 (divided over top nodes)
$E_{x,y}$ $[N/mm^2]$	17336
Factor E_3	0.65
ν [-]	0.12
Density $[kg/m^3]$	2029
Self-weight	Considered
Objective function	Minimising compliance
Constraint function	Volume
Volume limit	0.60
Penalisation power p	3
Penalisation power q	2.8
Filter radius	2.5
Filter	Robust
Solver	MMA
Material model	Transverse isotropic
Manufacturing constraint	AM0
Baseplate	Top
Tolx	0.001
Maxloop	200

Table 7.5: Parameters used in the topology optimisation algorithm

The resulting topologies can be seen in Figure 7.11 and the resulting values are listed in Table 7.6.

From the results it can be concluded that the maximum deflection of a slab is highly dependent on the boundary conditions. Without the additional supports on two sides the displacement is almost two times larger. Clearly visible is that the compliance of the structure is therefore much larger as well for the same volume fraction. Almost all stresses are larger for design 1B as well.

To make a fair comparison between the values coming from the plate theory (with volume fraction 100%, called P-100) and the finite element method used in this research, a simple non-optimised floor is calculated in the algorithm as well. This non-optimised floor has a volume fraction of 100% and is visualised in Figure 7.10 and denoted as TO-100.

The resulting values from the 100% volume floor (TO-100) and the results from the optimised structures (TO-60) are calculated in the algorithm and listed in Table 7.6.

Result	1A-TO-100	1A-TO-60	1B-TO-100	1B-TO-60
Volume fraction [-]	1.00	0.60	1.00	0.60
Compliance $[Nm]$	2.48	3.49	19.77	20.36
$\sigma_{vM_{max}}$ $[N/mm^2]$	0.53	0.77	1.23	1.73
$\sigma_{vM_{min}}$ $[N/mm^2]$	0	0	0	0
$\sigma_{xx_{max}}$ $[N/mm^2]$	0.23	0.67	1.22	1.52
$\sigma_{xx_{min}}$ $[N/mm^2]$	-0.23	-0.37	-1.22	-1.18
$\sigma_{yy_{max}}$ $[N/mm^2]$	0.39	0.61	0.07	0.58
$\sigma_{yy_{min}}$ $[N/mm^2]$	-0.39	-0.51	-0.10	-0.26
$\sigma_{zz_{max}}$ $[N/mm^2]$	0.50	0.10	0.02	0.15
$\sigma_{zz_{min}}$ $[N/mm^2]$	-0.35	-0.63	-0.68	-1.43
$\tau_{yz_{max}}$ $[N/mm^2]$	0.22	0.26	0.06	0.48
$\tau_{yz_{min}}$ $[N/mm^2]$	-0.22	-0.37	-0.06	-0.53
$\tau_{zx_{max}}$ $[N/mm^2]$	0.13	0.23	0.05	0.21
$\tau_{zx_{min}}$ $[N/mm^2]$	-0.16	-0.19	-0.00	-0.33
$\tau_{xy_{max}}$ $[N/mm^2]$	0.13	0.23	0.26	0.57
$\tau_{xy_{min}}$ $[N/mm^2]$	-0.17	-0.23	-0.03	-0.18
$U_{z_{max}}$ $[mm]$	0.03	0.29	0.12	0.66
$U_{z_{min}}$ $[mm]$	-0.17	-1.32	-0.97	-2.40
Total time $[s]$	42	2623	42	2688

Table 7.6: Results from the topology optimisations

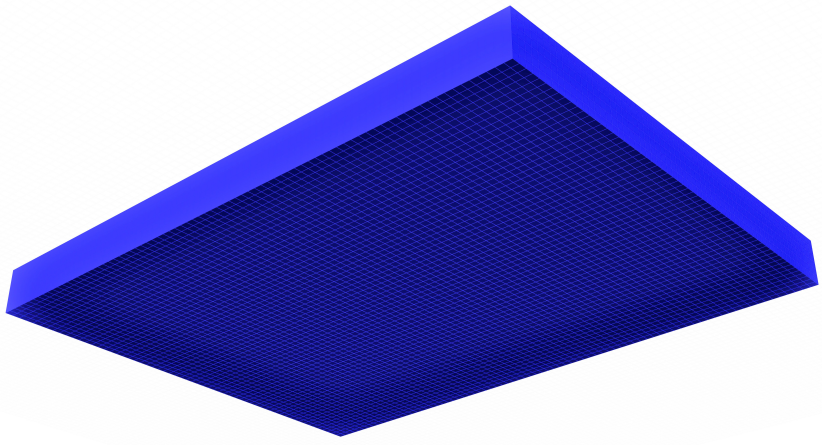


Figure 7.10: Non-optimised outcome of the created algorithm: TO-100

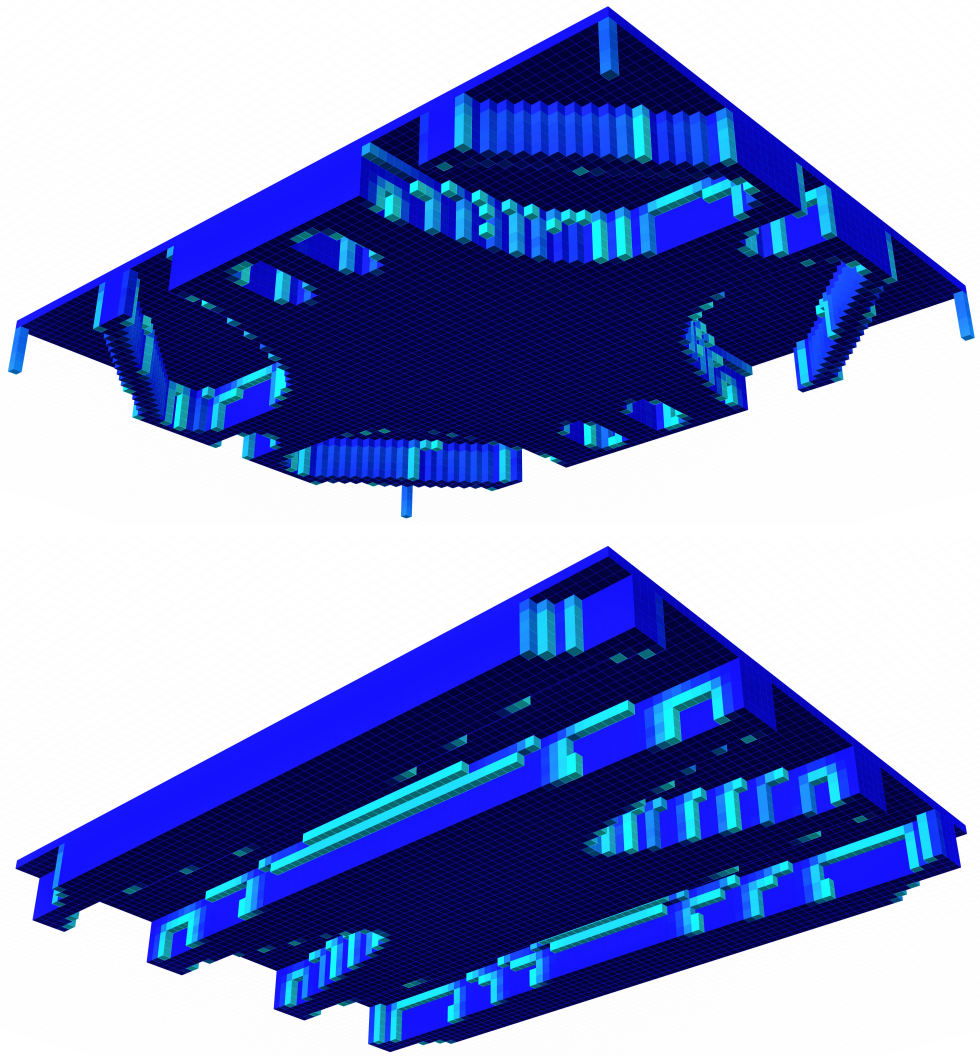


Figure 7.11: Final topologies for different boundary conditions 1A-TO-60 (top) and 1B-TO-60 (bottom)

In addition to a comparison between flat floors and topology optimised floors, the results of the improvements made in this research are studied as well. To make a clear comparison between a standard code and the code created in this work, the designs 1A and 1B are also optimised without all the adaptations. The isotropic material behaviour is used, self-weight is not considered, the density filter is used instead of the robust filter and no manufacturing filter is considered. The deviating result can be seen in Figure 7.12.

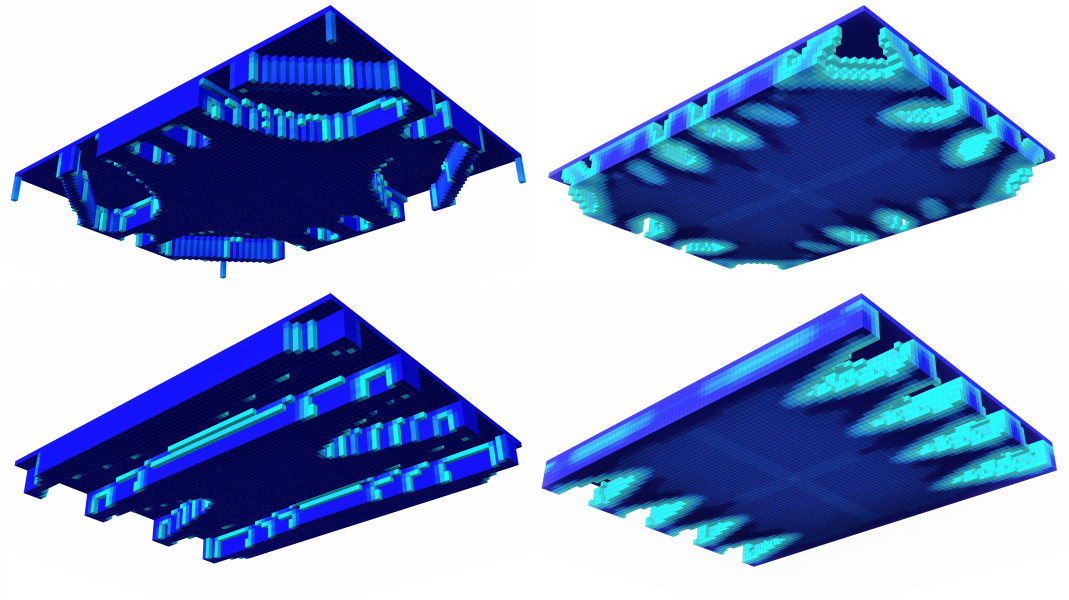


Figure 7.12: Final topologies for 1A-TO-60 (top) and 1B-TO-60 (bottom) (left = the created algorithm, right = standard TO)

Result	1A-TO-60-std	1A-TO-60-real	1B-TO-60-std	1B-TO-60-real
Volume fraction [-]	0.60	0.60	0.60	0.60
Compliance $[Nm]$	0.931	3.49	6.31	20.36
$\sigma_{vM_{max}}$ $[N/mm^2]$	0.79	0.77	1.19	1.73
$\sigma_{vM_{min}}$ $[N/mm^2]$	0	0	0	0
$\sigma_{xx_{max}}$ $[N/mm^2]$	0.283	0.67	0.81	1.52
$\sigma_{xx_{min}}$ $[N/mm^2]$	-0.244	-0.37	-0.72	-1.18
$\sigma_{yy_{max}}$ $[N/mm^2]$	0.345	0.61	0.27	0.58
$\sigma_{yy_{min}}$ $[N/mm^2]$	-0.266	-0.51	-0.17	-0.26
$\sigma_{zz_{max}}$ $[N/mm^2]$	0.059	0.10	0.08	0.15
$\sigma_{zz_{min}}$ $[N/mm^2]$	-0.348	-0.63	-0.91	-1.43
$\tau_{yz_{max}}$ $[N/mm^2]$	0.125	0.26	0.24	0.48
$\tau_{yz_{min}}$ $[N/mm^2]$	-0.184	-0.37	-0.25	-0.53
$\tau_{zx_{max}}$ $[N/mm^2]$	0.200	0.23	0.15	0.21
$\tau_{zx_{min}}$ $[N/mm^2]$	-0.269	-0.19	-0.16	-0.33
$\tau_{xy_{max}}$ $[N/mm^2]$	0.221	0.23	0.47	0.57
$\tau_{xy_{min}}$ $[N/mm^2]$	-0.271	-0.23	-0.06	-0.18
$U_{z_{max}}$ $[mm]$	0.03	0.29	0.09	0.66
$U_{z_{min}}$ $[mm]$	-0.14	-1.32	-0.66	-2.40
Total time $[s]$	2245	2623	2382	2688

Table 7.7: Results from the standard (std) and the more realistic (real) topology optimisations

7.3 Discussion

In this section the results of the plate theory and the optimised results are discussed. Note that the displacements U in the algorithm are negative, because the z-axis is pointing up. In the plate theory the z-axis is pointing down, so the deflections w are positive. In the tables below the maximum deflection is positive and called w_{max}.

In the following two tables the non-optimised design from the algorithm (TO-100) is listed in the first column, the topology optimised design (TO-60) in the second column and the outcome of the plate theory with 100% of the volume in the third column (P-100). In the last column a floor with the same amount of material as the optimised result is listed (P-60). In both cases a floor with a thickness of 180 mm is calculated.

The results for design 1A, a slab simply supported on all edges with a distributed load on the top surface, is listed Table 7.8.

Result	1A-TO-100	1A-TO-60	1A-P-100	1A-P-60
Volume fraction [-]	1.00	0.60	1.00	0.60
w_{max} $[mm]$	0.17	1.32	0.15	0.54
σ_x $[N/mm^2]$	0.23	0.67	0.26	0.56
σ_y $[N/mm^2]$	0.39	0.61	0.43	0.94
τ_{xy} $[N/mm^2]$	0.17	0.23	0.33	0.72

Table 7.8: Combined results for design 1A

For design 1B, a slab simply supported on the two short edges with a distributed load on the top surface, the following result are obtained.

Result	1B-TO-100	1B-TO-60	1B-P-100	1B-P-60
Volume fraction [-]	1.00	0.60	1.00	0.60
w_{max} $[mm]$	0.97	2.40	0.96	3.47
σ_x $[N/mm^2]$	1.22	1.52	1.49	3.24
σ_y $[N/mm^2]$	0.10	0.58	0.11	0.23
τ_{xy} $[N/mm^2]$	0.26	0.57	0.12	0.25

Table 7.9: Combined results for design 1B

- *Comparison between TO and plate theory results (TO-100 and P-100)*
  Clearly visible in the tables above is that, as expected, the non-optimised results from the algorithm (TO-100) are very similar to the results from the plate theory (P-100). Small differences can be explained by the use of very simple elements in the finite element method. The used linear elements cannot capture the kinematics of deformation. In the theory we assume that the shear strain can be neglected. However, because of the linear elements the presence of a shear strain is inevitable. Because first-order solid elements are used, the displacement field is approximated by a linear function. The strains are a derivative of the displacement, which is a constant value for linear functions. This leads to a constant strain in the elements, which is not

really the case. This leads to a different stiffness, which leads to different deflections. This problem can be diminished by refining the mesh, however the computational time grows exponentially when more elements are used. Improving that aspect is out of the scope of this research. It is also possible to solve this problem by choosing second or higher order elements.

- *Comparison between total volume and 60% of the volume (TO-100 and TO-60)*
  Another expected result visible in the comparison is that the stresses and displacements become larger if material is saved. The material is thus used in a more efficient way. Neither the stress limit (σ_{zz} <1.0 N/mm^2), nor the deflection limit ($w_{max} < 0.004L$=16 mm) is exceeded, so more material can be saved. Of course there is tension (in the other directions), so actually reinforcement is needed anyhow. This aspect is outside the scope of the research as well, however it should be mentioned.

- *Comparison between topology optimised floors and thinner floors (TO-60 and P-60)*
  When comparing the topology optimised results (TO-60) with the thinner floors (P-60) it is clear that in case of design 1A the maximum deflection and the maximum stress $sigma_x$ are higher in case of TO than in case of a thinner floor. Considering stiffness and the used material volume it is therefore better to choose a thinner floor in this situation. For stresses in the other directions the topology optimised floor is preferred.

  In case of design 1B the opposite is true. In this case the topology optimised result (1B-TO-60) leads to a smaller maximum displacement and a lower maximum stress $sigma_x$ than the thinner floor (1B-P-60). So it depends on the boundary conditions what is the best thing to do, when considering the deflection and thus the stiffness of the structure. When considering the strength or stresses in the structure it depends on the boundary conditions and the stress criterion what method is preferred.

In addition to the comparison between the theory and the optimisation it is also interesting to compare the standard topology optimisation with the created algorithm. In that way it can be clearly visualised what the effects of the material model, filters and other changes are.

- *Comparison regular TO with the created algorithm (TO-60-std and TO-60-real)*
  From the results in Table 7.7 and the resulting topologies it is clear that completely different outcomes are created by the different TO approaches. The standard TO results in much lower compliances, is therefore stiffer and has a maximum deflection that is much smaller than in the created algorithm. This makes sense, because the new results are limited by manufacturing constraints and the real behaviour of the material is included. When considering the topology it is very clear that many holes are present in the standard optimised floor. These holes cannot be printed by a 3D concrete printer, without moulds.

In addition to designs 1A and 1B more boundary conditions have been used to optimise floors. For example, a floor supported on the four corner points (1C) and a floor supported on a column at the bottom and centre of the slab (1D). These results are not used to compare results numerically, however the differences in topologies can be compared.

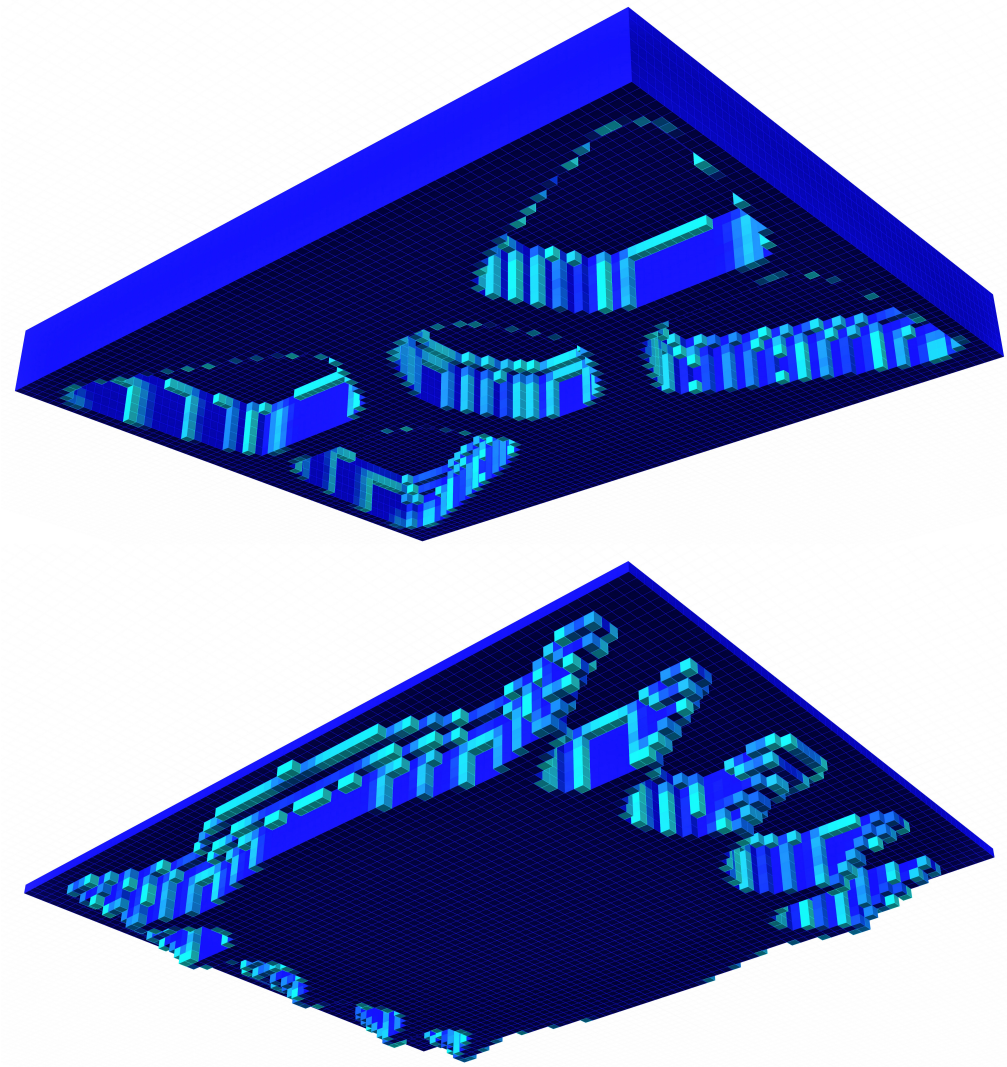


Figure 7.13: Final topologies for boundary conditions 1C (top) and 1D (bottom)

8 Conclusions

In this chapter the most important conclusions of the study are listed and an answer to the research question is given.

8.1 Answer to the research question

The research question of this study was:

What possibilities can additive manufacturing of concrete provide in the design and construction of topology optimised structures?

As the main conclusion and as an answer to the research question it can be said that combining topology optimisation and additive manufacturing can lead to *material savings* and a more *automated building process*. However, from this study it can be concluded that taking into account manufacturing constraints and the real material behaviour of printed concrete has a great effect on the structural performance of the optimised results. From the case study, for example, it can even be said that it depends on the boundary conditions if it is more beneficial to use topology optimisation in combination with 3D printing or to simply design a thinner rectangular floor.

8.2 General conclusions

- From the literature study on recent concrete printing projects, it can be concluded that all over the world different groups are focussing on large scale 3D printing. From the case study it can be concluded that *the manufacturing constraints heavily influence the structural behaviour*. Because many projects are focussing on concrete printing, the processes are improving and more experience is gained. This will

probably lead to *more form-freedom and therefore more-optimised structures in the years to come*. The knowledge of the material will improve as well.

- By applying a new material model to the topology optimisation more realistic designs are the result of the optimisation. The stiffness behaviour in the optimisation is changed from isotropic to the special orthotropic case: transverse isotropic. This is done assuming that the material behaves isotropic in its plane and different in the height (printing direction). From the examples in Section 6.1.4 it can be concluded that *a transverse isotropic material model results in considerably different topologies than an isotropic material model*. To improve the material model decent research is needed in the behaviour of printed concrete in topology optimised structures.

- The stresses in different directions are not processed in the material model, but in the stress constraints. In the research a von Mises stress constraint and a Drucker-Prager stress constraint can be applied. By applying these constraints to the optimisation the stress limits (average stress or tension/compression) will not be exceeded if possible with the chosen material and the chosen design space and loads. A *stress constraint has a considerable effect on the resulting topology*, as can be concluded from Section 6.2.3. As a conclusion from Section 6.3 it can also be said that the *choice for the von Mises or the Drucker-Prager constraint is very important for the final design*.

- To take into account a certain print angle, two additive manufacturing filters are included in the created algorithm. One filter tries to steer the optimum towards a design without an overhang. The second filter allows a maximum overhang of 45 degrees. The degrees are measured from a baseplate. This baseplate can be changed to all the six border surfaces of the design space. In case another baseplate than the bottom surface is chosen, the final design should be rotated after printing. *Taking into account a manufacturing filter and changing the baseplate results in very different topologies* as can be concluded from Section 6.2.1 and from the case study (Figure 7.12).

By applying all these constraints, filters and a different material model to the standard topology optimisation codes, the final outcome of the process is a much *more realistic design* than the final designs coming from the standard methods. The obtained final results are much closer to a printable design than the resulting structures of the optimisation without the filters, constraints and with the standard isotropic material model. However, from the case study it can also be concluded that this is *at the expense of the strength and stiffness* of the structure. Nevertheless, the case study also shows that for different boundary conditions, different parts of a box-shaped floor (design space) can be removed, without exceeding the limits for stiffness or strength.

- The developed algorithm makes it *possible to design different optimised structures by making use of several user-friendly steps*. From working with the created algorithms it can be concluded that setting up a design space is convenient, the optimisation is done only by running the Python script and in the end the final design can be visualised, adapted and exported in Rhino very straightforward. The total flow of work is therefore easy and very flexible. Because of the link between Rhino/Grasshopper and Python every parameter can be changed in a visually appealing way instead of coding inside Python. The results from the optimisation can finally be viewed in different ways: as Excel lists, as squares/cubes with colours that match the values

for density or stress, or the results can be smoothed and used as a 3D model for visualisations or further calculations. For example, in Figure 8.1 the created floors from the case study are smoothed and rendered in Rhino.

Figure 8.1: 3D impression of topology optimised concrete floors

- Unfortunately, sometimes the final results are not pleasing. In some cases the resulting structure is not converging to a crisp black and white solution. If the structure is going to be printed, this needs to be the case. Virtual densities between 0 and 1 cannot be printed. *The designer needs to think carefully about specifying the optimisation problem, to avoid unwanted results.* For example, if a complete structure is loaded in tension, it is not possible to come up with a structure that does not contain tensile forces. The same holds for cases where additive manufacturing filters are used in an illogical way, such as the last topology in Figure 6.14 in Section 6.2.1. In those cases an almost fully filled design space would be needed, however if a volume limit is set, this will not be the result of the optimisation. It should therefore be noted that parameters must be chosen carefully and that results should be examined afterwards.

9

Recommendations

In this chapter the recommendations for further research are presented.

The combination of topology optimisation and additive manufacturing can lead to better concrete structures in the future, however only if more research is done and some aspects of the optimisation are improved.

- First of all the safety of 3D printed optimised results should be assured. Before the safety can be guaranteed a lot of tests should be performed first to give a clear *insight in the effect of changing (manufacturing) parameters and the real material model*. For example, research must be done in the effect on the strength and stiffness of the material if the time in between printing two layers next to or on top of each other varies. The algorithm now considers a material that is transverse isotropic, however the material behaviour depends on a large amount of variables, so that assumption should be examined. Another aspect that is not considered in this research but might influence the safety of the result is the *non-linear behaviour of concrete*. This is not researched in this study because the deflections are assumed to be small, so the material is still in the elastic phase.

- Furthermore, it is also recommended to study *better stress constraints*. The created algorithm searches for the best result without exceeding a certain von Mises stress or an equivalent Drucker-Prager stress. These failure criteria do not consider the different strengths in different directions. Printed concrete, for example, is stronger in tension in the direction of printing than in tension perpendicular to the print direction (in z-direction). That distinction is not made in the Drucker-Prager stress criterion, while the von Mises stress criterion does not even make a distinction between tension and compression. It might be a good idea to incorporate the Mohr-Coulomb criterion in the optimisation. In that case both the stiffness as well as the strength limit are non-isotropic.

- In the optimisation the restriction of the printer that is implemented is the angle of printing. The choice can be made between an angle of zero or 45 degrees. It is also possible to switch off this constraint. In future work it should be possible to choose any desired angle, because that angle is different for different materials and different printers. Maybe *other manufacturing constraints* can be incorporated into the optimisation as well. The size of the printer, the size of the printing nozzle, the speed of the printer, the angle of printing (which is a different angle than the support angle) or the start/stop-system of the printer.

- Another practical challenge that needs to be resolved is the *link between an optimised model and the actual construction* of that design. In the current algorithm a design is formed based on square or box-shaped elements, however it is not possible to print those elements exactly. The idea is now that a 3D smoothed model is created from the optimisation results, which is later sliced. For every slice a printing path is created. This process is not ideal, because it might be possible that a slice cannot be printed in one go. It would probably results in a lot of ”stops” and ”goes”. In future research it might be an option to take printing paths as starting point for the structural optimisation. It then might be possible to combine structural demands with functional and physical needs.

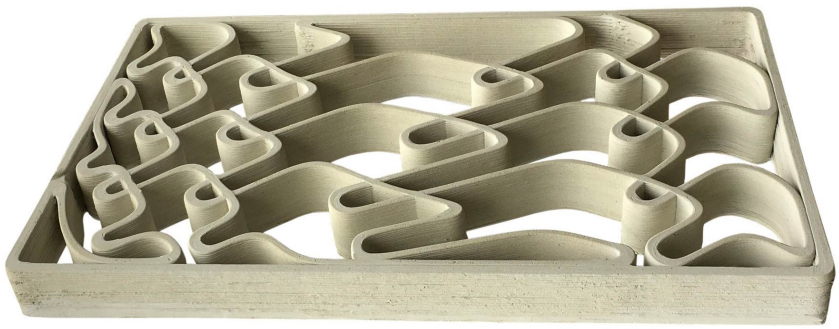


Figure 9.1: Printing paths as starting point to regulate/optimise light income (Tissink 2017)

- The next aspect that needs attention, before topology optimisation and additive manufacturing can be used in designing concrete structures is the *computational time* needed for the optimisations. In this research a pretty coarse mesh is used to calculate results. In real designs the design spaces are much larger and much more complicated, so a larger design space and a finer mesh are needed to analyse these structures properly. This will lead to a large increase in the amount of elements, which will exponentially increase the computational effort. The part of the code that is computationally the most costly is the finite element calculation, the determination of the displacements. If the matrix calculations in this part of the research can be improved, the speed of the optimisations would be much higher. It might also be good to visualise the final outcome in a different way. When importing the results in Rhino, it takes a lot of time to convert the information into a visual. The elements with a virtual density higher than 0.5 are all visualised by a small box with six surfaces. In, for example, the case study this leads to more than 100,000 surfaces.

- Another practical recommendation is to use *different elements* in future research. Now eight node hexahedrons are used in 3D and four node quadrilaterals are used in 2D for simplicity. To obtain better results for stresses and deflections, it is beneficial to use non-linear (higher order) elements.

- In the case study of this research a minimum compliance problem with a volume constraint is solved. As discussed in this work, it might be a better solution to minimise the volume subjected to a stress constraint and a deflection constraint. This is not done, because of the high computational time of the stress constraint and the fact that it is hard to give a value for the compliance on beforehand. To use the full potential of topology optimisation, the *computational effort for the stress constraint should be lowered*. And to use the stiffness constraint in a proper way, it might be an idea to fill in a maximum deflection. In the latter case the *relationship between the total compliance and the maximum deflection should be defined*.

- In applying the forces a total force is equally divided over an amount of nodes. In reality if a surface load is applied corner nodes receive a quarter of the load, line nodes half of the load and inner nodes receive the full load of a considered surface. Another aspect that can *improve the assignment of the loads* is the use of safety factors. Actually different load factors should be applied in different calculations. It might be an option to solve multiple displacement vectors, for example one for the serviceability limit state and one for the ultimate limit state. However, this would lead to double the amount of FEA calculations and thus double the computational time.

- Finally, a practical improvement that can be made is the *combination of the total process in one program*. Now it is necessary to change programs and export and import data. It would be more efficient to do all the work in a single program.

Bibliography

Al Nageim, H.: 2003, *Structural Mechanics: Loads, Analysis, Design and Materials*, Pearson Education.

Alamo, J. and da Silva, F.: 2012, Adapting the simp model for topology optimization of biomechanical structures, *12th Pan-American Congress of Applied Mechanics-PACAM XII*.

Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S. and Sigmund, O.: 2011, Efficient topology optimization in matlab using 88 lines of code, *Structural and Multidisciplinary Optimization* **43**(1), 1–16.

ASTM International: 2013, Standard terminology for additive manufacturing technologies ASTM F2792 - 12a, `http://www.astm.org/cgi-bin/resolver.cgi?F2792`. [29 May 2018].

Atzeni, E. and Salmi, A.: 2012, Economics of additive manufacturing for end-usable metal parts, *The International Journal of Advanced Manufacturing Technology* **62**(9-12), 1147–1155.

Autodesk: 2018, Dynamo Studio, software program, `https://www.autodesk.com/products/dynamo-studio/overview`. [29 May 2018].

Belblidia, F. and Bulman, S.: 2002, A hybrid topology optimization algorithm for static and vibrating shell structures, *International journal for numerical methods in engineering* **54**(6), 835–852.

Bendsøe, M. P. and Kikuchi, N.: 1988, Generating optimal topologies in structural design using a homogenization method, *Computer methods in applied mechanics and engineering* **71**(2), 197–224.

Bendsøe, M. P. and Sigmund, O.: 1995, *Optimization of structural topology, shape, and material*, Springer.

Bendsøe, M. P. and Sigmund, O.: 2003, *Topology Optimization: Theory, Methods, and Applications*, Springer.

Brackett, D., Ashcroft, I. and Hague, R.: 2011, Topology optimization for additive manufacturing, *Proceedings of the solid freeform fabrication symposium, Austin, TX*, pp. 348–362.

Bruggi, M.: 2008, On an alternative approach to stress constraints relaxation in topology optimization, *Structural and multidisciplinary optimization* **36**(2), 125–141.

Bruggi, M. and Duysinx, P.: 2013, A stress–based approach to the optimal design of structures with unilateral behavior of material or supports, *Structural and Multidisciplinary Optimization* **48**(2), 311–326.

Bruns, T. E. and Tortorelli, D. A.: 2001, Topology optimization of non-linear elastic structures and compliant mechanisms, *Computer Methods in Applied Mechanics and Engineering* **190**(26), 3443–3459.

Bruyneel, M., Duysinx, P. and Fleury, C.: 2002, A family of mma approximations for structural optimization, *Structural and Multidisciplinary Optimization* **24**(4), 263–276.

Buren, A.: 2015, Lewis Grand Hotel teams with Andrey Rudenko to develop world's first 3D printed hotel, planning construction of homes, `http://www.3ders.org/articles/20150909-lewis-grand-hotel-andrey-rudenko-to-develop-worlds-first-3d-printed-hotel.html`. [29 May 2019].

Buren, A.: 2016, WinSun 3D prints two gorgeous concrete Chinese courtyards inspired by the ancient Suzhou gardens, `http://www.3ders.org/articles/20160331-winsun-3d-prints-two-gorgeous-concrete-chinese-courtyards-inspired-by-the-ancient-suzhou-gardens.html`. [29 May 2018].

Buswell, R. A., Soar, R. C., Gibb, A. G. F. and Thorpe, A.: 2007, Freeform construction: mega-scale rapid manufacturing for construction, *Automation in construction* **16**(2), 224–231.

Cai, K.: 2011, A simple approach to find optimal topology of a continuum with tension-only or compression-only material, *Structural and Multidisciplinary Optimization* **43**(6), 827–835.

Carpinteri, A.: 2013, *Structural Mechanics Fundamentals*, CRC Press.

Ceccanti, F., Dini, E., De Kestelier, X., Colla, V. and Pambaguian, L.: 2010, 3d printing technology for a moon outpost exploiting lunar soil, *61st International Astronautical Congress, Prague, CZ, IAC-10-D3*, Vol. 3.

Charron, K.: 2015, Exclusive: WinSun China builds world's first 3D printed villa and tallest 3D printed apartment building, `http://www.3ders.org/articles/20150118-winsun-builds-world-first-3d-printed-villa-and-tallest-3d-printed-building-in-china.html`. [29 May 2018].

Cheng, G. and Guo, X.: 1997, ε-relaxed approach in structural topology optimization, *Structural and Multidisciplinary Optimization* **13**(4), 258–266.

Christensen, P. W. and Klarbring, A.: 2008, *An introduction to structural optimization*, Vol. 153, Springer.

Coenders, J.: 2008, CIE5251-09 Structural Design: Special Structures, (3), 331–366.

Colla, V. and Dini, E.: 2013, Large scale 3d printing: From deep sea to the moon, *Low-Cost 3D Printing, for Science, Education & Sustainable Development; Canessa, E., Fonda, C., Zennaro, M., Eds* pp. 127–132.

Corke, G.: 2010, Develop3D - The man who prints buildings, `http://www.develop3d.com/profiles/the-man-who-prints-buildings`. [29 May 2018].

Dini, E.: 2016, What is D-shape, `https://d-shape.com/what-is-it/`. [29 May 2018].

Doomen, C. C. M.: 2016, *The effect of layered manufacturing on the strength properties of printable concrete*, Master's thesis, Eindhoven University of Technology.

Drucker, D. C. and Prager, W.: 1952, Soil mechanics and plastic analysis or limit design, *Quarterly of applied mathematics* **10**(2), 157–165.

Duysinx, P. and Bendsøe, M. P.: 1998, Topology optimization of continuum structures with local stress constraints, *International journal for numerical methods in engineering* **43**(8), 1453–1478.

Duysinx, P. and Sigmund, O.: 1998, New developments in handling stress constraints in optimal material distribution, *Proc of the 7th AIAA/USAF/NASAISSMO Symp on Multidisciplinary Analysis and Optimization*, pp. 1501–1509.

Duysinx, P., Van Miegroet, L., Lemaire, E., Brüls, O. and Bruyneel, M.: 2008, Topology and generalized shape optimization: Why stress constraints are so important?, *International Journal for Simulation and Multidisciplinary Design Optimization* **2**(4), 253–258.

Fancello Ancello, E. A. and Pereira, J. T.: 2003, Structural topology optimization considering material failure, *Latin American Journal of Solids and Structures* **1**(1), 3–24.

Fleury, C.: 1989, Conlin: an efficient dual optimizer based on convex approximation concepts, *Structural optimization* **1**(2), 81–89.

Frazier, W. E.: 2014, Metal additive manufacturing: a review, *Journal of Materials Engineering and Performance* **23**(6), 1917–1928.

Gaynor, A. T. and Guest, J. K.: 2016, Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design, *Structural and Multidisciplinary Optimization* **54**(5), 1157–1172.

Gibson, I., Rosen, D. and Stucker, B.: 2014, *Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing*, Springer.

Goehrke, S. A.: 2015, Concrete Plans: CyBe's Berry Hendriks Describes Plans to 3D Print with Mortar, `https://3dprint.com/35727/cybe-berry-hendriks-concrete/`. [29 May 2018].

Griffiths, S.: 2014, 3D printer creates 10 houses in a day, `http://www.dailymail.co.uk/sciencetech/article-2615076/Giant-3D-printer-creates-10-sized-houses-DAY-Bungalows-built-layers-waste-materials-cost-3-000-each.html`. [29 May 2018].

Guest, J. K., Prévost, J. H. and Belytschko, T.: 2004, Achieving minimum length scale in topology optimization using nodal design variables and projection functions, *International journal for numerical methods in engineering* **61**(2), 238–254.

Guo, X., Cheng, G. and Yamazaki, K.: 2001, A new approach for the solution of singular optima in truss topology optimization with stress and local buckling constraints, *Structural and Multidisciplinary Optimization* **22**(5), 364–373.

Hammer, R.: 2016, Test with 3D-Printed Concrete Formwork a Success, `https://www.heijmans.nl/en/news/test-3d-printed-cement-formwork-success/`. [29 May 2018].

Hofmeyer, H., Schevenels, M. and Boonstra, S.: 2017, The generation of hierarchic structures via robust 3d topology optimisation, *Advanced Engineering Informatics* **33**, 440 – 455.

Holmberg, E., Torstenfelt, B. and Klarbring, A.: 2013, Global and clustered approaches for stress constrained topology optimization and deactivation of design variables, *10th World Congress on Structural and Multidisciplinary Optimization, May 19-24, 2013, Orlando, Florida, USA*, pp. 1–10.

Huang, X. and Xie, M.: 2010, *Evolutionary topology optimization of continuum structures: methods and applications*, John Wiley & Sons.

Huang, X. and Xie, Y. M.: 2007, Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method, *Finite Elements in Analysis and Design* **43**(14), 1039–1049.

Hull, C. W.: 1986, Apparatus for production of three-dimensional objects by stereolithography. US Patent 4575330 [29 May 2018].

Jokić, S. and Novikov, P.: 2016, IAAC- Minibuilders, `http://robots.iaac.net/`. [29 May 2018].

Jokić, S. and Novikov, P.: 2017, IAAC - 3D printed bridge, `https://iaac.net/research-projects/large-scale-3d-printing/3d-printed-bridge/`. [29 May 2018].

Kestelier, X.: 2011, Design potential for large-scale additive fabrication. Free-form construction, *Fabricate: Making Digital Architecture* pp. 244–249.

Khoshnevis, B.: 1998, Innovative rapid prototyping process makes large sized, smooth surfaced complex shapes in a wide variety of materials, *Materials Technology* **13**(2), 52–63.

Khoshnevis, B.: 2004, Automated construction by contour crafting—related robotics and information technologies, *Automation in construction* **13**(1), 5–19.

Khoshnevis, B., Bodiford, M. P., Burks, K. H., Ethridge, E., Tucker, D., Kim, W., Toutanji, H. and Fiske, M. R.: 2005, Lunar contour crafting—a novel technique for isru-based habitat development, *43rd AIAA Aerospace Sciences Meeting and Exhibit—Meeting Papers*, pp. 7397–7409.

Kirsch, U.: 1990, On singular topologies in optimum structural design, *Structural and Multidisciplinary Optimization* **2**(3), 133–142.

Kiyono, C. Y., Vatanabe, S. L., Silva, E. C. N. and Reddy, J. N.: 2016, A new multi-p-norm formulation approach for stress-based topology optimization design, *Composite Structures* **156**, 10–19.

Knutt, E.: 2014, Commercial 3D concrete printing robot, `http://www.bimplus.co.uk/news/exclusive-skanska-build-first-commercial-3d-concre/`. [29 May 2018].

Krassenstein, B.: 2015, Exclusive: How Winsun Stole IP from Contour Crafting and Is "Faking" Their 3D Printed Homes & Apartments, `https://3dprint.com/57764/winsun-3d-print-fake/`. [29 May 2018].

Krassenstein, E.: 2014, World's First 3D Printed Castle is Complete – Andrey Rudenko Now to Print a Full-size House, `https://3dprint.com/12933/3d-printed-castle-complete/`. [29 May 2018].

Langelaar, M.: 2017, An additive manufacturing filter for topology optimization of print-ready designs, *Structural and Multidisciplinary Optimization* **55**(3), 871–883.

Langenberg, E.: 2015, Dreaming of 3D printing 'zero-mile' homes, using local natural materials., `http://www.3dprintingarchitecture.net/?p=521`. [29 May 2018].

Le, C., Norato, J., Bruns, T., Ha, C. and Tortorelli, D.: 2010, Stress-based topology optimization for continua, *Structural and Multidisciplinary Optimization* **41**(4), 605–620.

Leary, M., Merli, L., Torti, F., Mazur, M. and Brandt, M.: 2014, Optimal topology for additive manufacture: a method for enabling additive manufacture of support-free optimal structures, *Materials & Design* **63**, 678–690.

Levy, M.: 1899, Sur l'équilibre élastique d'une plaque rectangulaire, *Comptes Rendus Acad. Sci. Paris* **129**, 535–539.

Lim, S., Buswell, R. A., Le, T. T., Austin, S. A., Gibb, A. G. and Thorpe, T.: 2012, Developments in construction-scale additive manufacturing processes, *Automation in construction* **21**, 262–268.

Lim, S., Buswell, R. A., Le, T. T., Wackrow, R., Austin, S. A., Gibb, A. G. and Thorpe, T.: 2011, Development of a viable concrete printing process.

Lim, S., Le, T., Webster, J., Buswell, R., Austin, S., Gibb, A. and Thorpe, T.: 2009, Fabricating construction components using layered manufacturing technology, *Global Innovation in Construction Conference*, pp. 512–520.

Liu, K. and Tovar, A.: 2014, An efficient 3d topology optimization code written in matlab, *Structural and Multidisciplinary Optimization* **50**(6), 1175–1196.

Love, A. E. H.: 1888, The small free vibrations and deformation of a thin elastic shell, *Philosophical Transactions of the Royal Society of London. A* **179**, 491–546.

Lu, B., Li, D. and Tian, X.: 2015, Development trends in additive manufacturing and 3d printing, *Engineering* **1**(1), 085–089.

MathWorks: 2010, MATLAB version 7.10.0 (R2010a), software program, `https://www.mathworks.com/products/matlab.html`. [29 May 2018].

McNeel, R. and Associates: 2017, Rhinoceros & Grasshopper, software program, `http://www.rhino3d.com/`. [29 May 2018].

Melchels, F. P. W., Feijen, J. and Grijpma, D. W.: 2010, A review on stereolithography and its applications in biomedical engineering, *Biomaterials* **31**(24), 6121–6130.

Moretti, M.: 2015, Concrete beam created with 3D printing, `http://www.wasproject.it/w/en/concrete-beam-created-with-3d-printing/`. [29 May 2018].

Moretti, M.: 2016, Company - World's Advanced Saving Project, `http://www.wasproject.it/w/en/wasp/`. [29 May 2018].

Nádai, A.: 1925, *Die elastischen platten*, Springer.

Nerella, V. N. and Mechtcherine, V.: 2016, Studying printability of fresh concrete for formwork free concrete on-site 3d printing technology, *Conference on Rheology of Building Materials* **1**(1), 1–11.

Newnham, R. E.: 2005, *Properties of materials: anisotropy, symmetry, structure*, Oxford University Press on Demand.

NumPy Developers: 2017, NumPy 1.12.0, software program, `http://www.numpy.org/`. [29 May 2018].

Oasys: 2017, GSA 9.0, software program, `http://www.oasys-software.com/products/engineering/gsa-suite.html`. [29 May 2018].

Osher, S. and Sethian, J. A.: 1988, Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations, *Journal of computational physics* **79**(1), 12–49.

Oxman, N.: 2012, MIT - About the Mediated Matter Group, `http://matter.media.mit.edu/about`. [29 May 2018].

Oxman, N.: 2016, MIT- Digital Construction Platform (DCP) v.2, `http://matter.media.mit.edu/tools/details/digital-construction-platform-dcp`. [29 May 2018].

Oxman, N., Duro-Royo, J., Keating, S., Peters, B. and Tsai, E.: 2014, Towards robotic swarm printing, *Architectural Design* **84**(3), 108–115.

París, J., Navarrina, F., Colominas, I. and Casteleiro, M.: 2010, Block aggregation of stress constraints in topology optimization of structures, *Advances in Engineering Software* **41**(3), 433–441.

Pegna, J.: 1997, Exploratory investigation of solid freeform construction, *Automation in construction* **5**(5), 427–437.

Pereira, J. T., Fancello, E. A. and Barcellos, C. S.: 2004, Topology optimization of continuum structures with material failure constraints, *Structural and Multidisciplinary Optimization* **26**(1), 50–66.

Python Software Foundation: 2018, Python 3.6.5, software program, `https://www.python.org/`. [29 May 2018].

Querin, O. M., Young, V., Steven, G. P. and Xie, Y. M.: 2000, Computational efficiency and validation of bi-directional evolutionary structural optimisation, *Computer methods in applied mechanics and engineering* **189**(2), 559–573.

Rozvany, G. I. N.: 2009, A critical review of established methods of structural topology optimization, *Structural and Multidisciplinary Optimization* **37**(3), 217–237.

Rozvany, G. I. N. and Birker, T.: 1994, On singular topologies in exact layout optimization, *Structural and Multidisciplinary Optimization* **8**(4), 228–235.

Rudenko, A.: 2016a, 3D Concrete House Printer - About Us, `http://www.totalkustom.com/about-us.html`. [29 May 2018].

Rudenko, A.: 2016b, 3D Concrete House Printer - Rudenko's 3D Printer, `http://www.totalkustom.com/`. [29 May 2018].

Sakin, M. and Kiroglu, Y. C.: 2017, 3d printing of buildings: Construction of the sustainable houses of the future by bim, *Energy Procedia* **134**, 702–711.

Salet, T. A. M.: 2016, TU/e - 3D Concrete Printing, `https://www.tue.nl/en/university/departments/built-environment/research/research-programs/structural-design/research/research-areas/concrete-research-areas/3d-concrete-printing/`. [29 May 2018].

SciPy Developers: 2018, SciPy 0.19.1, software program, `https://www.scipy.org/`. [29 May 2018].

Sigmund, O.: 1997, On the design of compliant mechanisms using topology optimization, *Journal of Structural Mechanics* **25**(4), 493–524.

Sigmund, O.: 2007, Morphology-based black and white filters for topology optimization, *Structural and Multidisciplinary Optimization* **33**(4-5), 401–424.

Sigmund, O. and Petersson, J.: 1998, Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima, *Structural optimization* **16**(1), 68–75.

Slager, G. J.: 2017, *Influence of the interface between layers on the tensile properties of 3d printed concrete*, Master's thesis, Eindhoven University of Technology.

Svanberg, K.: 1987, The method of moving asymptotes—a new method for structural optimization, *International journal for numerical methods in engineering* **24**(2), 359–373.

Svanberg, K.: 1995, A globally convergent version of mma without linesearch, *Proceedings of the first world congress of structural and multidisciplinary optimization*, Vol. 28, Goslar, Germany, pp. 9–16.

Sved, G. and Ginos, Z.: 1968, Structural optimization under multiple loading, *International Journal of Mechanical Sciences* **10**(10), 803–805.

SymPy Development Team: 2018, SymPy 1.0, software program, `http://www.sympy.org/nl/index.html`. [29 May 2018].

Thompson, C.: 2015, WASP builds 40-foot 3d printer to build clay homes, `http://www.techinsider.io/wasp-builds-40-foot-3d-printer-to-build-clay-homes-2015-9`. [29 May 2018].

Timoshenko, S. P. and Woinowsky-Krieger, S.: 1959, *Theory of plates and shells*, McGraw-hill.

Tissink, A.: 2017, Betonprinten lijkt definitief de kliederfase voorbij, `https://www.cobouw.nl/bouwbreed/nieuws/2017/05/betonprinten-lijkt-definitief-de-kliederfase-voorbij-101248238`. [29 May 2018].

Verbart, A., Dijk, N., Tin, L. D., Langelaar, M. and Keulen, F.: 2011, Effect of design parameterization and relaxation on model responses in topology optimization with stress constraints, *Proceedings of the 9th World Congress on Structural and Multidisciplinary Optimization*.

Verbart, A., Langelaar, M., Van Dijk, N. and Van Keulen, F.: 2012, Level set based topology optimization with stress constraints and consistent sensitivity analysis, *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials*

Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, p. 1358.

Wang, F., Lazarov, B. S. and Sigmund, O.: 2011, On projection methods, convergence and robust formulations in topology optimization, *Structural and Multidisciplinary Optimization* **43**(6), 767–784.

Wang, M. Y., Wang, X. and Guo, D.: 2003, A level set method for structural topology optimization, *Computer methods in applied mechanics and engineering* **192**(1), 227–246.

Wheeler, A.: 2015, Cybe be 3d printing concrete walls, `http://3dprintingindustry.com/news/cybe-3d-printing-concrete-walls-39422/`. [29 May 2018].

Xie, Y. M. and Steven, G. P.: 1993, A simple evolutionary procedure for structural optimization, *Computers & structures* **49**(5), 885–896.

Xie, Y. M. and Steven, G. P.: 1997, *Basic evolutionary structural optimization*, Springer.

Yang, R. J. and Chen, C. J.: 1996, Stress-based topology optimization, *Structural optimization* **12**(2-3), 98–105.

Yang, X. Y., Xei, Y. M., Steven, G. P. and Querin, O. M.: 1999, Bidirectional evolutionary method for stiffness optimization, *AIAA journal* **37**(11), 1483–1488.

Zhou, M. and Rozvany, G.: 2001, On the validity of eso type methods in topology optimization, *Structural and Multidisciplinary Optimization* **21**(1), 80–83.