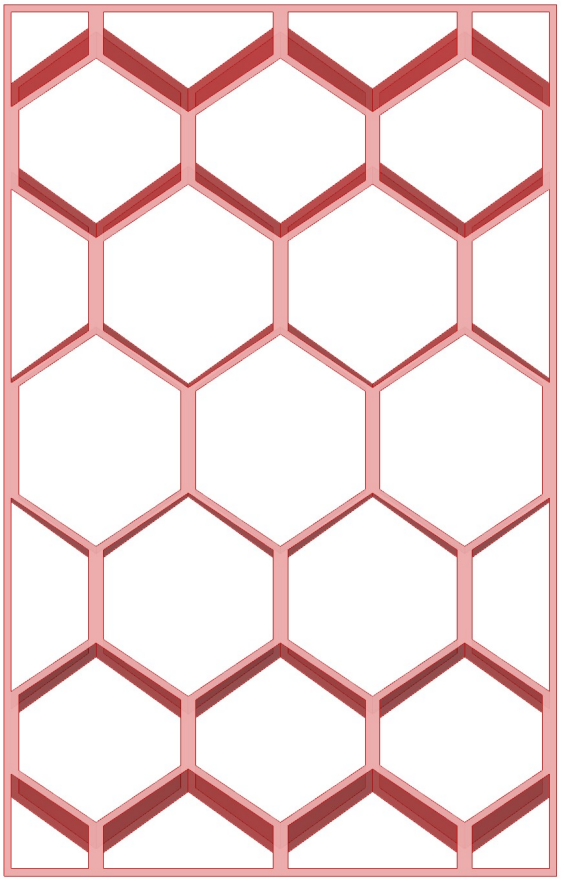
Delft University of Technology

Master Thesis

Multi-criteria optimisation model for glass facade element with 3D printed structure



Author:
Max Driessen

Supervisors:
Prof. dr. ir. P.C. Louter
Dr. ir. M. Nogal Macho
Dr. ir. F. Messali
Dipl. ing. Daniel Pfarr

5th December 2023

Abstract

This thesis investigates the usability and creation of a multi-criteria optimisation tool for a facade element consisting of two outer glass layers and a 3D-printed core. The chosen constraint is related to the structural requirements laid down by the NEN codes based on the ULS and SLS. The criteria are: the transparency criterion, which maximises the transparency of the panel; the insulation criterion, which minimises the U-value; and the daylight criterion, which maximises the indirect sunlight at the summer solstice and the total sunlight during the winter solstice.

Topology optimisation is the first considered optimisation algorithm that allows for complete freedom of the core design through discritisation of the core volume with finite elements. Nevertheless, topology optimisation is not considered usable in this tool as a result of the inability to add relevant criteria to topology optimisation integrated into software, the complexity of developing an optimisation tool with topology optimisation in Python or Matlab, and the intricate optimal core designs, which are not easily printable.

Next, a hexagon structure is used as the core design with some related variables, and it is created in Rhino 7. The daylight criterion is performed with the Ladybug library, and a mesh sensitivity study is conducted. The structural constraint uses Karamba3D, and Karamba3D's accuracy is investigated with validation with Diana and stiffness experiments. The experiments revealed a large discrepancy between the stiffness observed in the experiments and the predicted stiffness in the Karamba3D and Diana models. The insulation and transparency criteria are added, and the genetic algorithm is used to determine the near-optimal solution. Finally, a case study based on the Echo building is used to compare the near-optimal sandwich panels with regular windows. Although the computational demand inhibits quick results, the sandwich panel performs well compared to the original window in relation to the three criteria.

Acknowledgements

I would like to express my gratitude to my committee in no particular order. Prof. dr. ir. P.C. Louter was the chairman and explained the process of conducting sizeable research, and he increased my understanding in glass used for structural purposes. Next, dr. ir. M. Nogal Macho provided insightful guidance in all parts related to the optimisation process, and dr. ir. F. Messali supported me in creating a comprehensive validation study, designing Diana models, and evaluating topology optimisation. Finally, dipl. ing. Daniel Pfarr has greatly supported me in building the optimisation tool in Rhino Grasshopper, giving me the ability to 3D print some geometries and providing a starting point for the tool through his previously developed Rhino model that creates honeycomb patterns.

Subsequently, I would like to thank prof. dr. ir. M. Langeraar for providing a starting point for the research in topology optimisation and dr E. Brembilla for providing an introduction to the importance of natural light in buildings and formulating an optimisation criterion. To conclude, this thesis is a continuation of the previously conducted studies and could not have been created without the work of Brugman [25], Neeskens [54], Saleh [65] and van der Weijde [74].

List of Figures

List of Tables

Table of contents

1 Introduction

Glass is currently one of the main building materials, and in modern architecture, glass structures are more prominent compared to older buildings. The usage of glass can be traced back to the Roman Empire, where it was used in small objects, such as jewels. In the Middle Ages, glass was used in buildings for the first time. These windows were small and only brought in a small amount of natural light. During the second half of the first millennium, the window size increased as glass became more available [25]. The subsequent increase in glass usage came during the Industrial Revolution, which made glass available to the masses. This resulted in an enormous increase in its demand and consequently, an increased interest in glass production techniques. This interest focused on increasing the production speed and window size, which led to the invention of new production methods. The main advantage of glass in structures and the driver of its increased popularity is its transparency. Furthermore, studies concluded that natural light is beneficial for health and productivity [32].

Many different types of glass exist, as each application requires different qualities. For example, soda-lime glass is the dominant type of glass in the building industry. Additionally, in the past decades, different production processes have emerged, such as laminated and tempered glass, as well as the usage of different coatings to enhance curtain properties [59]. Furthermore, current building regulations demand certain insulation properties. For example, the Dutch building code of 2012 demands a U-value smaller than $1.65\,W/m^2\,K$. This has led to the use of multiple glass layers in windows. Nowadays, most windows consist of two to three glass layers, where the glass panels have a thickness between 4 and 6 mm [53]. However, in the near future, the regulations require a U-value that must be smaller than $0.6\,W/m^2\,K$ [25]. This will require, in general, three or four glass layers in windows. The use of more glass layers will increase the use of raw materials, and the required energy for glass production is relatively high. For example, the production of soda-lime glass requires around 10 times more energy than concrete based on Portland cement [16]. Furthermore, as the window weight increases, the surrounding structure has to be stronger, which also requires additional materials.

This development is especially troublesome now, as one of the largest challenges of the twenty-first century is climate change. This drove the change to sustainable energy sources, and limiting or improving the efficiency of current energy usage. The consecutive consequences for the construction sector resulted in a focus on reducing emissions in the creation, lifetime, and demolition of constructions. Furthermore, creating windows with limited insulation capabilities is no option, as this will increase the energy required to control the indoor climate of the buildings as well as increase the cost for their inhabitants. This last argument has become clear with the recent energy crisis. At last, the human population is growing and is expected to grow to ten billion in the coming decades, which creates the need for additional buildings [1]. Subsequently, overall prosperity increases, which further enhances the demand for glass.

Thus, an alternative to the standard windows is required, which requires fewer raw materials, demands less energy, needs less heaviness. An option is thin aluminosilicate glass with a maximum thickness of two mm. This type of glass requires around 1.5 times more energy than soda-lime glass; however, the thickness is much smaller, which results in an overall lower energy usage [25]. Thereby, the reduced thickness creates a lighter structure. However, this has the disadvantage of relatively high flexibility compared to regular window glass, while the deformations are limited by the building regulations. Thus, the glass panels need to be stiffened. The thin glass can be stiffened with the use of a 3D printed structure, also known as additive manufacturing (AM) that is designed to optimally bear the load with minimum material usage. This configuration is called a sandwich panel. As a result of the weight reduction, transportation costs decrease, as do carbon dioxide CO_2, emissions. Moreover, the printed structures create novel functions of the facade element compared to a regular window, such as daylight control, as the sunlight can be reflected in a beneficial way, and the ability of architects to use numerous designs instead of using regular windows [49, 51].

Several studies on this facade element have been conducted in recent years. One of the earlier reports is from 2017. This research explores the applicability of thin glass facade elements with a 3D printed structure and the advantages over regular windows. The discussed advantages are weight reduction, improved insulation, and daylight control [74]. Another report from 2018 investigates the use of combining thin glass, produced from aluminosilicate with a maximum thickness of two mm, and a 3D printer structure to enhance the load-bearing efficiency of the facade element [54]. One of the recommendations after this study is the creation of a tool that calculates the optimal topology of the printed structure. The third report, published in 2019, covers the durability of the facade element, made of thin glass and a 3D printed structure of recycled PET, polyethylene terephthalate plastic, related to UV, ultraviolet, radiation, and fire safety [65]. Finally, a report from 2019 about the thermal insulation and structural properties of three different topologies of the facade element made of thin glass and a 3D printed structure [25].

1.1 Problem statement

The windows have to increase their insulation properties, while limiting their raw material usage, energy requirements, and weight. In order to meet these criteria, an alternative has to be found for the regular windows that are currently widely used. An alternative is to use thin aluminosilicate glass with a 3D printed core. This sandwich panel has shown promising results in previous research. This difference in the sandwich panel in comparison with the regular window design is captured in Figure 1, and some examples of the core geometry are shown in Figure 2.

However, determining the optimal design of the 3D printed core remains unknown. Thereby, the capability to shape the core in different designs creates the possibility to add other criteria that reflect the desired functionalities of the panel, for example, daylight control and transparency.

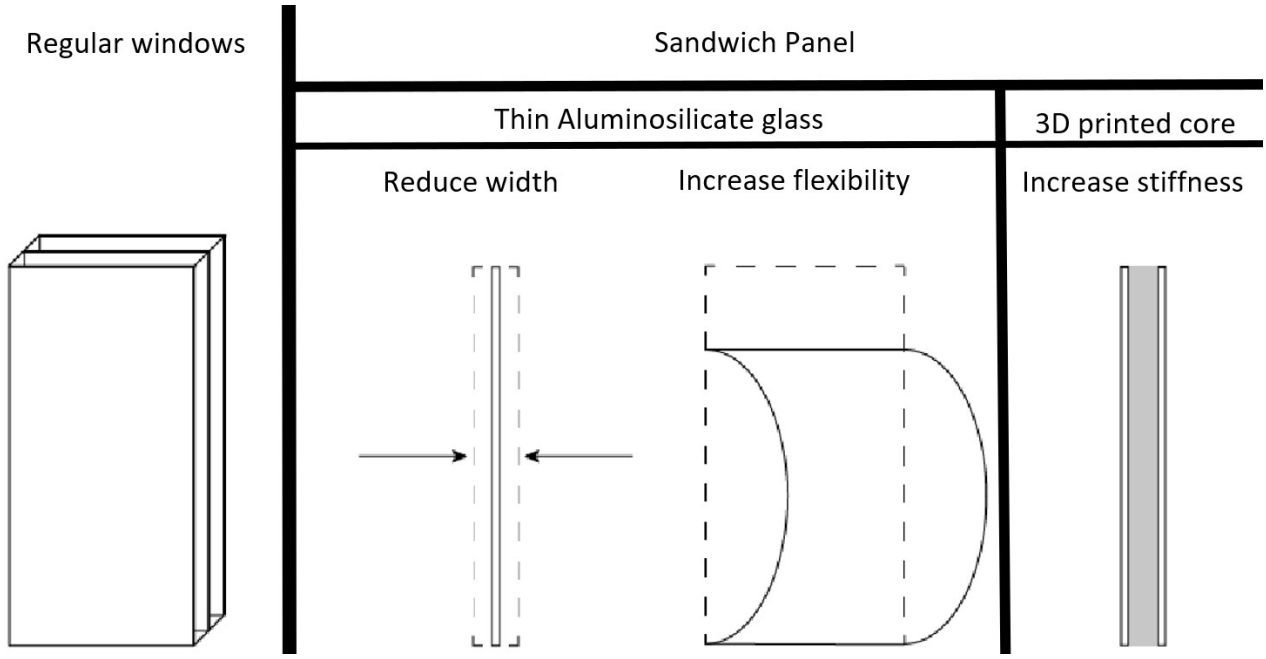


Figure 1: Idea behind thin aluminosilicate glass with 3D printed core [25].

Figure 2: Examples of different designs of the core [11, 54, 74].

1.2 Objective

The objective of this thesis is the creation of and research on the development of an optimisation tool for the design of the facade element. The optimisation tool will consider multiple objectives and constraints, which are in reference to the documented functionalities of the panel in the recent literature. The criteria consist of the transparency, the insulation, and the daylight quality. The daylight quality considers the beneficial influence of the core geometry on the natural light. The only constraint takes the structural safety into account. The main question, therefore, is stated as follows:

How to develop a multi-criteria optimisation tool for a glass facade element with a 3D-printed core?

In order to answer this question, multiple aspects have to be considered. First, it is important to note that the considered facade panels will consist of two glass layers connected with a 3D-printed core, also known as a sandwich panel. In theory, more glass layers and 3D-printed cores can be used, which provide specific advantages and disadvantages; however, this is not part of this thesis. Subsequently, before the aspects of the multi-criteria model are considered, a brief introduction to sandwich panels is provided. This explains the advantages of a sandwich panel and the application of such a panel in this market. Thus, the first subquestions is:

What is a sandwich panel and its application for windows?

Next, the most important requirement of a window, or something with similar purposes, is structural safety. This implies that the window must be able to bear the loads to prevent structural failure and unease within the community. The most obvious load is the wind load, as it is placed on the outside of a building; however, the confined air layer between the glass layers creates additional forces. Additionally, it is used that the facade panel is a non-load-bearing construction element, which means that it is not part of carrying the loads from other above-positioned construction elements, such as the roof. The structural safety is researched and will provide an overview of the relevant loads and restrictions on stresses and deflections. As these guidelines are defined by law, only Dutch guidelines are used, and the question is stated as follows:

What are the required structural properties of a facade element?

The subsequent step is to investigate the insulation characteristics of a sandwich panel. This will lead to a formula that gives an estimation of the insulation properties. This is mainly based on the thesis of Brugman [25] and the subquestion is formulated as:

How to determine the insulation characteristics?

The other criterion of the tool is related to the natural light coming in through the facade panel. The effects of natural light are investigated, and a division between direct and indirect sunlight has been made. The perception of the effects of natural light is partly climate-sensitive, such as direct sunlight. Therefore, the perception is based on climates similar to the Dutch climate. The question answered is:

What is the role of natural light in the building environment?

As all the criteria and constraints are currently discussed, the next step is determining the optimal design of the core geometry. A frequently-used method in structural engineering is called topology optimisation. This method is based on discretising the geometry into smaller elements to predict the behaviour of the structure under certain boundary and loading conditions. Therefore, the topology optimisation theory is first explored to provide a solid foundation before exploring the available applications that use topology optimisation. The considered applications are written codes in Python or Matlab that allow for great freedom in altering the process and fitting it to this specific panel, although they require a good understanding of programming. The other application is using a program that already has topology optimisation embedded into its software, such as ANSYS. Finally, in a few cases, both applications can be used intertwined to benefit from the advantages of both applications. Then, a test is performed in ANSYS to investigate the usability for the optimisation tool, which looks into a combination of objectives and constraints as well as boundary conditions. The following question will be answered:

How to determine an optimal design of the core geometry with topology optimisation?

Beside the possible incorporation of topology optimisation, different optimisation algorithms are explored to discover the best suitable optimisation algorithm for this tool. The considered optimisation algorithms are the linear and nonlinear models and the (meta)heuristics models. In the end, an optimisation algorithm is chosen, and the hereafter mentioned question is answered.

What is the best optimisation algorithm for this application?

After exploring all the different components that are used in a multi-criteria optimisation tool, the tool itself will be developed. The tool is based on the work of Pfarr and Louter [61] which uses a honeycomb pattern as the core design and was created in Rhino Grasshopper. This design is used as the basis of the core, and some variables within this pattern are added to optimise for. For example, the number of hexagons can vary, and an inclination of the pattern over the thickness can be incorporated with different angles. First, the basis of the model is explained, which comprises the input parameters, the variables and constraints, and the objectives and fitness function. Subsequently, the inner workings of the Rhino model are explained, which explains all the required steps and components used to create the design of the facade element and optimise it.

Thereafter, in order to calculate the structural behaviour of the panel, Karamab3D is used, which is validated with a Finite Element Analysis in DIANA. The validation also takes into account the computational time and which structural model provides sufficient accuracy while minimising the calculation time. Additionally, two core patterns are printed and tested in order to compare the structural models with reality. This is mainly performed to estimate the magnitude of the error when the structural models assume a perfect connection between the core and glass layers. Subsequently, the daylight criterion uses the Ladybug components, which require additional insight into the result sensitivity to the applied mesh and a sensitivity analysis relating to the natural light.

Finally, as the optimisation tool is constructed and validated, a case study is used to study the usability of the tool in a real-life situation as well as explore the received optimal patterns in different simulations. All this information is needed to answer:

How to create a multi-criteria optimisation tool?

1.3 Outline

The second chapter gives an introduction to the theory behind a sandwich panel and explains how this facade panel is related to it. It provides an overview of the recent literature published on this specific panel and elaborates on its findings. This chapter answers the first sub-question. The subsequent chapter, answering the second sub-question, sets out the structural requirements for a facade panel, which are based on the Dutch NEN codes. It first covers the wind load calculations used in the Netherlands and continues to determine the cavity pressure that arises because of the enclosed air between the glass sheets. Lastly, it explains the procedure for load combinations and load factors as well as the two assessments: extreme limit state and serviceability. The fourth and fifth chapter are dedicated to answering the subquestion related to insulation and natural light, respectively.

The following three chapters try to answer the subquestion related to topology optimisation. The first of the three chapters investigates the theory behind topology optimisation, while the second chapter provides an overview of the different applications. These applications are divided into freestanding Matlab or Python codes, integration in software, and a combination of codes and software. These applications are compared based on their usability for this specific case. The last of the three chapters conducts an experiment with one of the promising applications (ANSYS) to further investigate the usability of such an application. Next, in the ninth chapter, answer the subquestion to find the best optimisation algorithm for this case. It briefly covers the different optimisation methods, such as linear, nonlinear, and heuristic models.

As all the required information is collected to build an optimisation tool, the next six chapters are focused on the last question. The first chapter explains the basic composition of the tool. It covers the used variables and input parameters, the constraints, and the objectives and optimisation function. The next chapter provides the basic inner workings of the tool created in Rhino. It explains which steps are needed to start with the variables and input, then obtain the core design, and additionally determine the optimal design. A more detailed explanation of the Rhino model is provided in Appendix H. The subsequent chapter validates the structural model of Karamba3D used in Rhino to determine the deflection and stresses. This model uses shell elements and, therefore, is compared with a Diana model with shell elements and another with solid elements. Diana is a finite element program. After validating Karamba3D, two designs have been printed, and a bending test is performed. These results are then compared with Diana models. This chapter investigates the effect of the assumptions or simplifications made in the digital models, where the main simplification is to assume a perfect connection between the core and glass layers.

The fourteenth chapter covers a mesh sensitivity study of the Daylight criterion, as this criterion uses Ladybug components, which use a mesh of sensors to estimate the incoming daylight through the facade panel. The last of the six chapters is a case study to understand how the tool will perform in practice. The case study is based on a campus building, Echo, at the Technical University in Delft. Finally, the overall conclusion and discussion will provide an answer to the main question and reflect on the research.

2 Sandwich panel

The various components of the facade element, as well as the theory behind the sandwich panel, are briefly discussed in this chapter. Finally, the recent literature is discussed regarding the facade element.

2.1 Sandwich panel theory

A sandwich panel has two outer layers and a relatively low-density layer in between. The outside layers are generally thin compared to the total thickness of the sandwich panel, while the core is relatively thick but has a lower density. The core can be made from a wide range of materials, for example, metal, wood, foams, and synthetics. The sandwich panel has a relatively high bending stiffness and strength, as shown in Figure 3 [34].

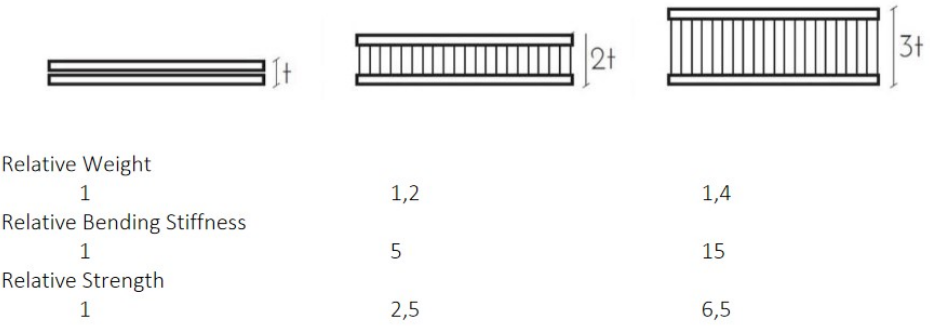


Figure 3: Development of the panel's characteristics as the center layer thickens [25].

2.1.1 Manufacturing process

The sandwich panels are constructed by bonding the two outer layers to the core. Different processes are used for bonding the layers together; for example, the glueing process of a honeycomb sandwich panel is done by mould processing, heated pressing, or vacuum bag processing [37]. The mould processing technique is used mostly for batch production, where heated pressing is used, in general, for flat and simple geometry panels. At last, the vacuum bag process can be used for panels with complex and possibly curved geometries [74].

When glass panels are used, chemical joining is the preferred technique [76]. For example, adhesive bonding, of which the most common types are acrylate, epoxy, polyurethane, and silicon [50]. A small overview of their properties is shown in Table 1. Additionally, the paper found that acyrlate has the best mechanical performance considering temperature. Moreover, the structural failure occurred in the glass, thus, the adhesive is not the limiting factor. A more in-depth review and additional results can be found in the paper of Louter et al. [50].

Adhesive type	Shear strength [MPa]	Curing time	Colour	Gap-filling capacity [mm]
Acrylate	23	30 to 60 seconds	Transparent	$<$0.1
Epoxy	18	$>$10 hours	Grey	Up to 5
Polyurethane	7	$>$ 8 hours	Transparent	0.05 to 0.1
Silicone	1.06	$>$ 7 days	Black	6 to 15

Table 1: Properties of mostly used adhesive types with glass panels [50].

Another chemical bonding technique is laminated bonding, where foils of polyvinyl butyrate, PVB, are mostly used in the application of structural glass, although SentryGlass, SG, has the highest strength and stiffness. PVB exhibits ductile behaviour and the standard thickness starts at 0.38 mm. These layers have time- and temperature-dependent strength as well as a different thermal expansion coefficient than glass [74]. An overview of the properties of PVB and SG is provided in Table 2. More information and a comparison between the most commonly used interlayers can be found in the paper of Schittich [67] and Santarsiero [66].

	Density $[g/cm^3]$	Thickness [mm]	Tensile strength [MPa]	Shear modulus [GPa]	Poisson ratio
PVB	1.07 to 1.08	0.38, 0.76	20	0 to 4	0.45 to 0.49
SG	0.95	0.72, 0.89, 1.52, 2.28	34.5	80 to 200	0.44 to 0.5

Table 2: Properties of PVB and SG [66].

2.2 Outerlayer sandwich panel as thin glass

The term "thin glass" is used when a glass sheet is thinner than 2 mm and thicker than 0.1. The use of thin glass is mainly situated in the mobile electronics sector; however, an increase in interest has developed in the building industry in recent years. For example, research by Simoen, C., to demonstrate the feasibility of thin glass in the building industry, and a study by Ribeiro Silveira, R., to investigate the behaviour of thin glass in applications inside the building industry [63, 70].

Thin glass sheets are typically made of aluminosilicate glass, whereas thicker glass sheets are typically made of soda-lime glass [54]. Table 3 shows the material properties of both glass types. In general, aluminosilicate glass has greater strength and stiffness than soda-lime glass.

	Soda-Lime glass	Aluminosilicate glass
Density [$1000kg/m^3$]	2.44 to 2.49	2.49 to 2.54
Young's modulus [GPa]	68.2 to 71.7	84.8 to 89.1
Shear modulus [GPa]	28 to 29.5	34.3 to 36
Yield strength [MPa]	31 to 34.2	39.9 to 43.9
Tensile strength [MPa]	31 to 34.2	39.9 to 43.9
Compressive strength [MPa]	310 to 342	339 to 439
Flexural strength [MPa]	40 to 44.5	51.9 to 57.1
Fatigue strength [MPa]	29.4 to 32.5	37.8 to 41.8
Poisson's ratio [-]	0.21 to 0.22	0.23 to 0.24
Thermal expansion coefficient	9.16 to 9.53	4.11 to 4.28

Table 3: Properties of soda-lime and aluminosilicate glass [54].

The thin glass falls within the float glass category. This category is the most used glass type in glass panels because it can be used for mass production. The technique was developed in the middle of the twentieth century by the glass manufacturer Pilkington. The process is illustrated in Figure 4. Although thin glass is float glass, the general thickness is between 2 and 19 mm. Therefore, an alternative production process is used to accommodate the smaller thickness. This is called the micro-float process. Subsequently, additional production processes can be used to create glass panels for specific purposes, such as pre-stressing, lamination, mechanical processing, coating, and bending [54, 74]. In the case of thin glass panels, the only viable strengthening method is chemical strengthening. This process creates surface compression, which increases the ability to cope with bending. As it is shown in Table 3, the tensile strength is much lower than the compressive strength [54].

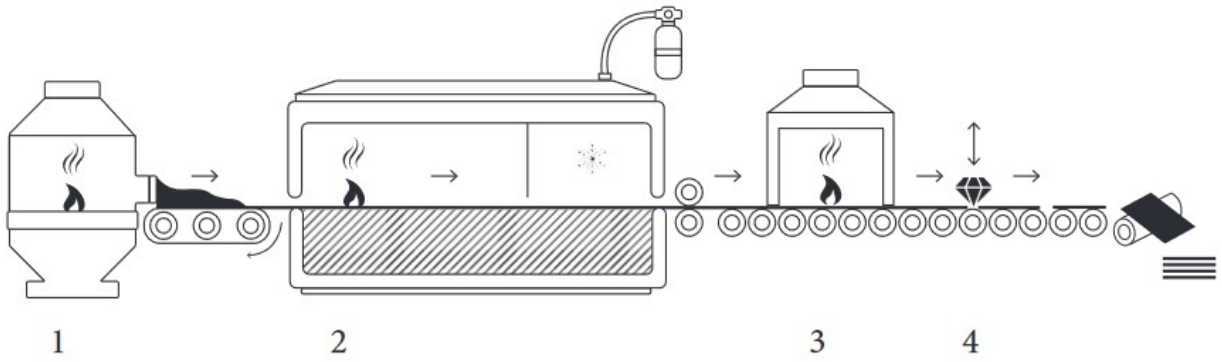


Figure 4: Floating glass technique with the melting process shown at 1, floating at 2, annealing at 3, and cutting at 4 [74].

2.3 Core sandwich panel

The core of a sandwich panel can be made from different materials, some of which are discussed below. Next, different geometries are possible, and some of the most noticeable designs are highlighted.

2.3.1 Material

A widely used material is aluminium, which has a small density and a good strength-to-weight ratio. Additionally, the material allows heat transfer, which may be desirable in some cases. Furthermore, aluminium is a relatively cheap material [37].

Subsequently, the core can be made from a composition that uses a type of resin, such as aramid, impregnated paper, or glass polyimide. The paper core is not common in structural applications, and the aramid and glass polyimide provide good thermal insulation, although the glass polyimide core is an expensive option, which limits the number of applications [74]. Next, stainless steel is used in applications that require a high level of UV resistance as well as resistance to corrosive environments. However, the main disadvantage is its relatively high density compared to the other core materials.

Also, different types of plastics are used, which offer the opportunity to use different colours or create transparent cores. The main downside of plastics is the effect of UV light and the temperature at which it melts or yields, which could create challenges in the production process. Finally, additive manufacturing can be used, which can be used in combination with topology optimisation. This creates a seamless connection between a digital design and the actual design, as well as options to use different materials. However, printability should be taken into account, which limits the number of designs.

In Table 4, different core materials and their properties are shown. These core materials were used in previous literature, which will be discussed in this chapter. PETG, Polyethylene terephthalate glycol, is similar to PET, but glycol is added to modify the properties. Additive manufacturing changes the material properties by reducing the overall stiffness and strength, which is demonstrated in the thesis of Saleh [65].

Material properties	PETG	PET
Density $[kg/m^3]$	1260	1260 to 1280
Young's modulus [GPa]	2.1	2.8 to 3.0
Poisson's ratio	0.4	0.4
Tensile strength [MPa]	unknown	60 to 66
Compressive strength [MPa]	unknown	57.5 to 63.5
Shear modulus [GPa]	unknown	0.716 to 0.752

Table 4: Properties of PETG and PET [31, 54].

2.3.2 Geometry

The core can have an ample number of geometries, especially with additive manufacturing. A selection of geometries is discussed that create a lightweight and mostly transparent facade element [54]. The space frame, Voronoi pattern, foam structure, and topology optimisation are considered.

The first geometry type is a space frame, which consists of triangular shapes and is also known as a truss-like structure. This shape is widely used in the building industry because of its efficient structural load-bearing capacity. The truss structure can also be a good starting point for further optimisation of the geometry. Examples of the space frame pattern are visible in Figure 5.

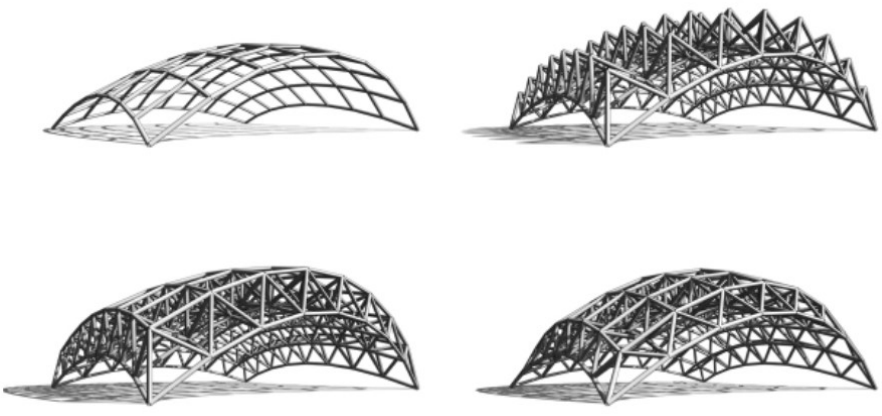


Figure 5: Different curved space frames [54].

Another possible shape is the Voronoi structure, which is visible in figure 6. The pattern is a result of forcing mutual distance between points in a defined point cloud and can be made in 2D or 3D [54]. The pattern has an irregular shape, which could be interesting for architects. Moreover, the point cloud determines the final shape, and incorporating design parameters into the point cloud can result in an optimal design. Subsequently, foam structures can be used, which are very similar to the Voronoi pattern. An example is shown in Figure 7. The difference between the two patterns is that the Voronoi pattern has three members per node, while the foam structure has four in general. However, a foam structure is mostly a soft structure, which has to be taken into consideration as the core needs to increase the overall stiffness.

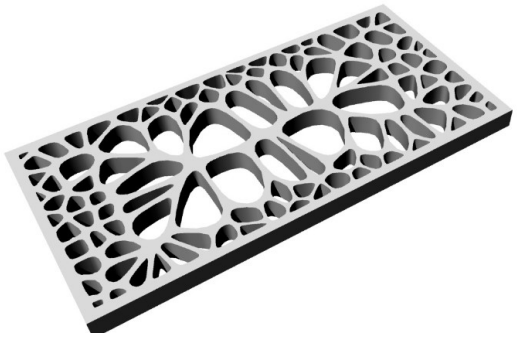


Figure 6: An example of a 2D Voronoi pattern [54].

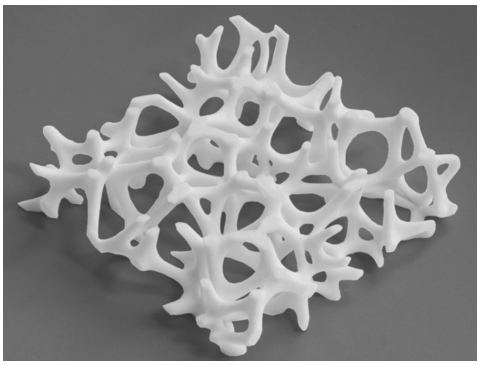


Figure 7: An example of a foam structure [54].

At last, topology optimisation can be a method to create the optimal design. The process of optimisation is based on the defined boundary conditions, load, objectives, and constraints. Therefore, this method creates an optimal and unique design per case. However, the main disadvantage is the high computation time [19]. The process is depicted in Figure 8, where the left-hand side represents the initial design and the optimal design is shown on the right-hand side.

Figure 8: The procedure of topology optimisation [54].

2.3.3 Additive manufacturing

One method of manufacturing the core structure is additive manufacturing, also called 3D printing. This method has been mainly used in the recent research on this facade element and therefore will be explained briefly.

3D printing is used to create 3D geometries by building them layer by layer. This technique first appeared in 1983 and was invented by Chuck Hull. While the popularity of 3D printing increased rapidly in recent decades, the 3D printers became more accessible to a variety of users, such as individuals, companies, and educational institutions [23]. A widely used method of printing is FDM (fused deposition modeling), which is also used in the thesis of Saleh [65]. The process is displayed in Figure 9, where PLA is a type of plastic. This could be replaced by another type of plastic, such as PET or PETG.

Despite the advantages of creating complex designs, having relatively low production costs, and limiting waste, there are some disadvantages. For example, limited size, long printing time, high energy consumption, and uncertainties about the long-term structural behaviour [17, 60].

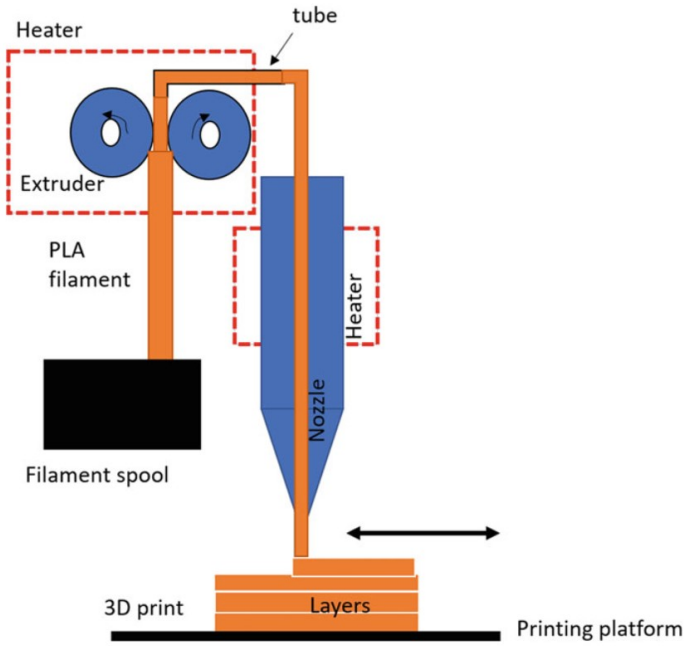


Figure 9: The principle of FDM [52].

2.4 Facade panel consisting of thin glass and plastic core

The facade element containing thin glass layers and an added core to increase the stiffness, also called a sandwich panel, was researched in 2017 in a master's thesis by Van der Weijde, I.[74]. The thesis investigated the capabilities of the facade element related to stiffness, insulation, safety requirements, and the design of such a facade panel. A honeycomb geometry was used as the core, as shown in Figure 10.

This research involved multiple experiments. One of the tests was to determine the best bonding material without compromising the transparency of the facade panel. The tests considered acrylic adhesive bonding and SG laminated bonding. Both bonding techniques were performed with different cell sizes, and the SG was also tested with a polycarbonate core. It concluded that the laminated bond provided the best adhesion, and all bonding techniques prevented full transparency caused by the lens effect.

Secondly, the performed bending tests compared the difference in structural behaviour between a single glass sheet of 1 mm, a double glass sheet of 2 mm combined, and a sandwich panel with an aramid honeycomb core structure of 10 mm with annealed glass of 1 millimetre and chemically strengthened glass of 2 mm. The sandwich panel had, as expected, an undoubtedly higher failure load. Thereby, the chemically strengthened glass showed ductile behaviour, while the other were brittle. A complete overview can be viewed in Chapter 7 [74]. At last, a case study was performed, which also considered the insulation of the facade panel.

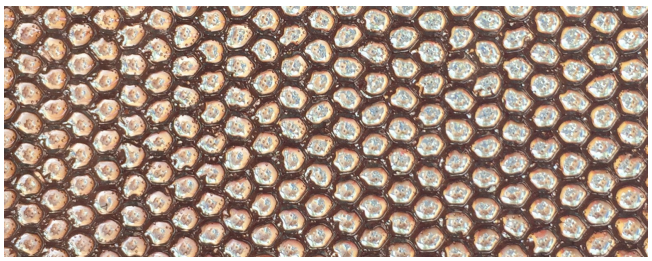


Figure 10: Honeycomb pattern tested in the thesis of van der Weijde [74].

Neeskens (2018) wrote the subsequent paper on this facade [54]. The research question was stated as: "How can thin glass and a 3D printed spacer pattern work together to create a stiff and self-supporting sandwich panel that increases the structural efficiency of the material?". This research focused more on additive manufacturing and finding the optimal geometry of the core structure compared to the previous research. It provides an extensive overview of different additive manufacturing processes as well as a comparison of the most commonly used materials. This can be read in Chapter 2.2. Furthermore, different geometries are discussed as possible

optimal designs for the core structure. This is discussed in Chapter 2.3.2. Subsequently, multiple panel designs are generated using the Voronoi pattern. The research concluded that it was the best design choice because of the sufficient surface area to connect the core and the outside layers, its stiffness, and material usage for support [54]. The designs were first tested with a finite element analysis and then a physical bending strength test. PETG is used as a core material. The results can be found in Chapter 5 of the thesis, and one of the used designs is shown in Figure 6. Additionally, a Lloyd's algorithm was used for some designs, which created a more symmetrical design, which would be a good design because the structural behaviour is symmetrical along two axes in the case of two or four edge supports. An example of this design is given in Figure 11. The recommendations after the report include the optimisation of the Voronoi design, studying the thermal insulating behaviour and fire safety, and researching a curved panel.

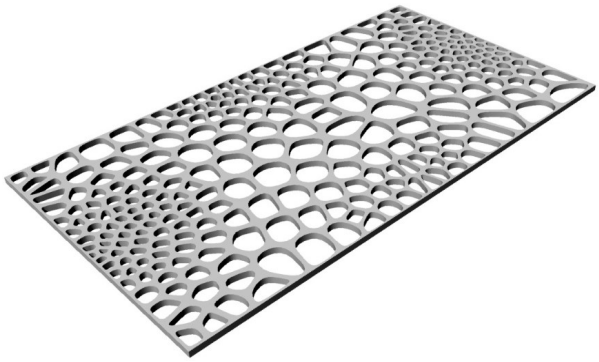


Figure 11: Symmetrical Voronoi pattern [54].

The next thesis from 2019 by Saleh [65], had the objective to understand the durability of the facade panel related to UV radiation, elevated temperatures, and fire safety. Recycled PET is used as a core material. The core geometry that is used is the honeycomb structure and the space truss pattern, visible in figures 12, 13. The FEA concluded that the honeycomb pattern results in lower deformations and lower shear stress compared to the space pattern. Both patterns have almost the same weight. Subsequently, a heatflow analysis was done, which showed similar behaviour for both patterns.

Additionally, physical tests were performed on 3D-printed panels. For example, a 3-point bending test was done. First, only the core was tested, which showed that the maximum force of the honeycomb pattern is roughly 40 % higher than the space pattern and behaves stiffer. Subsequently, the test was conducted with the glass panels attached to the core. This also showed that the honeycomb pattern is stronger.

Furthermore, a 3-point bend test at elevated temperatures was performed, which revealed that the space pattern behaved stiffer but the yield strength of the honeycomb pattern was higher. Next, a bending test with fire was executed. At last, a 3-point bending test was performed with UV-aged panels. This showed stiffer behaviour and an increase in strength compared to the unaffected panels. It was argued that due to the UV light, the bonding material hardened, which created a better bond between the core and glass [65]. In the end, it is concluded that overall, the honeycomb pattern scores best on the criteria of weight, stiffness, printing time, transparency, complexity, UV resistance, and fire resistance.

Figure 12: The honeycomb pattern [65].

Figure 13: The space pattern [65].

The next thesis is written by Brugman, with the research question: "To what extent can a thin glass composite panel, with a polymetric 3D-printed trussed core, be improved to meet the thermal insulation and structural regulations of today, to be used as a facade element in the building industry?" [25]. The structural behaviour with in-plane loading is tested. Additionally, a method estimating the thermal insulation of a sandwich panel is provided as well as a heat test to validate the method. It concluded that the hand calculations are not completely in line with the physical tests. Furthermore, two panels are used, with material filling 26 % and 29 % of the core volume and U-values of 2.37 and 2.39, respectively. Thus, an increase in the core material does not have a profound effect on the thermal insulation. A facade panel with three glass layers and two cores is tested as well, which has a U-value of 1.90. This shows a significant increase in insulation compared to the other two panels. All panels are displayed in Figure 14. Additionally, the in-plane loading test of the facade elements showed that the failure depends on the connection between the panel and the core, as delamination occurred. However, the tests proved that the facade element could handle its own weight. The panels tested were 150 by 250 mm, where the thin glass panels had a thickness of 0.5 mm and the core had a thickness of 10 mm.

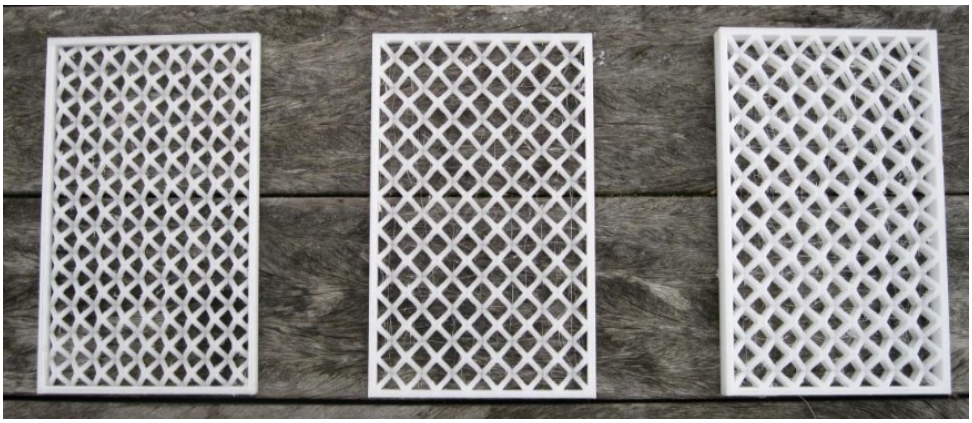


Figure 14: Tested facade elements by Brugman [25].

Finally, a paper published in 2023 by Pfarr and Louter [61], explores the use of digital techniques in the design and manufacture of a facade panel consisting of thin glass outer layers and an additively manufactured core. It states that the core design can be divided into four categories: aesthetic design, functional design, manufacturing design, and structural design. Each category consists of multiple objectives that influence the design. An overview of the categories and possible objectives and constraints is provided in Figure 15.

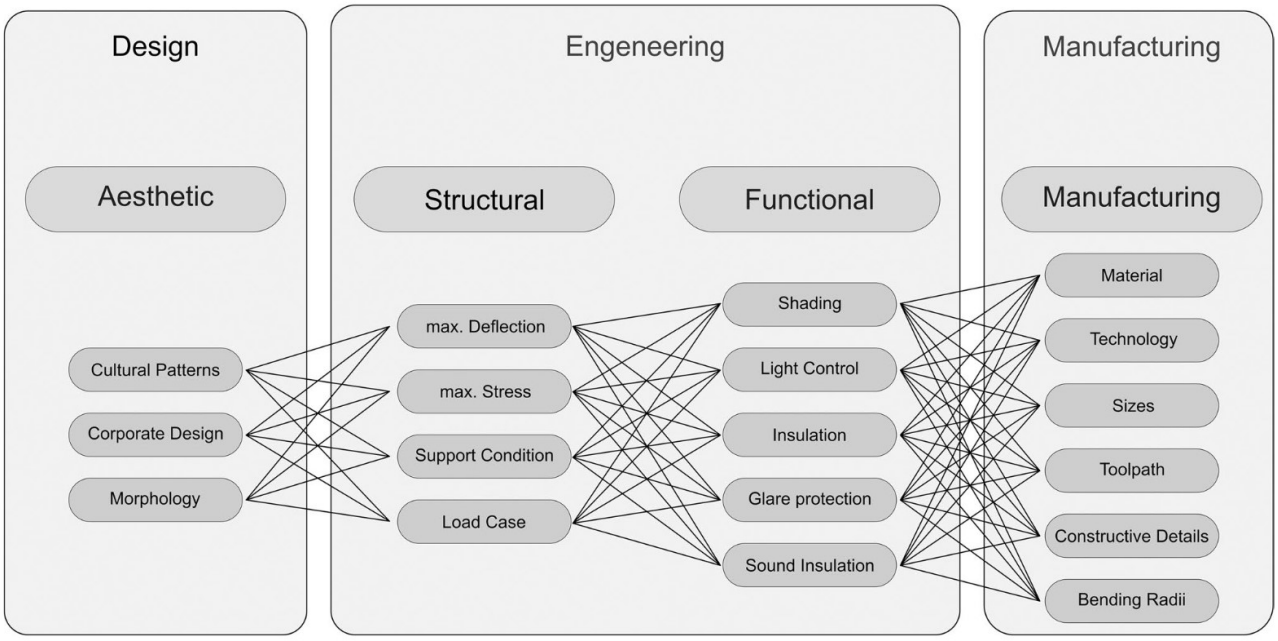


Figure 15: Overview of optimisation objectives and constraints divided into three disciplines [61].

The core design used is a honeycomb pattern, and post-recycled PETG, known as PIPG, is the core material. Additionally, a model is presented to change the design of the facade element in Grasshopper with the use of sliders, for example, by changing the facade geometries and the hexagon geometry. Subsequently, an FEA and a physical test were performed with a test specimen, which is visible in Figure 16. The test was performed by simulating a wind load on a panel with hinge supports along all four edges. For simplicity, the FEA was linear elastic and assumed a rigid and infinitely thin bonding layer. This caused the FEA to deviate from the physical test with larger loads, where non-linearities and the effect of the bonding layer became significant. The panel was loaded with 4 MPa and had a displacement of 2.18 mm and 1.82 mm in the physical test and FEA, respectively.

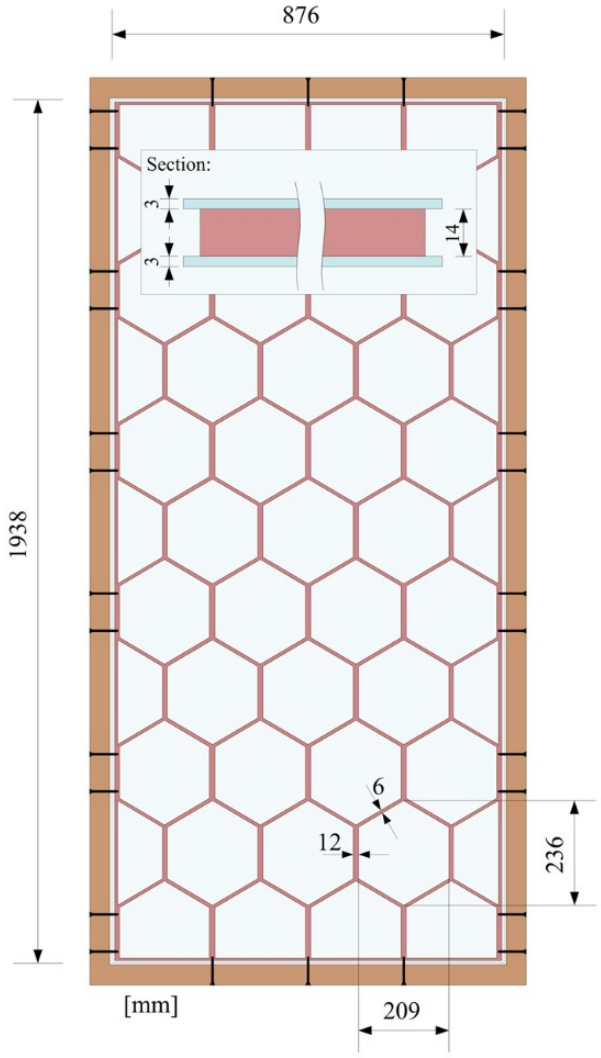


Figure 16: Design of the tested specimen Pfarr and Louter [61].

3 Structural requirements

In this chapter, the relevant loads on the facade element will be discussed. The only relevant load on the facade element is the wind load. The wind load determination changes per country and region; thus, it is chosen to only look at the Netherlands. Therefore, the NEN-EN codes will be used, which are the European calculation norms focused on the Netherlands. The latest version of the norms will be used, which is the NEN-EN 1991-1-1-4+A1+C2:2011/NB:2019 + C1:2020. First, the equation to calculate the wind force will be given, and then all the variables in the equation will be discussed.

3.1 Wind loads in the Netherlands

The wind force is an uniform, permanent load applied to the outside of the facade element and can be both pressure and tension. It is calculated with the following equation:

$$F_{wind} = c_s c_d \cdot c_{pe} \cdot q_p(z_e) \cdot A_{ref}. \tag{1}$$

Where:

- $c_s c_d$ is the structural factor. [-]
- c_{pe} is the pressure coefficient. [-]
- $q_p(z_e)$ is the peak velocity pressure at reference height z_e. $[N/m^2]$
- A_{ref} is the reference area, which is equal to the width multiplied by the height. $[m^2]$

3.1.1 Structural factor

The structural factor is defined as $c_s c_d$ and can be calculated with the detailed procedure as described in the NEN [56]. This gives the following equation:

$$c_s c_d = \frac{1 + 2 \cdot k_p \cdot l_v(z_e) \cdot \sqrt{B^2 + R^2}}{1 + 7 \cdot l_v(z_e)} \tag{2}$$

Where:

- z_e is the reference height, which is the average height of the construction element. $[m]$
- k_p is the extreme value as the ratio between the maximum value of the fluctuating part of the response and the standard deviation. [-]
- l_v is the turbulence intensity. [-]
- B^2 is the background response factor. [-]
- R^2 is the resonance response factor. [-]

However, for facade elements with a natural frequency larger than 5 Hz, the $c_s c_d$ is equal to 1. The natural frequency of glass facade elements with a width smaller than 3 m is normally above 5 Hz, according to the NEN codes. The natural frequency can be calculated with Appendix F in the NEN-EN 1991 1-4 2011.

$$n_{1,B} = \frac{K^2}{2 \cdot \pi \cdot L^2} \cdot \sqrt{\frac{EI_b}{m}} \tag{3}$$

Where:

- L is the length of the largest span of the element. $[m]$
- E is stiffness. $[N/m^2]$
- I_b is the moment of inertia. $[m^4]$
- m is the mass per unit of length in the center of the span. $[kg/m]$
- K is a value dependent on the span and support condition. [-]

In the case of the facade element, the K value is equal to π. To calculate the moment of inertia, the topology of the facade element must be known, but this is not the case a priori. Therefore, as a lower bound, the assumptions are that the material of the 3D printed structure does not influence the moment of inertia, and the centre of gravity is in the facade panel's center. This results in the following formula for the moment of inertia:

$$I_b = 2 \cdot \left(\frac{1}{12} \cdot b_g \cdot t_g^3 + (b_g \cdot t_g) \cdot (0.5t_L + 0.5t_g)^2 \right) \tag{4}$$

Where:

- b_g is the width of the facade element. $[m]$
- t_g is the thickness of the glass layer. $[m]$
- t_L is the thickness of the 3D printed layer. $[m]$

3.1.2 Pressure coefficient

The pressure coefficient depends on the size of the construction element, which is equal to $c_{pe,1}$ if the area is not larger than 1 m^2. Otherwise, Equation (5) is used for construction elements with an area between 1 and 10 m^2, where A is the area of the construction element in m^2. In the case of a facade element and the assumption that the building has a rectangular shape, the coefficient depends on the location of the facade element and the wind direction. The different areas are displayed in Figure 17. This area is used to determine the values of $c_{pe,1}$ and $c_{pe,10}$ in Table 5. If the ratio of h/d is between the stated values, a linear interpolation can be applied.

$$c_{pe} = c_{pe,1} - (c_{pe,1} - c_{pe,10}) \cdot log_{10}(A) \tag{5}$$

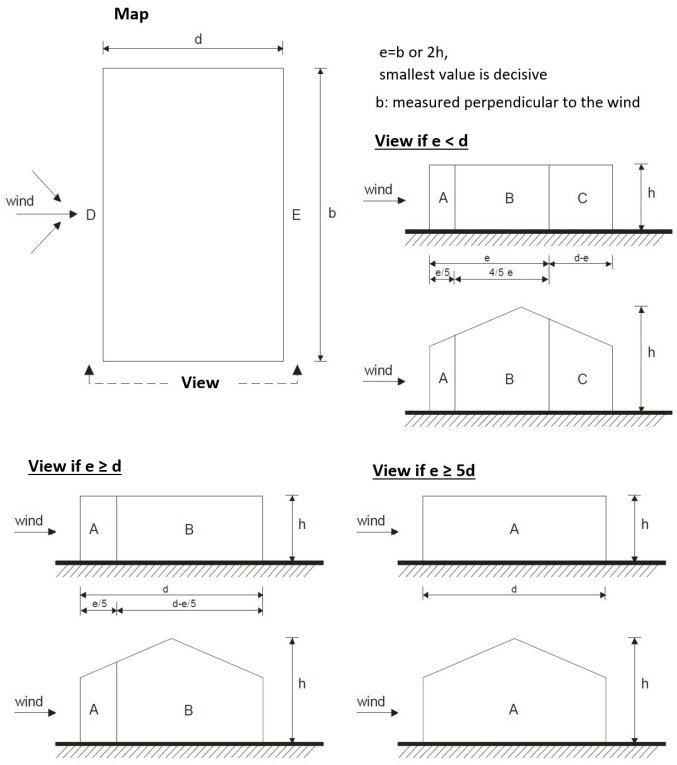


Figure 17: Areas related to c_{pe} value [56].

Zone	A	A	B	B	C	C	D	D	E	E
h/d	$c_{pe,10}$	$c_{pe,1}$	$c_{pe,10}$	$c_{pe,1}$	$c_{pe,10}$	$c_{pe,1}$	$c_{pe,10}$	$c_{pe,1}$	$c_{pe,10}$	$c_{pe,1}$
5	-1.2	-1.4	-0.8	-1.1	-0.5	-0.5	+0.8	+1.0	-0.7	-0.7
≤ 1	-1.2	-1.4	-0.8	-1.1	-0.5	-0.5	+0.8	+1.0	-0.5	-0.5

Table 5: Values for $c_{pe,1}$ and $c_{pe,10}$ for different areas and h/d ratios [56].

In cases where the ratio of h/d is greater than 5, Equation (6) can be used. The value $c_{f,0}$ depends on the ratio between the width and height of the element and is given in figure 7.23 in the NEN codes. The value of ψ_λ can be calculated with Table 7.16, Formula 7.28, and Figure 7.36 from the NEN codes. The ψ_λ depends on the volume fraction of the material inside the facade element, and the λ depends on the orientation of the construction and the length of the construction element.

$$c_{pe} = c_{f,0} \cdot \psi_\lambda \tag{6}$$

Where:

- $c_{f,0}$ is the force coefficient of a rectangular cross section without correction of the end effects. [-]
- ψ_λ is the end effect factor for elements where end effects are relevant. [-]

3.1.3 Peak velocity pressure

The peak velocity pressure depends on the wind area in the Netherlands, the surrounding terrain, and the height. All these aspects will be explained in the respective order. The Netherlands is divided into 3 areas based on the wind speed that occurs in a region, and the divisions are as follows:

- Area I: Markermeer, IJsselmeer, Waddenzee, Waddeneilanden and the province of Noord-Holland north of the municipalities Heemskerk, Uitgeest, Wormerland, Purmerend en Edam-Volendam
- Area II: the remaining part of the province of Noord-Holland and the main land of the provinces Groningen, Friesland and the provinces Flevoland, Zuid-Holland and Zeeland.
- Area III: the remaining part of the Netherlands

At the border of two different wind areas, a transition zone is introduced, which starts 5 km from the border and is fully located inside the area with the relative higher wind speed of the two neighbouring wind areas. In the transition zone, a linear transition between the two different wind speeds is created.

The other factor is related to the environment the construction is built in. The environment is divided into three categories: coastal, rural, and urban. The codes provide more detailed information on when to use certain environments and a more detailed calculation to incorporate the environment of the building and its effect on the wind load. However, as the tool is designed for architects, the simple method in the codes is used. This method only requires the area, terrain category, and height. This results in a peak velocity pressure, and the values are given in table NB.5 of NEN-EN, where for intermediate heights, a linear interpolation is used. The table is visible in Appendix A.

3.1.4 Reference Area

The reference area A_{ref} is simply equal to the width multiplied by the total thickness of the facade element because of the rectangular shape of the facade element.

3.2 Cavity pressure

The cavity pressure is another relevant load because the element consists of two glass layers with a layer of air between them. The air layer is a closed system, and changes in temperature and elevation result in isochoric pressures being exerted on the glass panels. The NEN 2608-2014 is used to determine the pressure [57]. First, the isochoric pressure as a result of temperature changes is discussed, and then as a result of changes in elevation. Finally, the pressure as a result of the external wind load is discussed.

3.2.1 Isochoric pressure due to temperature changes

The isochoric pressure caused by temperature changes is a variable load that is uniformly distributed over the glass area and is applied perpendicular to the glass surface. The temperature change is the difference in air temperature between the production process and the local temperatures that occur after placement. This load is only relevant when the width or height of the facade element is less than one metre [57]. The representative isochoric pressure, $P_{c,rep}$, is calculated via Equation (7).

$$P_{c,rep} = \phi \cdot C_c \cdot (T_s - T_p) \tag{7}$$

Where:

- $P_{c,rep}$ is the representative isochoric pressure. $[kN/m^2]$
- ϕ is the insulation factor of double glass, and the calculation steps are described in Appendix C. [-]
- C_c is the factor related to the pressure increase due to temperature differences and is 0.340. $[kN/m^2K]$
- T_s is the temperature of the gas between the glass layers after installation, visible in Table 6. $[°K]$
- T_p is the temperature of the gas during production visible in Table 6. $[°K]$

	Minimum	Maximum
T_s $[°C]$	-6.6	35.6
T_p $[°C]$	25.0	17.0

Table 6: Temperature ranges of T_s and T_p in facade elements with two glass layers [57].

3.2.2 Isochoric pressure due to height changes

Isochoric pressure can also be created as a result of height changes between production and installation and is a permanent, uniform load perpendicular to the glass surface. However, the pressure is only relevant when the height difference is greater than 150 m and the width or height of the facade element is less than one meter. [57]. The representative isochoric pressure is denoted as $P_{h,rep}$ in Equation (8). The isochoric pressure due to height and temperature is displayed in Figure 18.

$$P_{h,rep} = \phi \cdot C_h \cdot (h_{NAP} - h_{NAP,p}) \tag{8}$$

Where;

- $P_{h,rep}$ is the representative isochoric pressure. $[kN/m^2]$
- C_h is the factor concerning the increase of isochoric pressure caused by height differences, and is 0.012. $[kN/m^2/m]$
- h_{NAP} is the height relative to NAP, Amsterdam Ordnance Datum, after placement. $[m]$
- $h_{NAP,p}$ is the height relative to NAP during production. $[m]$

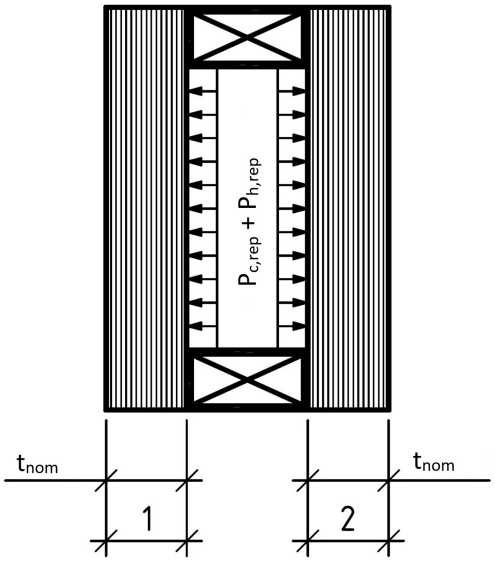


Figure 18: Isochoric pressure due to height and temperature [57].

3.2.3 Pressure due to external load

The last type of cavity pressure is the result of an external force applied to the glass, for example, the wind load. This force presses on the air cavity and results in an outward pressure, which is displayed in Figure 19. The respective uniform pressure, P_E, calculated with Equation (9), depends on the wind load, called F_{wind}.

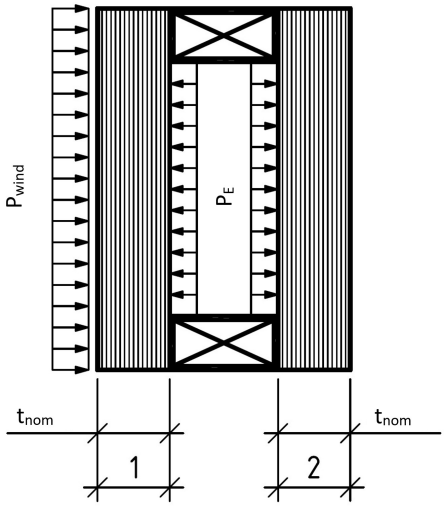


Figure 19: Cavity pressure as a result of an external force [57].

$$P_E = (1 - \phi) \cdot \frac{t^3_{blad,2,ser}}{t^3_{blad,1,ser} + t^3_{blad,2,ser}} \cdot F_{wind} \tag{9}$$

Where:

- $t_{blad,1,ser}$ is the equivalent glass thickness of layer 1, described in Appendix C. $[mm]$
- $t_{blad,2,ser}$ is the equivalent glass thickness of layer 2, described in Appendix C. $[mm]$

3.3 Load combinations

In the previous sub chapters, the different relevant loads are described. These loads are either permanent loads, the isochoric pressure due to height, or variable loads, the wind load, the isochoric pressure due to height, and cavity pressure due to an external load. The subsequent step is the combination of these loads in load combinations according to NEN-EN 1990: 2019, and determining the different load combinations in the ULS, ultimate limit state, and SLS, serviceability limit state [55]. First, the design lifetime is an important aspect in the calculation and is divided into five classes, where the lifetime ranges from 10 to 100 years. In the case of the facade element, which is used primarily in buildings and other ordinary structures, the class is equal to 4 or 5, which translates to a design lifetime of 50 to 100 years. The design life is related to the consequence classes. Structures are classified based on the impact that occurs in cases of structural failure, which results in the extra factor K_F to account for this. This factor is displayed in Table 7 with examples per consequence class.

Consequence class	K_F factor for loads	Examples
CC1	0.9	Greenhouses
CC2	1.0	Residential and office buildings
CC3	1.1	Concert hall

Table 7: Consequence classes according to NEN-EN 1990 [55].

Subsequently, the wind load and the cavity pressure due to an external load described in the previous sub chapters are called the characteristic loads in the NEN-EN codes, and they need to be altered to representative loads, which can be calculated with Equation (10). In contrast, the isochoric pressure due to temperature and height are already representative loads and do not require multiplication with the factor ψ.

$$F_{rep} = \psi \cdot F_k \tag{10}$$

Where:

- F_{rep} is the representative value of the load.
- F_k is the characteristic value of the load.
- ψ is the partial loadfactor reflecting possible unfavourable deviations from F_k. It equals to 1.00 or ψ_0, ψ_1, ψ_2, which are factors, depending on the rate of occurrence within a reference period, related to the combination value, frequent, and quasi-permanent value of variable load, respectively. [-]

The load combination formula with the permanent and variable loads in the case of the ULS is given in Equation (11) and (12). The least favourable answer of these formulas results in the load applied to the structure in the given load combination. The partial factors are given in Table 7.

$$\sum_{j\geq 1} \gamma_{G,j} \cdot G_{k,j} + \gamma_{Q,1} \cdot \psi_{0,1} \cdot Q_{k,1} + \sum_{i>1} \gamma_{Q,i} \cdot \psi_{0,i} \cdot Q_{k,i} \tag{11}$$

$$\sum_{j\geq 1} \gamma_{G,j} \cdot G_{k,j} + \gamma_{Q,1} \cdot Q_{k,1} + \sum_{i>1} \gamma_{Q,i} \cdot \psi_{0,i} \cdot Q_{k,i} \tag{12}$$

Where:

- $\gamma_{G,j}$ is the partial factor for permanent load j. [-]
- $G_{k,j}$ is the characteristic value of the permanent loads.
- $\gamma 0,1$ is the partial factor for the predominant variable load. [-]
- $\psi 0,1$ is the factor related to the combination value of the predominant variable load. [-]
- $Q_{k,1}$ is the characteristic value of the predominant variable load.
- $\gamma_{Q,i}$ is the partial factor for the variable load i. [-]
- $\psi_{Q,i}$ is the factor related to the combination value of a variable load i. [-]
- $Q_{k,i}$ is the characteristic value of the other variable loads.

CC	Equation	Permanent unfavourable	Permanent favourable	Variable unfavourable	Variable favourable
CC1	11	1.2	0.9	0	1.35
CC1	12	1.1	0.9	0	1.35
CC2	11	1.35	0.9	0	1.5
CC2	12	1.2	0.9	0	1.5
CC3	11	1.5	0.9	0	1.65
CC3	12	1.3	0.9	0	1.65

Table 8: Partial factors related to equation 11 and 12 with different Consequence Classes [55].

Furthermore, in the SLS, the load combination formula is given in Equation (13), which results in the load applied to the structure in a given load combination. According to NEN-EN 2608, the wind load has to be the predominant variable load [57]. Finally, the combination factors of the wind load and cavity pressure due to the wind load are displayed in Table 9.

$$\sum_{j\geq 1} G_{k,j} + \psi_{1,1} \cdot Q_{k,1} + \sum_{i>1} \psi_{2,i} \cdot Q_{k,i} \tag{13}$$

Combination factor	Value
$\psi_{0,i}$	0.6
$\psi_{1,1}$	0.9

Table 9: The combination factor of wind load and cavity pressure due to the wind load [55, 57].

3.4 Assessment structural requirements glass

The structural requirements of glass layers are the extreme limit state and the serviceability. The extreme limit state is the requirement that prevents structural failure, while serviceability is the requirement that ensures the element will fulfill its service, and the deformations are limited to prevent people from feeling unsafe.

3.4.1 Extreme limit state

The Dutch regulations state that the maximum stress in the glass panel must fulfill the following requirement:

$$\frac{\sigma_{pl,mt,i,d}}{f_{mt,u,d}} \leq 1.0 \tag{14}$$

Where:

- $\sigma_{pl,mt,i,d}$ is the calculation value of the bending tensile stress in glass layer i. $[N/mm^2]$
- $f_{mt,u,d}$ is the calculation value of the bending tensile strength of not prestressed glass. $[N/mm^2]$

The glass panels used in the facade element have a maximum stress that can be applied according to Dutch regulations stated in NEN2608-2014; see Equation (15). The first fraction on the right-hand side is related to the strength of the glass layer itself, and the second fraction is the strength due to prestressing the glass layer.

$$f_{mt,u,d} = \frac{k_a \cdot k_e \cdot k_{mod} \cdot k_{sp} \cdot f_{g,k}}{\gamma_{m,A}} + \frac{k_e \cdot k_z \cdot (f_{b,k} - k_{sp} \cdot f_{g,k})}{\gamma_{m,V}} \tag{15}$$

Where:

- k_a is a factor related to the surface effect. [-]
- k_e is a factor related to the edge quality of the element, equal to 1.0 with thermally tempered glass. [-]
- k_{mod} is a modification factor which depends on the load duration and reference period. [-]
- k_{sp} is the factor of the surface structure of the element, equal to 1.0 with thermally tempered glass. [-]
- k_z is the factor for the zone of the glass panel, related to prestressing the glass. [-]
- $f_{b,k}$ is the characteristic value of the bending tensile strength of the prestressed glass. $[N/mm^2]$
- $\gamma_{m,A}$ is the material factor of glass, which is 1.6 when the wind or isochoric pressure is the dominant variable load and else 1.8. [-]
- $\gamma_{m,V}$ is the material factor of prestressing, and equal to 1.2. [-]

In the facade element investigated in this research, the glass type is thermally tempered glass in cases where the thickness of the glass panel is at least two millimeters. Otherwise, the glass type is chemically tempered glass. The Dutch regulations, however, only include thermally tempered glass, and the values can be used from Table 10.

$k_{sp} \cdot f_{g,k}$	$f_{b,k}$	$f_{b,k} - k_{s,p} \cdot f_{g,k}$
45	120	75

Table 10: Characteristic values of bending tensile strength of prestressed glass in N/mm^2 [57].

The factor k_a, can be calculated with Equation (16), where A_{load} is equal to the area of the load in mm^2. This equation is applicable if uniform loads are present and a non-linear calculation is used to determine the bending tensile stress; otherwise, the factor is equal to 1.0.

$$k_a = 1.644 \cdot A_{load}^{-(\frac{1}{25})} \tag{16}$$

The factor k_{mod} is calculated using Equation (17), where t_{dur} is the duration in seconds of the shortest load in the used load combination and c_{cor} is the corrosion constant, which is temperature and humidity dependent and equal to 16 [57]. Table 11 displays the load duration.

$$k_{mod} = \left(\frac{5}{t_{dur}}\right)^{\frac{1}{c_{cor}}} \tag{17}$$

Load	**Load duration**
Wind	5 seconds
Isochoric pressure caused by temperature differences	12 hours
Isochoric pressure caused by height differences	permanent

Table 11: The load duration of relevant loads [57]

Finally, k_z depends on the location of the stresses on the glass panel as a consequence of the tempering process. The k_z is equal to 0.9 with stresses around the edge of the panel, 0 around the corner, and otherwise 1.0.

3.4.2 Serviceability

The assessment of the serviceability relates to the maximum displacement of the glass panel at the sides and in the middle.

The maximum displacement at the side of the panel is required to prevent leakage from the air layer between the glass panels, which would lower the insulation value significantly. This requirement is only applicable in cases where the facade element is supported along two edges instead of all four edges. It is presented in Equation (18), where E_{Length} is the length of the unsupported edge in mm, and u_{max} is the maximum displacement of the edge in mm.

$$u_{max} \leq \frac{E_{Length}}{200} \tag{18}$$

Furthermore, the deflection of the glass panels is limited in the middle, as shown by Equation (19), where L_{dia} is the largest diagonal of the glass layer in mm and $u_{dia,max}$ is the maximum deflection in the middle of the glass layer in mm [57].

$$u_{dia,max} \leq \frac{L_{dia}}{65} \leq 50 \tag{19}$$

4 Insulation

In order to answer the sub-question related to insulation, first a brief overview of the theory of insulation is provided, and followed by a calculation method that can be used in the optimisation tool.

4.1 Heat transfer

The insulation is equal to the resistance to heat transfer. Heat transfer can occur through three different modes: radiation, convection, and conduction. Thus, in order to know the insulation value, first the individual components have to be understood. A representation of the different modes and the effects of them on a facade panel is shown in Figure 20.

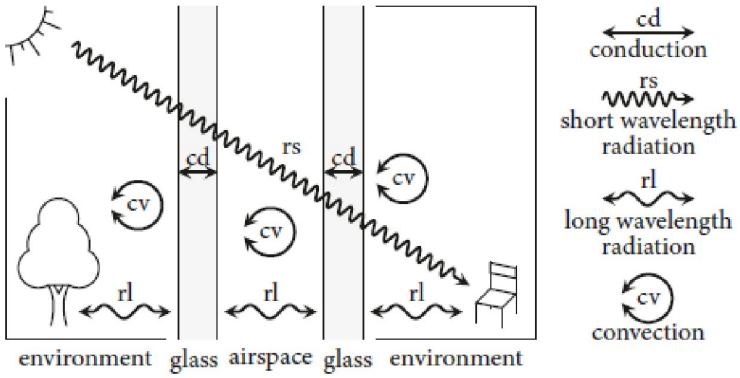


Figure 20: The modes of heat transfer through a double layered glass panel [64].

Radiation is the transfer of heat through electromagnetic waves, for example, the light of the sun. The radiation is a result of the temperature of the body, which means that every material with a temperature higher than absolute zero releases electromagnetic waves. An important characteristic of radiation is that heat transfer can go through a vacuum as well as matter. The radiation heat transfer can be calculated with the Stefan-Boltzmann Law, which gives the heat transfer from a body to its surroundings [25].

$$H_{rad} = \epsilon_{em} \sigma_{sb} T_{sur}^4 \tag{20}$$

Where;

- H_{rad} is the radiation heat transfer $[W/m^2]$
- ϵ_{em} is the emission constant of the surface [-]
- σ_{sb} is the Stefan-Boltzmann constant $[5.670 \cdot 10^8 W/m^2 K^4]$
- T_{sur} is the surface temperature [K].

Convection is the second mode, which is the transfer of heat caused by the movement of molecules in gases and liquids. This process occurs in the core of a facade panel. Moreover, convection requires the movement of molecules; therefore, this mode of heat transfer is normally limited to gases and liquids and not solids. The calculation of the heat transfer through convection can be done with Newton's law of cooling, which is shown below [40].

$$H_{conv} = h_{trans} \Delta T \tag{21}$$

Where:

- H_{conv} is the convection heat transfer $[W/m^2]$
- h_{trans} is the heat transfer coefficient $[W/m^2 K]$
- ΔT is the temperature difference [K].

Finally, conduction is the process of heat transfer within a body or between bodies in contact, which does not require a flow of molecules. It is an exchange of kinetic energy between molecules. The rate of conduction depends on the temperature difference and the resistance to heat transfer of a material, also known as thermal conductivity. This depends on the dimensions and physical properties of the material. This type of heat transfer occurs usually within or between solids. The heat transfer caused by conduction can be calculated with Fourier's law of thermal conduction, which is presented below [40].

$$H_{cond} = k_{mat} \nabla T \tag{22}$$

Where:

- H_{cond} is the conduction heat transfer vector $[W/m^2]$
- k_{mat} is the material conductivity $[W/mK]$
- ∇T is the temperature gradient [K/m].

4.2 Calculation method

The thesis of Brugman investigated the thermal resistance of a sandwich panel [25]. The resistance was calculated as well as measured through physical tests. This showed that the calculation methods all overestimated the U-value. Furthermore, multiple panels were tested, which showed the correlation of the density of the core to the U-value. This showed that the computation program Trisco, developed by Physibel, does not match the physical measurements. After comparing the increase in the U-value as the core density increases, the pessimistic calculation shows the best correlation compared to the measurements. The measurements showed an increase of 1.06 %, the pessimistic calculation 1.06, the optimistic 0.72 %, the thermal bridge effect 0.38 %, and Trisco 9.93 %. Thus, the pessimistic calculation method is used to calculate the U-value, which is shown below for a sandwich facade panel.

$$U = \frac{1}{\frac{A}{\frac{A_1}{R_1} + \frac{A_2}{R_2}} + R_{outside1} + R_{outside2} + \alpha_{trans}} \tag{23}$$

$$\alpha_{trans} = \frac{1}{\alpha_{inside}} + \frac{1}{\alpha_{outside}} \tag{24}$$

$$R = \frac{t}{\lambda_{ther}} \tag{25}$$

Where:

- A is total surface area of the panel $[m^2]$
- A_1 is the total contact area of core material $[m^2]$
- A_2 is the total surface area of the panel minus the contact area of the core, thus $A2 = A - A1$ $[m^2]$
- R_1 is the heat resistance of the solid core material $[m^2K/W]$
- R_2 is the heat resistance of the gas core $[m^2K/W]$
- $R_{outside1}$ is the heat resistance of the first outside layer $[m^2K/W]$
- $R_{outside2}$ is the heat resistance of the other outside layer $[m^2K/W]$
- α_{trans} is the inverse heat transmission coefficient inside and outside $[m^2K/W]$
- α_{inside} is the heat transmission coefficient inside layer $[W/m^2K]$
- $\alpha_{outside}$ is the heat transmission coefficient outside layer $[W/m^2K]$
- λ_{ther} is the thermal conductivity of the material $[W/m^2K]$
- t is the thickness $[m]$.

5 The role of natural light in buildings

The role of natural light in buildings and its connection with the necessity of windows have undergone a major change since the mass usage of electricity. Prior to electricity, daylight was essential for the indoor climate as the main source of light. This led to multiple developments and trends in architecture to allow for sufficient daylight to reach the buildings interior, for example, skylights, bay and bow windows, and Palladian windows. Through the invention of more primitive light sources such as candles, oil lamps, and gas lamps, it introduced other ways of providing well-lit rooms, although they were not really substituting the need for windows.

As electrical light sources and air conditioning became more prominent, a strong belief emerged that these inventions could create an indoor environment, especially in the office, which is superior to daylight and sunlight [62]. This gave rise to building designs that depended chiefly on electrical-based light sources to light the interior. The discovered disadvantages were a reduction in working efficiency, eye strain, and headaches, which were most prominent during the 1980s.

The reassessment of the role of natural light on the indoor climate has been an important consideration during the last two decades. This change has been mainly caused by the awareness that daylight and the ability to see the surrounding environment are major factors for human well-being and the intent to lower the overall energy consumption due to climate change.

5.1 Effect of day light on the indoor climate

This paragraph intends to briefly cover the effect of natural light on humans. First, the positive effect of natural light on humans in the interior has been researched since the end of the twentieth century.

One of the benefits of daylight is that it consists of a full spectrum of light, which is not the case for unnatural light sources. The day light is important to human health through numerous biological effects, for example, affecting the circadian rhythm [32]. Additionally, according to Terman [72], day light could positively effect energy loss and work disturbance.

Furthermore, an increase in productivity in the office is a well-known effect of day light, as studies have shown [32]. The increase in productivity is further enhanced by a reduction in absenteeism. Subsequently, according to Boyce [24], an increase in alertness and attention occurs during the after-lunch dip as a result of natural light.

Beside the natural light coming into the building, it creates the opportunity for the building occupants to see the outside environment, which is an important physiological aspect [32]. Finally, it has been shown since the 1970s that office workers highly value windows and affect job satisfaction [29, 30, 36, 79].

5.2 Effect of direct sun light on the indoor climate

While daylight has many advantages and is considered a necessity for a comfortable indoor climate, direct sunlight has two important disadvantages. The first disadvantage is related to the solar heating, which would result in discomfort for the occupants or a higher energy cost to maintain a certain indoor temperature. Second, the glare is an unwanted result of direct sunlight. This could overstimulate the human eye, degrade light-sensitive objects, and make computer screens unreadable [45].

5.3 Conclusion

As discussed above, daylight has numerous advantages and is essential to providing a healthy and comfortable indoor climate. However, direct sunlight increases the indoor temperature and produces glare. Thus, in the design of the facade panel, the panel should allow for sufficient day light and ideally limit the direct solar radiation. Furthermore, in a location in the Netherlands, the winters have limited sun hours, shorter days, and temperatures below the average preferred indoor temperature. This implies that the disadvantages of direct sunlight during the winter are not as disadvantageous, and mainly during the summer, the direct light should be limited.

6 Topology optimisation theory

Structural optimisation exists in three categories: size, shape, and topology. The size optimisation, also known as topometry optimisation, is a global and local size optimisation, where each element can be optimised independently [44]. The shape optimisation category is divided into non-parametric, parametric, and topographic optimisation. The final category, topology optimisation, is where the distribution of material is spatially optimised by maximising or minimising one or more objective functions [73]. Thus, topometry optimisation optimises the initial geometry with respect to, for example, its height, length, width, and thickness; topographic optimisation optimises the shape of thin structures through the existence or absence of beads; and topology optimisation can optimise a vast array of structures through discretising the geometry in small areas containing material or being void. Subsequently, the support and load conditions are defined, and additional design restrictions can be added in the form of prescribed solid areas and holes [19]. In this case, only isotropic material is used, and anisotropic material is not discussed. This chapter briefly explains the theory of topology optimisation.

6.1 Principles of topology optimisation

The topology optimization is expressed mathematically by a design variable, the material distribution, a relationship between loads and displacements linked through kinematic and constitutive relationships and equilibrium, and the objective function or functions. The objective functions can be, for example, minimum compliance, minimum weight, and minimum displacement. Compliance is equal to the inverse of the stiffness of the system, and minimising the compliance is equal to maximising the stiffness of the system. Subsequently, the design space, Ω, is discretised, and an equilibrium constraint is introduced, which in linear problems is equal to:

$$\mathbf{K} \cdot \mathbf{U} = \mathbf{F}. \tag{26}$$

The $\mathbf{K}$ is called the stiffness matrix, $\mathbf{U}$ the displacement vector, and $\mathbf{F}$ the force vector. A traditional compliance topology optimisation is discussed by Bendsøe and Sigmund [19], in one of the earlier papers in this domain. The objective function is to minimise the compliance, which is the same as maximising the global stiffness. In general, finite elements are used to discretise the problem. As mentioned above, two fields are relevant: the stiffness, which will be called E, and the displacement, named u. In the case one finite element mesh is used in both fields and in each element the stiffness is discretized as a constant, the discrete form is:

$$\begin{aligned} &min(c) \\ &c = \mathbf{U}^T \mathbf{K} \mathbf{U} = \sum_{e=1}^{N} E_e \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e \,. \end{aligned} \tag{27}$$

In Equation ((27)), the $\mathbf{F}$ and $\mathbf{U}$ are the global load and displacement vectors, $\mathbf{K}$ is the global stiffness matrix, and element e, with $e = 1, ..., N$, where N is the total number of elements [19]. The $\mathbf{u}_e$, and $\mathbf{k}_0$ are the element displacement vector and element stiffness matrix for an element with unit Young's modulus. Subsequently, the E_e is the stiffness matrix of element e and the x_e is the location of element e in the domain Ω [68]. The stiffness of each element can be either equal to the stiffness of the material or equal to zero if it is void. A mathematical representation is given in Equation (28), where E^0 is the material stiffness tensor, Ω^{mat} is the material domain, and V_{mat} is the amount of material, which can be constraint[19].

$$\begin{aligned} &E_e = a_\Omega \, E^0, \; a_\Omega = \begin{cases} 1 \,\text{if}\, \gamma(x_e) \in \Omega^{mat} \\ 0 \,\text{if}\, \gamma(x_e) \in \Omega \backslash \Omega^{mat} \end{cases} \\ &\int_\Omega a_\Omega \, d\Omega = V_{mat}(\Omega^{mat}) \end{aligned} \tag{28}$$

The discrete formulation in Equation (28) has multiple problems, such as the checkerboard problem displayed in Figure 21, mesh-dependent solutions, and computational limitations [73]. However, the computation limitations can be overcome by replacing the variable a_Ω by a continuous variable. The checkerboard problem and the mesh-dependent solutions can be solved with another method, which is discussed in the next sub-chapter. Although this formulation solves the problems of the discrete formulation, an extra problem is created. In the discrete formulation, the element can be either material or void, while in the continuous formulation, the element can be a combination of material and void; for example, a_Ω can be equal to 0.5. In order to limit the problem, a penalty can be included to direct the solution to use mostly discrete values, namely zero and one [19].

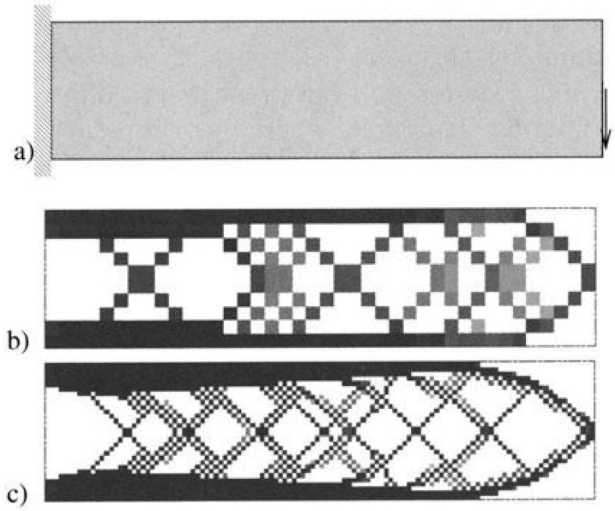


Figure 21: Illustration of a checkerboard pattern appearing after topology optimisation [19]

One of the commonly used method and proven efficient, is the penalized, proportional stiffness model known as SIMP, which is an acronym for "Solid Isotropic Material with Penalisation" [19]. This method uses a continuous formulation and is used in 83 % of the software applications [73]. The continuous equation used for the stiffness matrix is displayed in Equation (29), where p is the penalty factor steering the solution to a discretised form and $\rho(x_e)$ is a continuous variable giving the density of element e and replacing the discretised variable a_Ω.

$$\begin{aligned} &E_e = \rho(x_e)^p E^0 \text{ with } p > 1, \\ &\int_\Omega \rho(x_e) d\Omega \leq V_{mat}, \\ &0 \leq \rho_{min} \leq \rho(x_e) \leq 1,\ x_e \in \Omega \end{aligned} \tag{29}$$

In practice, a minimum value, for example, ρ_{min} is 10^{-3}, of $\rho(x_e)$ is introduced to prevent singularities in the equilibrium problem [19]. Additionally, p is used to penalise the intermediate values, especially for larger values of p. As discussed in the book "Topology Optimization" by Bendsoe, M.P., and Sigmund, O., in problems with an active volume constraint, experience showed that using a sufficiently large p, usually $p \geq 3$, is necessary to obtain a design with a few intermediate $\rho(x_e)$ values [19].

Another method to prevent singularities is the modified SIMP-method. This modification is shown in Equation (30), where E_{min} is the modulus of elasticity of void material and non-zero to prevent singularities, and $\beta(x_e)$ is equal to $\rho(x_e)$. This variable can be zero without creating singularities as a result of the modified equation [69]. This method has the advantage of independence between the penalty value and the minimum value of the material's modulus of elasticity.

$$\begin{aligned} &E_e = E_{min} + \beta(x_e)^p \cdot \left(E^0 - E_{min}\right) \\ &0 \leq \beta(x_e) \leq 1,\ x_e \in \Omega \end{aligned} \tag{30}$$

Other popular methods are level set and evolutionary structural optimization, ESO [73]. The methods are only briefly explained because the SIMP-method is used in the vast majority of software.

The ESO method has been developed to solve various structural optimisation problems, for example, stress considerations, stiffness constraints, and frequency optimisation [80]. The method uses a fixed domain with finite elements representing the initial design, and the optimal design is a subset of this fixed domain consisting of finite elements. The optimal design is found by assessing the individual contribution of each element to the behaviour of the total structure under the applied boundary conditions and subsequently removing the elements with the least contribution in multiple iterations. Thus, the method is focused on the local elements instead of the global optimum and is computationally expensive [77]. A similar method is the BESO method, which is the bi-directional evolutionary structural optimisation method. This method can add as well as remove material during one iteration, which is a widely used method and has proven to be an efficient topology optimisation algorithm [39, 46].

The level set method defines the interface of material and void implicitly with iso-contours, which represent the boundary between material and void, of a level set function [75]. A level set function gives the distance to the boundary, the gradient of the boundary, and the curvature [42]. The advantage of this method is the clear description of the boundaries and the structural behaviour around the boundaries.

6.1.1 Objective functions

The SIMP-method and the modified SIMP-method have been explained in previous paragraphs, and the compliance objective is described in Equation (27). The compliance objective is a minimisation, which means that the optimal value can be found with the derivative of the compliance. The SIMP-method compliance derivative is given in equation 31, and the modified SIMP-method compliance derivative is given in Equation (32).

$$\frac{\delta c}{\delta \rho(x_e)} = -p\rho(x_e)^{p-1}(E_0)\mathbf{u}_e^T\mathbf{k}_0\mathbf{u}_e \tag{31}$$

$$\frac{\delta c}{\delta \beta(x_e)} = -p\beta(x_e)^{p-1}(E_0 - E_{min})\mathbf{u}_e^T\mathbf{k}_0\mathbf{u}_e \tag{32}$$

6.2 Drawbacks topology optimisation

As mentioned before, one of the problems is the dependency of results on mesh size. This relates to the phenomenon that, generally, more holes and using the same structural volume will create a more efficient structure. This increase in finite elements results in more holes in the structure. This can be a problem for additive manufacturing. Another issue is the creation of checkerboard patterns in the results, caused by the discretisation of a continuous structural problem.

First, for mesh-dependent solutions, the optimal design for additive manufacturing should be a design that is printable and has a clear location of the boundaries, which requires a high number of finite elements. Thus, the ideal goal of mesh refinement is a clearer location of the boundaries and better finite element modelling without receiving a different, more detailed structure. To address this problem, methods have been discovered to prevent it in an efficient and straightforward way. The goal of these methods is to prevent the appearance of fine-scale structures. Techniques to achieve this include reducing the parameter space directly for the design, introducing optimisation implementation filters, and adding optimisation problem constraints [19].

One of the methods described here is density filtering, which is widely used in the available software. This method is chosen as it is a local constraint, which in general prevents thin bars from occurring, while some other methods are global constraints and do not prevent the formation of them[19]. The SIMP-model is modified by this local constraint as follows:

$$\begin{aligned} \tilde{\rho}(x_e) &= \left(\sum_{i \in N_e} H_{ei}\right)^{-1} \sum_{i \in N_e} H_{ei}\, \rho(x_i) \\ H_{ei} &= \max\left(0, r_{min} - \Delta(e,i)\right). \end{aligned} \tag{33}$$

In Equation (33), N_e is a set of elements i that have a smaller center-to-center distance, $\Delta(e,i)$, to element e than the specified filter radius, r_{min} and $\tilde{\rho}(x_e)$ is the filtered density of element e [12].

As discussed earlier, the checkerboard is related to the discretisation and an overestimation of the stiffness of checkerboards [19]. The methods mentioned in the mesh refinement paragraphs already reduce or remove the possibility of checkerboards appearing because they enforce a geometry constraint. Only in cases where the geometry must not be constrained are other methods used to prevent checkerboards from occurring. However, in this research, the geometry needs to be constrained to be able to print the structure. Therefore, no methods are discussed that are specifically designed to eliminate checkerboards.

6.3 Topology optimisation procedure

The topology optimisation procedure can be divided into three main parts: pre-processing, optimisation, and post-processing.

In the pre-processing step, a reference domain is specified that contains all the material or possible material of the structure. The reference domain can be divided into a part that is used for the topology optimisation, a part with specified areas where material is present, and void areas. Additionally, the boundary conditions

are described, which specify the support conditions and the external forces on the structure. Also, the material properties have to be assigned, and the finite element mesh has to be created.

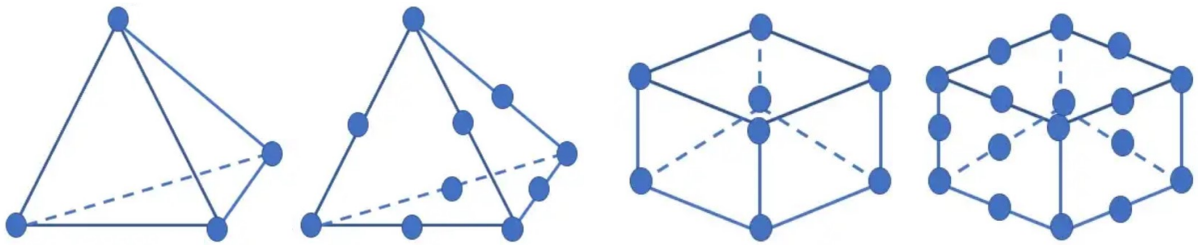
Secondly, in the optimisation step, the criteria and method need to be chosen, for example, the number of iterations, stop criteria, volume constraint, and algorithm, such as the SIMP-method. Subsequently, the optimisation itself will iterate and compute the stresses and strains as well as the material distribution.

Finally, the results are post-processed. The optimal design can be presented as CAD and prepared for a 3D printer. An important step in the post-processing with the use of continuous variables is the determination of material and void elements because intermediate values are present. For example, an element with $\rho(x)$ equal to 0.5 is material or void in the final design. Thereby, a finite element analysis can be used to determine if the stresses and displacement remain within safe values according to the regulations.

6.4 Finite Elements

Another important aspect of topology optimisation is the choice between different finite elements, as these have different advantages and disadvantages relative to each other. The topology optimisation is in 3D; therefore, only solid finite elements are discussed. Additionally, not a complete overview of finite elements is provided; only finite element types that can be used in the topology optimisation software are discussed.

First, two well-known finite element types are the thetrahedral and the hexahedral, visible in Figure 22a, and 22b, where the dots represent the position of the nodes. Both the linear and quadratic elements are shown per type, and the linear element only has nodes at the corners, while the quadratic element also has nodes at the midsection of the edges.



(a) Linear and quadratic tetrahedral

(b) Linear and quadratic hexahedral

Figure 22: Illustration well-known finite elements

The choice of finite elements depends on three main factors, which are computation costs, engineering costs, and engineering judgement. The computation costs relate to the number of elements used and the choice between linear or quadratic elements. As the number of elements increases, the number of equations to calculate the strains and stresses increases. Subsequently, the use of quadratic elements increases the number of nodes significantly, which results in additional computation time. The exact relationship between element size and linear or quadratic elements and the computation time depends on the software used. The relationship is shown more in depth in the chapter related to topology optimisation software.

Furthermore, the engineering costs are related to the required pre-processing and interpretation of the results. The pre-processing includes the creation of the mesh, which is in general easier with a tetrahedral element compared to a hexahedral. The interpretation of results relates to the control of strain and stress, and whether the results are aligned with expectations.

Finally, the engineering judgement determines the appropriate finite element for the model. In this case, four element types can be used. The benefit of tetrahedral elements is that, compared to hexahedral elements, meshing is easier. However, tetrahedral elements are more prone to shear locking, especially linear elements. Shear locking is an additional shear stress in the model compared to the exact solution. Therefore, the element appears stiffer than it actually is, which reduces the bending deformations and increases the stress. To prevent shear locking in the model, the element size should be decreased to use more elements along the thickness, or quadratic elements can be used. The second consideration is the use of linear or quadratic elements. Linear elements use linear shape functions, which result in the strain varying linearly. Thus, the linear elements do not capture the bending behaviour correctly, and a higher number of elements is required to create a good enough discretisation to represent the nonlinear behaviour. The quadratic elements have nodes in the middle of the edges and therefore use second-order shape functions. These functions are better able to represent bending deformation and complex geometries [78].

7 Topology optimisation application

Multiple methods have been invented in the last few decades to use topology optimisation. These methods can be placed into two categories: topology optimisation integrated into software and topology optimisation codes written in, for example, Python or Matlab. First, a short review of educational papers is provided. Then, some of the topology optimisation written in code is discussed, which covers one of the earliest methods to use topology optimisation. Subsequently, the topology optimisation software is discussed. Additionally, the tools are compared, where computation time is an important criterion. Finally, the SIMP-model or modified SIMP-model is used as the optimization procedure in all the discussed methods.

7.1 Review educational articles

One of the first educational papers on topology optimization is a 99-line Matlab code that uses the SIMP method [68]. This code has been optimised, and 3D functionality has been added in the following two decades. These codes will also be discussed and compared more thoroughly [12, 35]. Moreover, a 129-line Matlab script based on the level-set method has been created, which minimises the compliance of a statically loaded structure [27]. Furthermore, another Matlab code, published in 2010, solves a 2D stiffness optimisation problem that uses the BESO, bi-directional evolutionary structural optimization, method. This used a soft-kill scheme, which means that instead of using a discrete method with one and zero, the zero is replaced by a very small number [38, 39]. Also presented in 2010 is a Matlab code with 199 lines that uses pareto-optimal tracing [71]. Additionally, an often-cited Matlab code called 'Top3d', presented in 2014, includes a 169-line code to solve minimum compliance problems [47]. More recently, a Matlab code of 143 lines was presented, which implements the SERA, sequential element rejection and admission, method and can consider numerous load cases [13].

Through the increase in popularity of topology optimisation, professional software has added capabilities for topology optimisation, such as ANSYS, ABAQUS, Solidworks, and Rhino. These programs provide powerful meshing, modelling, analysing, and post-processing functionalities. Thereby, they have the option to use scripting inside the GUI, or general user interface, of the software itself. For example, ANSYS uses APDL, Ansys Parametric Design Language, and recently PyAnsys was developed, which adds the functionality to use the APDL capabilities in Python. ABAQUS can use Python scripts, and it created a Python library, 'abaqus'. At last, Rhino can use Python; however, the current version of Rhino, Rhino 7, uses Iron Python, which misses the recent libraries. This inspired the topology optimisation 100-line Python code of Zuo and Xie (2015). This code uses the ABAQUS scripting interface, which provides access to the FEA, and combines this with a code performing the optimisation process [81]. Subsequently, a 390-line code written in ANSYS APDL performs a 3D topology optimisation using the BESO method with a dynamic evolution rate to improve the convergence rate [46].

7.2 Matlab or Python codes

One way of incorporating the topology optimisation is through the use of written code in, for example, Python or Matlab. Fortunately, there are codes made available by the department of mechanical engineering and solid mechanics at the Technical University of Denmark, and particularly through the efforts of Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan S. Lazarov, and Ole Sigmund. One of the first numerical codes for topology optimisation was a 99-line script by Sigmund using the SIMP-model, and the codes mentioned below are a version of this code that uses the modified SIMP-model [68]. These available codes, which can be used for educational purposes, are a Python code for 2D topology optimisation, and a Matlab code for 2D and 3D topology optimisation. The 2D codes are explained in the article "Efficient topology optimisation in MATLAB using 88 lines of code" by E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund [12]. Thereby, the 3D codes are discussed in another article from 2020 [35].

The flexibility and possibility of changing the code for a specific intention are significant advantages of using a code. The codes allow the user to specify, for example, the finite element type, constitutive relations, and boundary conditions. As computational time is an important criteria, the Python code and Matlab code with 2D topology optimisation are compared, and the tables below display the relationship between the computational time and the number of elements. The structural problem that is used is a cantilever beam with a point load at the end, visible in Figure 21 in the previous chapter. The 2D topology optimisation uses rectangle elements, a node on each corner, and two degrees of freedom per node. Moreover, the 3D topology optimisation has also been run, and the results are displayed in the table below. The 3D topology uses a box element, a node on each corner, and 3 degrees of freedom per node. The last table shows clearly that the 3D topology optimisation in Matlab is very slow, and computation time increases exponentially with more elements. The computer used to

perform these calculations is the 'Nitro 5 Gaming Laptop' from Acer, which has an i7-12700H processor with 14 cores, a RTX 3060 graphics card, and 16 GB of RAM.

Number of elements	Number of nodes	Number of DOFs	Computation time (s)
90,000	90,601	181,202	1
250,000	251,001	502,002	3
490,000	491,401	982,802	6.2
1,000,000	1,002,001	2,004,002	17

Table 12: The computation time per iteration of different numbers of elements. The optimisation problem is equal to the one discussed in the paper and is a 2D topology optimisation with Python code.

Number of elements	Number of nodes	Number of DOFs	Computation time (s)
90,000	90,601	181,202	0.7
250,000	251,001	502,002	2.2
490,000	491,401	982,802	4.6
1,000,000	1,002,001	2,004,002	10.5

Table 13: The computation time per iteration of different numbers of elements. The optimisation problem is equal to the one discussed in the paper and is a 2D topology optimisation with the 88 lines Matlab code.

Number of elements	Number of nodes	Number of DOFs	Computation time (s)
27,648	88,474	265,422	6
54,000	172,800	518,400	80

Table 14: The computation time per iteration of different numbers of elements. The optimisation problem is equal to the one discussed in the paper and is a 3D topology optimisation with the 99 lines Matlab code [35].

7.3 Software integration

The other method to use topology optimisation is with software that integrates topology optimisation. In recent years, the popularity of topology optimisation has increased, which has resulted in an expanding number of software platforms supporting topology optimisation.

In January 2022, an interesting paper was published, which performed a comparative study of the commercial software platforms with topology optimisation and created a list of available commercial software platforms [73]. This paper stated that 31% of the available software platforms are open source and the other 69% are commercial, and the SIMP-model is available in 83 % of the platforms, while the ESO and Level set are only available in 17 % and 20 % of the software platforms, respectively.

Additionally, an important constraint in the choice between the software platforms is whether it has a student edition or if the university made it possible to use the software platforms. The paper addressed above compared ABAQUS, Solidworks, and ANSYS, where only ANSYS is available for the students at the University of Delft. Another available software platform is Rhinoceros, which is a 3D tool to create and analyse different shapes and has a topology optimisation package. Therefore, these two software platforms are used and compared in the subsequent paragraphs.

7.3.1 Rhinoceros

Rhinoceros is a versatile CAD program that contains a useful plug-in called Grasshopper, which is a parametric design environment with numerous capabilities and options to use plug-ins created by other users. One of those plug-ins is Topos [20]. This is a 3D topology optimisation tool based on the SIMP method and the same article as discussed in the previous paragraph. A large advantage, in contrast to the codes, is the integration within a CAD program and the capability of using different plug-ins within Grasshopper, such as Karamba3D, that can perform structural analyses of beams, plates, and shells.

The topology optimisation plug-in provides the possibility of specifying an area in which the optimal material distribution needs to be found. Furthermore, the load conditions can be specified as a point, line, or area load. However, only one load can be applied to the structure. Moreover, with the specified area, areas can be marked that must contain material or be void, which creates the possibility in the case of the facade element

to mark the areas where the glass panel is situated and specify that it must contain material. Additionally, only one material type can be used, with a specified Young's modulus and Poisson ratio. The software uses 3D solid cubical finite elements with a node on each corner and three degrees of freedom per node. Unfortunately, it is not specified which degrees of freedom are involved. Next, a volume fraction needs to be specified, and the software tries to find the material distribution with the highest global stiffness. Thus, the only objective function available is compliance. Subsequently, the density filter, discussed in the previous chapter, can be applied to create a printable geometry.

The computation time is an important aspect of choosing the right tool. Therefore, the table below provides an overview of the computation time with different numbers of elements. The design problem is similar to the design problem described in the thesis. The geometry is a square plate of 500 millimeters and 20 millimeters thick. An area load is applied perpendicular to the plate, and all four edges have fixed supports. This computation time is related solely to the optimisation process itself. Additionally, displaying the stresses and density of the elements adds to the computation time, but this increases the computation time barely depending on the number of elements. However, the creation of a mesh takes significant time and can range from 1 minute to 20 minutes, with a ranging number of elements between 64,000 and 1,000,000. The computation time with different numbers of elements is shown in Table 15.

Number of elements	Number of nodes	Number of DOFs	Computation time (s)
64,000	206,763	620,289	0.23
216,000	680,943	2,042,829	0.75
512,000	1,594,323	4,782,969	2.21
1,000,000	3,090,903	9,272,709	4.84

Table 15: The computation time per iteration with different numbers of elements.

Beside the geometry as an output, the software has the possibility to show the deflections and stresses, which is an important output as the optimised geometry does not change as the magnitude of the load changes. However, the units are not provided. Additionally, in Rhinoceros, the length unit influences the model, although it is not specified which unit should be used. For example, if the unit goes from mm to m, the deflection values as well as the stress values are increased by 1000, while one could argue that if the unit goes from mm to m, the magnitude of the value shrinks by a factor of 1000 and the deflection is reduced by 1000. Finally, the stress values are mesh-dependent, while they should be mesh-independent. Therefore, the stresses and deflections are not trusted.

7.3.2 ANSYS

Another software platform is ANSYS, which is a high-quality, commercial simulation tool. A sub-module is ANSYS Mechanical, which can be used for a variety of analyses, such as structural analysis and topology optimisation. In this thesis, ANSYS 2022 R2 is used.

The topology optimisation software can be based on density or level set, which were both discussed in the previous paragraph. ANSYS can also use other methods, such as mixable density, shape optimisation, and lattice optimisation, but these are not used in this thesis and therefore not discussed. Subsequently, the objective needs to be defined, which can be a minimisation of compliance, mass, volume, and stress, for example, the equivalent von-Mises stress. Beside using one objective, the software is capable of using different objectives and giving a user-defined weight to each objective. The next step is to define the constraints, which can be: volume, mass, center of gravity, moment of inertia, compliance, displacement, reaction force, global stress, and local Von-Mises stress.

Furthermore, computational time is an important aspect, and multiple simulations have been performed. The simulations used a compliance minimisation objective and a volume constraint. The used simulation is the same as the simulations in the subsequent paragraph. It used 648,000 linear cubical elements and took between 30 and 50 iterations, depending on the volume constraint. The average computation time per iteration is 1.7 minutes. The number of iterations as well as the computation time per iteration increase as a lower volume constraint is used. An additional test is performed with 85.500 elements to get an indication of the computation time per iteration. The computation time per iteration is 8 times smaller; thus, an approximation is that as the number of elements doubles, the computation time per iteration doubles as well. Thus, if 1,000,000 elements are used, it will take approximately 2.6 minutes.

7.4 Code combined with software application

As discussed in an earlier paragraph, in recent years, papers have appeared that use both a programming environment such as Python or Matlab and Finite Element Analysis, FEA, software. This has the advantage that the software packages have included topology optimisation; however, the capabilities are limited and based on usability for a large group of users. If a specialised topology optimisation has to be performed with specific objectives and constraints, a topology optimisation software such as ANSYS or ABAQUS is probably insufficient. Although the limited capabilities of the software, the Graphical User Interface, GUI, and FEA are mostly well established and optimised. Also, FEA software makes it possible to use their analysis from Python and Matlab, for example, the PyAnsys and ABAQUS scripting interfaces. Therefore, it is beneficial to still use these.

This results in the combination of Python or Matlab code, which defines the objectives, constraints, geometry, and mesh generation, and FEA of the software. For example, a paper released in 2015 combines ABAQUS and Python code to create a compliance minimisation optimisation with a volume constraint. The optimisation algorithm is the BESO method, and the whole code is only 100 lines long [81].

Additionally, a paper published in 2020 presents a topology optimisation written in ANSYS APDL code. Its objective is compliance minimisation and allows for multiple constraints, for example, volume, maximum displacement, and natural frequency. The Lagrangian method is used to have the possibility of adding multiple constraints. The code is in total 390 lines, and the optimisation algorithm is a dynamic rate BESO method [46]. This code can only be used inside ANSYS; however, with the release of PyAnsys, the code could be rewritten in order to work inside Python [41]. Furthermore, PyAnsys is a very recent library, and until now it was not able to use the topology optimisation capabilities of ANSYS, but this could change in the future. Thus, this development could provide an alternative method for topology optimisation in Python in the future.

7.5 Comparison

The previously discussed methods will be compared based on computation time. Matlab 3D is not considered in the comparison as the computation time is incomparably large. Table 16, shows clearly that Topos 3D has the shortest calculation time compared to the Python and Matlab codes, especially when the additional calculations are taken into account when using a 3D element instead of a 2D element. However, using a 3D topology optimisation will increase the number of required elements massively as a third dimension is added. For example, when 10 elements are used along the third axis, the computation time will be approximately five times as high as the Matlab 2D code, but the switch from 2D to 3D increases the usability and capabilities of the tool enormously. Finally, the table shows that ANSYS 3D has undeniably the longest computation time per iteration.

Methods	Time per 1 million elements	Time per 1 million nodes	time per 1 million DOFs
Python 2D	17	17	8.5
Matlab 2D	10.5	10.5	5.8
Topos 3D	4.84	1.57	0.52
ANSYS 3D	156	50.6	16.7

Table 16: The computation time per iteration in seconds of the different methods with 1,000,000 elements

Aside from the computation time, there are several differences. The most important advantage of the use of codes is the limitless flexibility and control of the whole optimisation process, for example, by choosing finite elements, altering mechanical properties such as the stiffness matrix, and incorporating density filters. Additionally, it is uncomplicated to integrate with other codes, and the 2D optimisation is relatively fast, especially in Matlab. However, the most important disadvantage is the slow optimisation process in 3D, which makes it impossible to use as 54,000 elements is just less than 40 x 40 x 40 elements and requires more than 1 minute per iteration. Furthermore, an excellent understanding of the code is required on both the mechanical and programming sides.

The Topos 3D plug-in also has multiple advantages. The primary advantage is the integration of the plug-in inside 3D visualisation software made to design and create. This results in an accessible way to 3D print the structure, as well as the possibility to use a fast array of different plug-ins and the capabilities of Rhino and Grasshopper. Beside the upsides, the main disadvantage is the limited control over the optimisation process. Only one finite element can be used, a box-shaped element with a node in each corner and three degrees of freedom per node, and multiple processes are black boxes. Another disadvantage is the absence of reliable stress and strain information, as explained in the previous sub-chapter on Topos 3D.

Finally, the most useful method for this tool is the ANSYS software because of the reliable stress and strain information.

8 Topology optimisation experiments

In this chapter, topology optimisation simulations have been performed with the intention of finding an optimal geometry for the core design of the facade element. The tests can be divided into two parts as a result of different boundary conditions. One of the tests used hinge supports along two edges, and the other tests used a hinge support along all four edges. First, the general outline of both tests is explained. Subsequently, the results of the conducted tests are displayed, followed by a discussion of the results with a closing remark.

8.1 Outline

The tests are performed on a sandwich structure. The geometrical properties are shown in Table 17. The width and length properties were chosen based on the available glass panel sizes. Furthermore, a relatively small facade element is used in order to limit the computation time. The simulations are performed in ANSYS 2022 R2.

Geometrical properties	[mm]
Total thickness	20
Thickness outer layers	2
Thickness core	16
Width	360
Length	760

Table 17: Geometrical properties

Subsequently, the boundary conditions of the tests. The applied load is an area pressure load of 5.4 kN/m^2 over the total area. The value is chosen based on the wind load. The wind load varies between roughly 1 and 5.4 kN/m^2, where the upper limit means a coastal area and a building height of 300 m. The chosen load is thus the upper limit of the wind load. The total applied load is equal to 1.48 kN. No cavity pressure was used in this experiment for simplicity. Additionally, in the case where two hinge supports are used, the hinge supports are placed on the shortest faces. This means that the supports are placed on the sides to create the maximum distance between them. The other case placed the hinge supports on all four sides. The hinge support defined in ANSYS has fixed all translations and two of the three translations. Also, in ANSYS, it is possible to specify the behaviour of the support, and this is set to rigid. Furthermore, only one material is used in the experiments instead of using a different material for the core and the outside layers. This has been done for simplicity. The material used is chemically strengthened glass. The material properties are displayed in Table 18. The material properties originate from NEN-EN2608 [57].

Material properties	
Young's modulus	70 [GPa]
Poisson's ratio	0.23 [-]
Density	2500 $[kg/m^3]$
Yield strength	120 $[N/mm^2]$

Table 18: Material properties

Additionally, solid linear cubical mesh elements are used with a size of 2x2x2 mm. This implies that the outer layers are 1 element thick and the core consists of 8 elements. Furthermore, a total of 684,000 elements are used (180x380x10), which results in 758,571 nodes. The main reason for the used element size is to have an outer layer thickness of 2 mm. The simulation uses a linear structural finite element analysis.

8.1.1 Objective and constraints

The facade panel has to fulfil the ULS and SLS requirements, which can be found in NEN-EN 2608. Only the requirements for the outside glass layers are considered, because of simplicity, and no difference has been made between core and outside material properties. The ULS requirements would be a maximum Von-Mises constraint set to 120 N/mm^2, which is the yield strength of the glass as shown in Table 18. The SLS requirement would be a maximum deflection of 12.94 mm. This is calculated based on equation 13 in chapter 9.3 [57]. Therefore, the two constraints are defined, and the objective is to minimise the total volume (i.e., mass) of the facade panel.

The simulation in the case of two hinge supports required 126 iterations in 54 minutes. The core geometry is displayed in Figure 23. This clearly shows that, as a result of the constraints, almost no core material is required to keep the displacement and stress below the defined threshold. The deformation and von-Mises stress are visible in Figures 24 and 25. A maximum stress of 88 MPa is present in the panel; however, this is a local extreme value and may be caused by a stress concentration.

Figure 23: Side view that displays the core geometry.

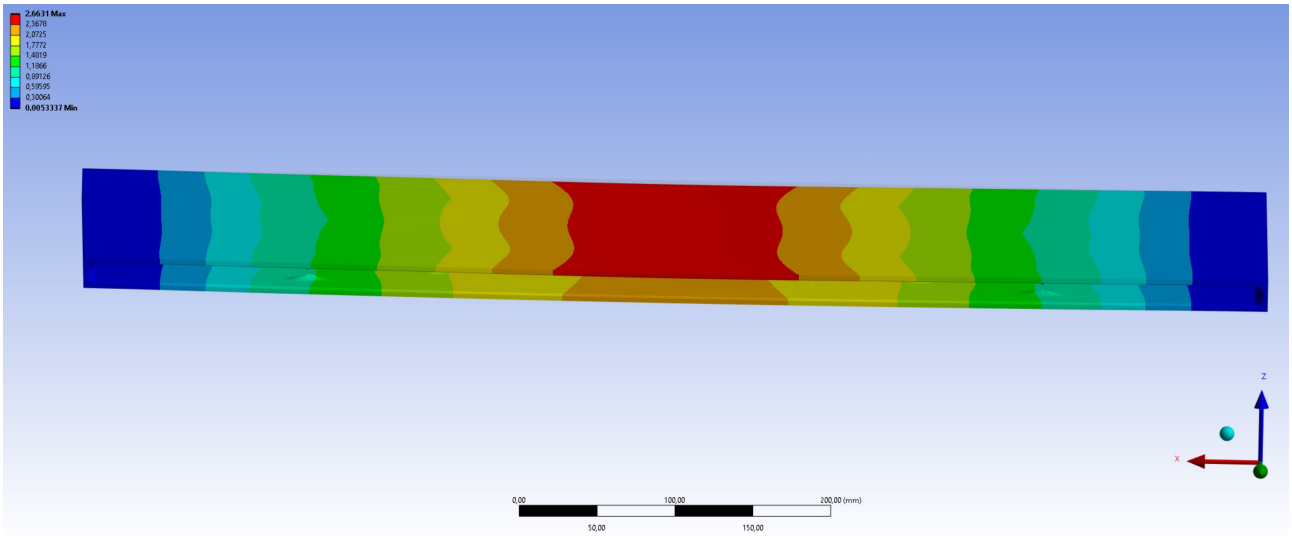


Figure 24: Top view of the deformations in millimetres.

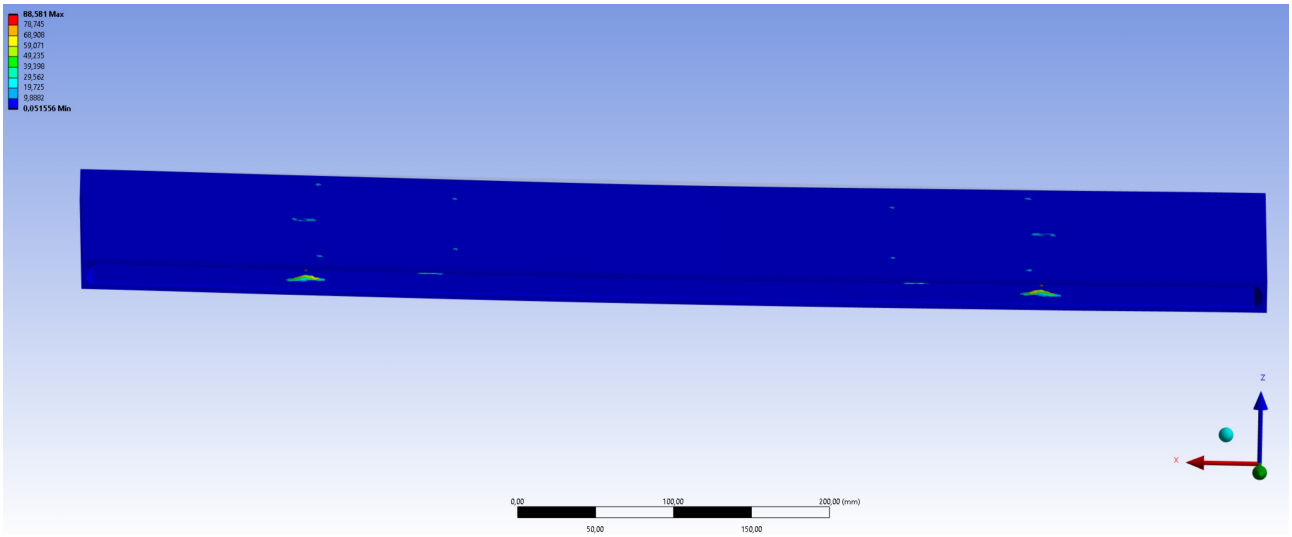


Figure 25: Top view of the equivalent von-Mises stress in MPa.

The simulation above shows that the given constraints result in a core that consists of very little material, and no core pattern appears, which could provide important insights. Furthermore, importance was placed on having the same objective and constraints in the simulations with two and four supports. Therefore, the chosen constraint is a mass constraint to understand the effect of different core volumes on the core geometry. Thus, the previous objective is changed into a constraint. Subsequently, the objective is changed to compliance minimisation which steers the solution to the stiffest geometry with the given mass constraint. This is a typical

set-up for topology optimisation. Finally, the chosen objective is compliance minimisation and a mass constraint. The mass constraint is set equal to 40 %, 30 %, and 20 % of the total possible mass in case the whole volume is filled with material in order to create a sandwich panel. This means that in total, six optimisations are performed: two hing supports with a mass constraint of 40 %, 30%, or 20 %, and four hinge supports with a mass constraint of 40 %, 30%, or 20 %. Next, a convergence criterion of 0.1 % change in the total mass is used, which is the default value of ANSYS and affects the number of iterations required to find the near-optimal solution. This means that if the change in mass is less than 0.1 %, the iteration process is stopped.

8.2 Results of facade element with two hinge supports

In order to have a complete understanding of the effect of the mass constraint, a first look is given to the case that the core has material throughout the entire mass. The displacement, normal stress, and shear stress are visible in Figures 26, 27, and 28. This shows that the displacement is only 0.513 mm, and the maximum normal and shear stress are 5.95 and 0.543 N/mm^2. In Figure 27 it is visible that the normal stress around the hinge supports are equal to zero, which means that the supports indeed act as a hinge instead of a clamped support. Also, the shape of the displacement shows that the support is free to rotate in the x-direction, which is the only free translation.

An analytical calculation is performed as a sanity check based on the beam theory. It uses Equation (45), and the obtained deflection is 0.503 mm. Although the analytical calculation uses a simplified representation of the plate as a beam, the maximum deflections show great similarity.

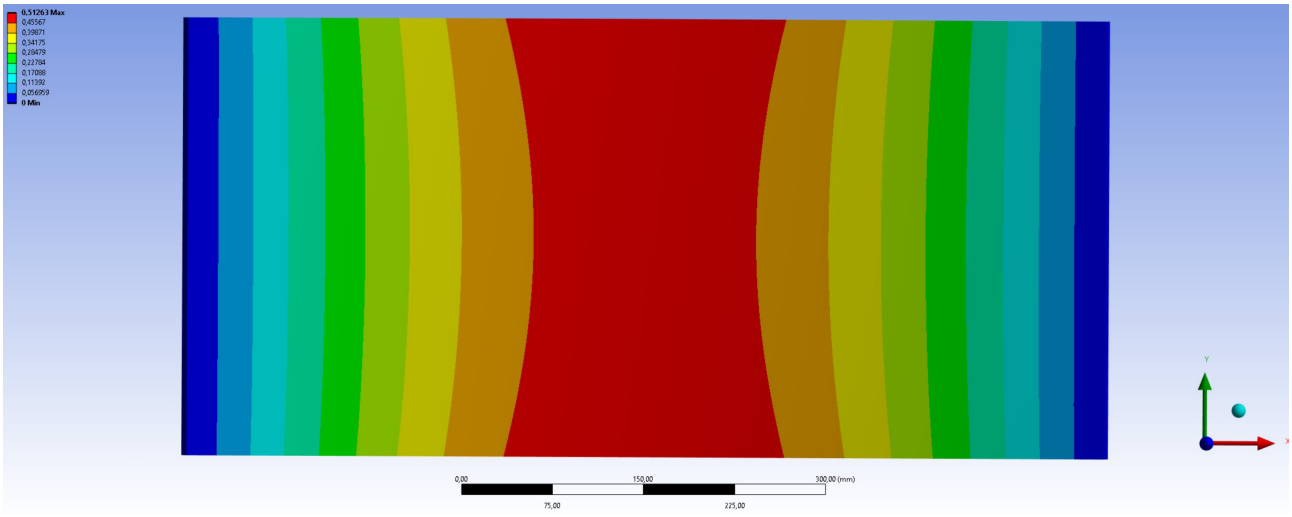


Figure 26: Top view of the displacement (mm) in case of a solid core.

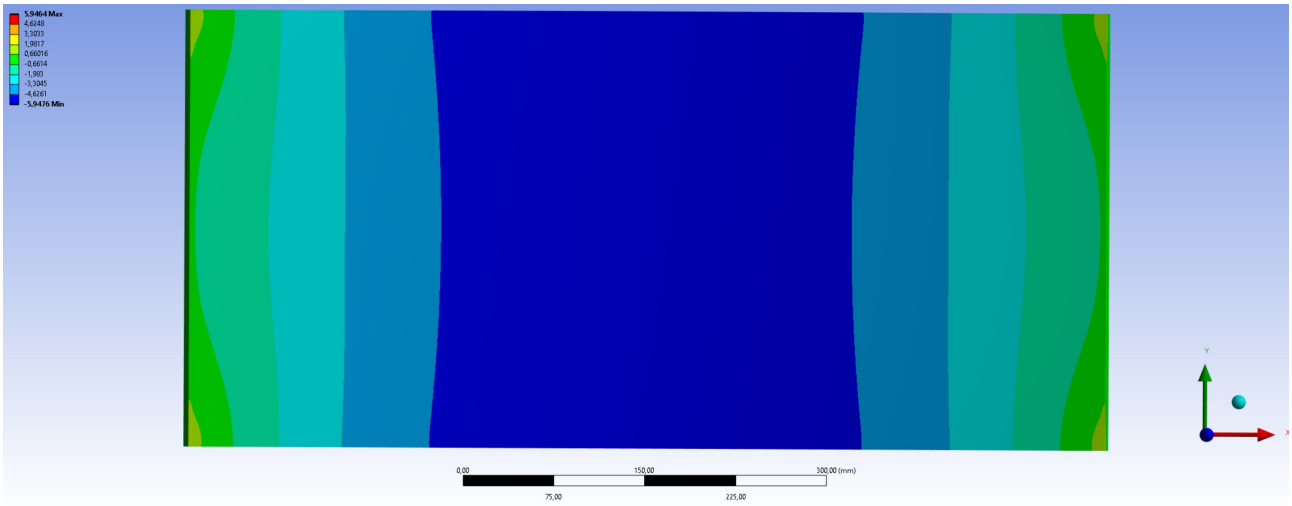


Figure 27: Top view of the normal stress (N/mm^2) in x-direction in case of a solid core.

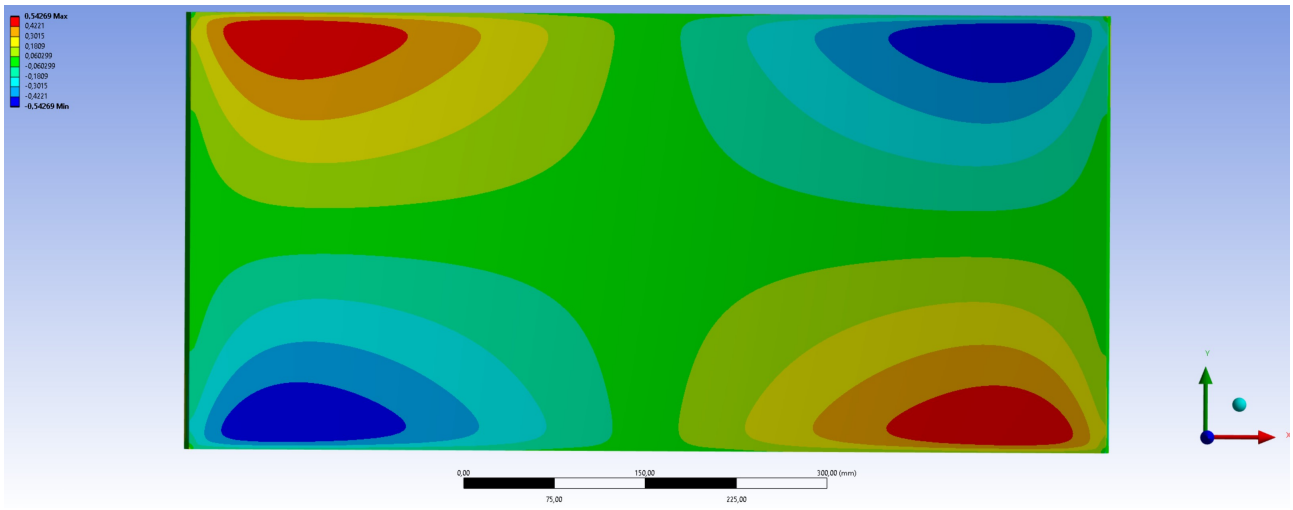


Figure 28: Top view of the shear stress (N/mm^2) in xy-direction in case of a solid core.

Subsequently, a topology optimisation has been conducted with a mass constraint of 40 % of the total possible mass. This took 28 iterations and 36 minutes to get a solution with a mass percentage of 40.266 %. The maximum displacement and maximum normal stress have increased, as shown in Figures 29, 30. Additionally, the shear stress has increased as well, as depicted in Figure 31. Then, the convergence and fulfillment of the mass constraint are visible in Figure 32. The above graph shows the change between individual iterations in terms of compliance. This shows that the mass constraint is already fulfilled in the first iteration, and then the software optimises the material distribution. At last, a mid-plane cross section is depicted in Figure 33. This is a bottom view of the center of the plate. The lines with a sharp boundary, such as the lines near the support and the dots in the center, represent material that is connected to the bottom and upper plates. In contrast, the thicker lines with softer boundaries represent material that is only connected to the top panel.

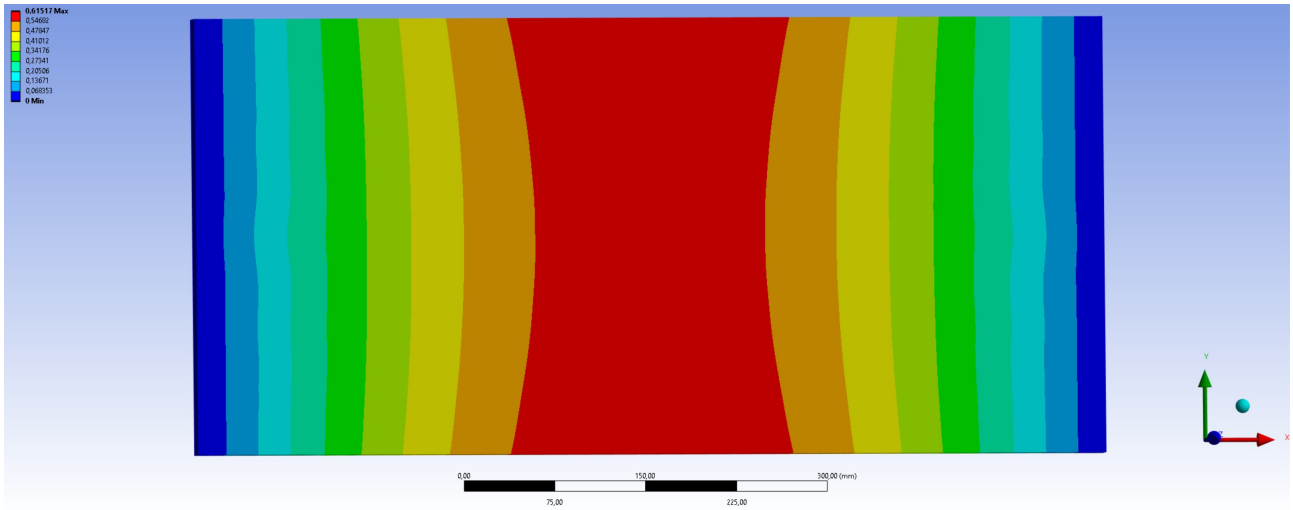


Figure 29: Top view of the displacement (mm).

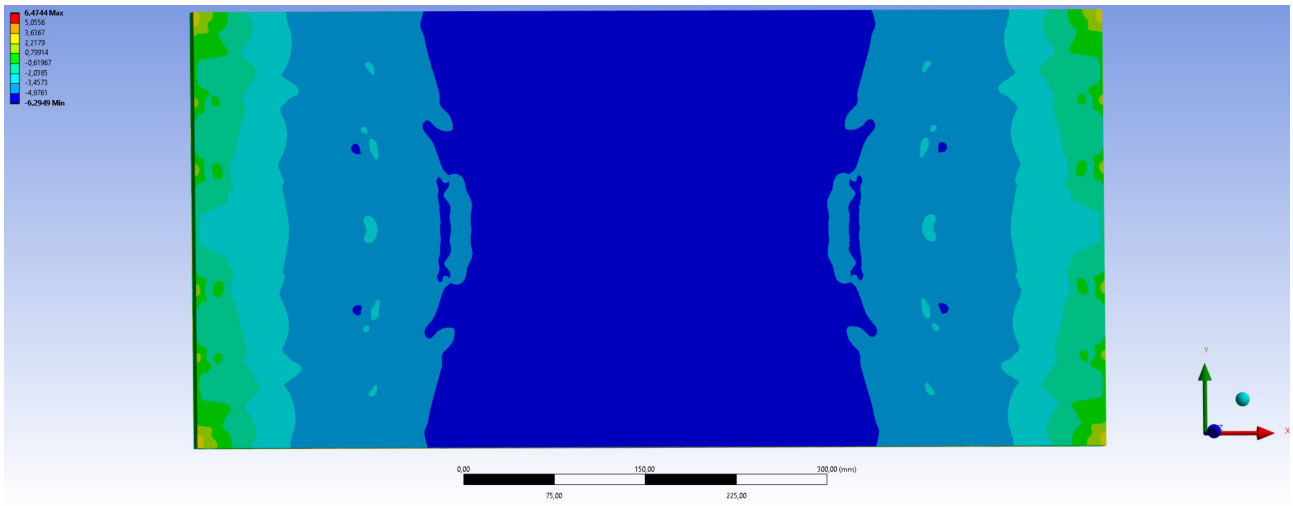


Figure 30: Top view of the normal stress (N/mm^2) in x-direction.

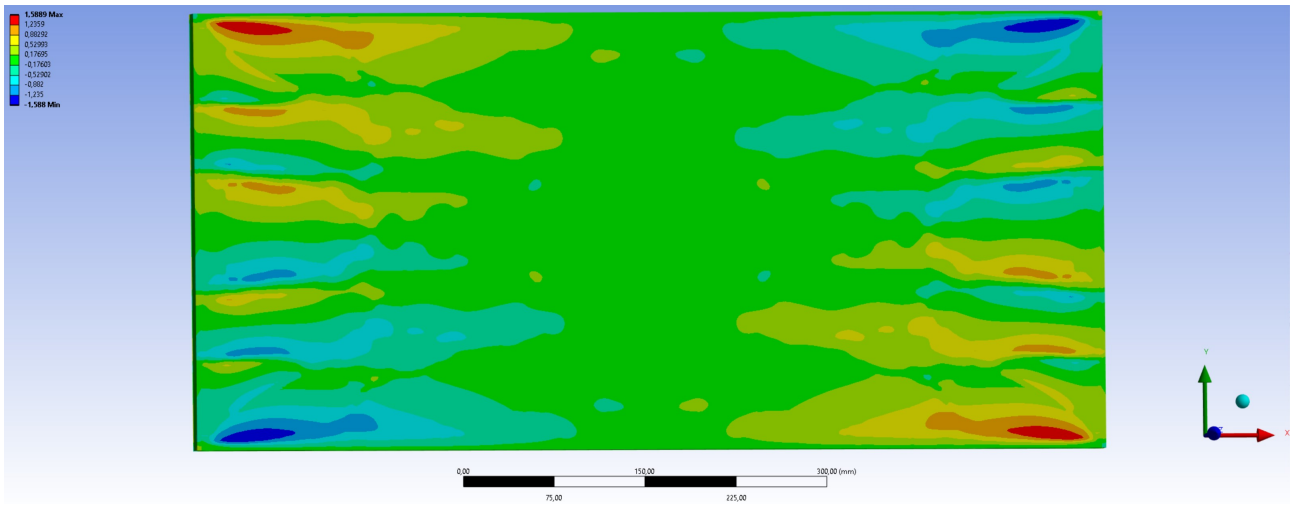


Figure 31: Top view of the shear stress (N/mm^2) in xy-direction.

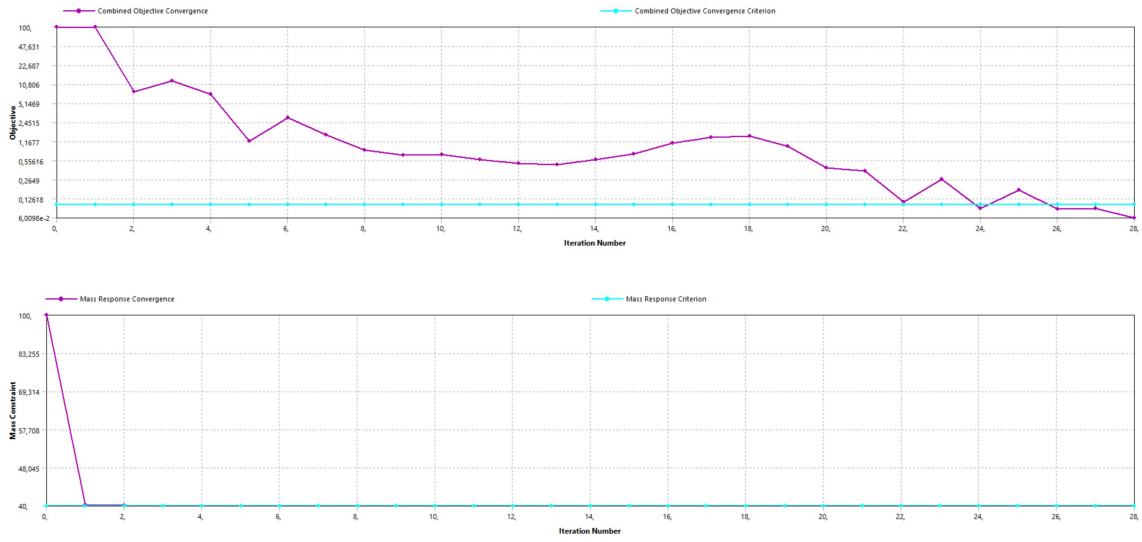


Figure 32: The convergence rate as well as the fulfillment of the mass constraint.

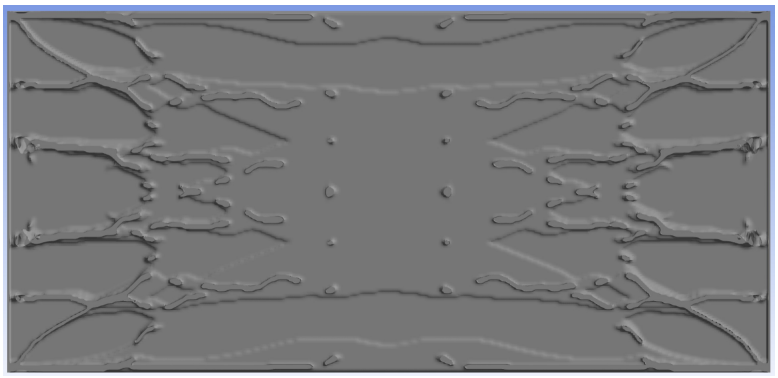


Figure 33: A mid-plane cross section.

The next optimisation uses a mass constraint of 30 % of the total possible mass. This took 26 iterations and 37 minutes, with a final volume percentage of 30.256 %. The displacement, normal stress, shear stress, convergence rate, and core geometry are visible in Figures 34, 35, 36, 37, and 38. The maximum displacement is 0.678 mm, and the absolute maximum normal stress is 7.16 N/mm^2. The maximum shear stress is equal to 2.09 N/mm^2. The convergence rate and mass constraint show a similar behaviour as in the previous experiment. The effect of the lower mass constraint is clearly visible in the core geometry.

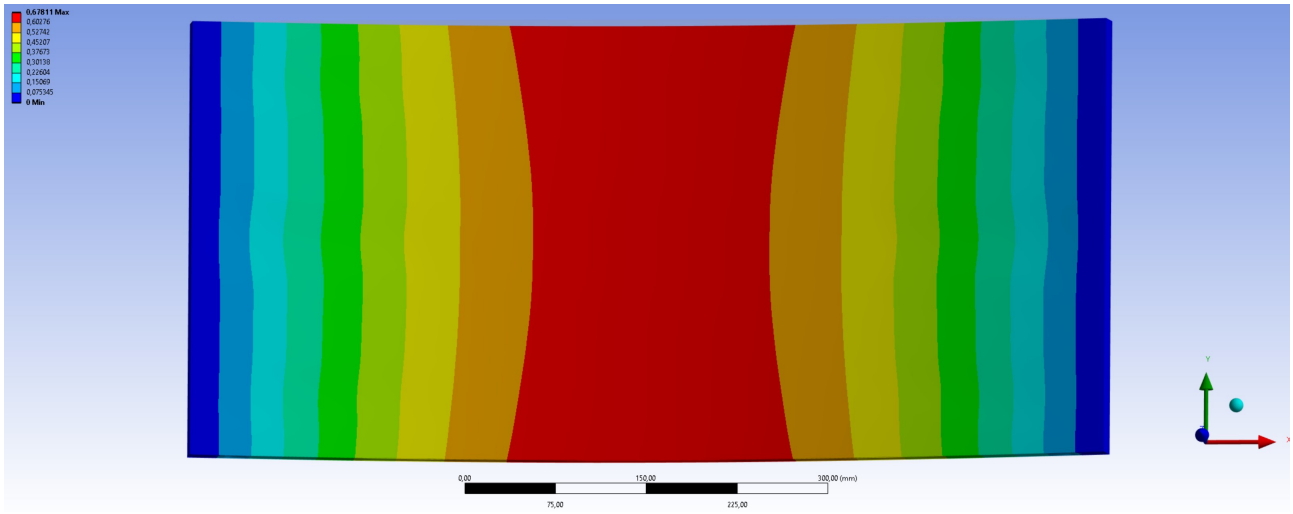


Figure 34: Top view of the displacement (mm).

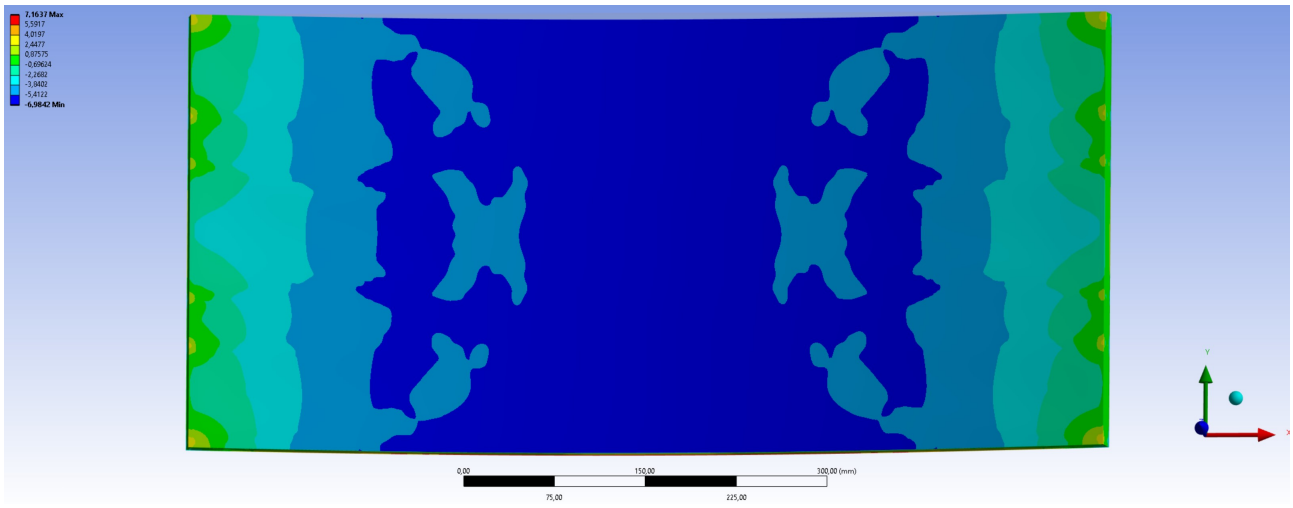


Figure 35: Top view of the normal stress (N/mm^2) in x-direction.

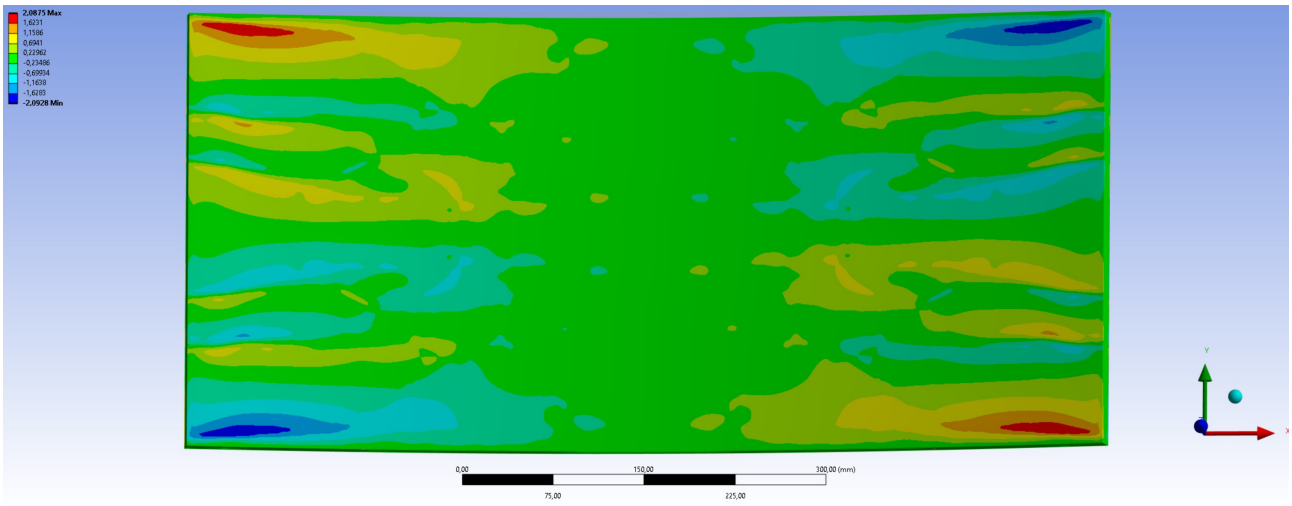


Figure 36: Top view of the shear stress (N/mm^2) in xy-direction.

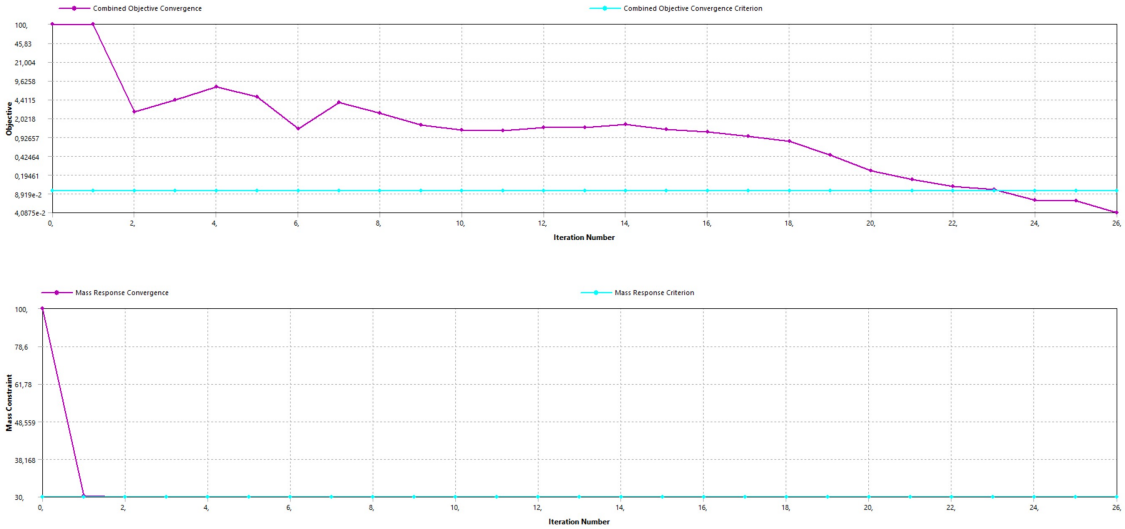


Figure 37: The convergence rate as well as the fulfillment of the mass constraint.

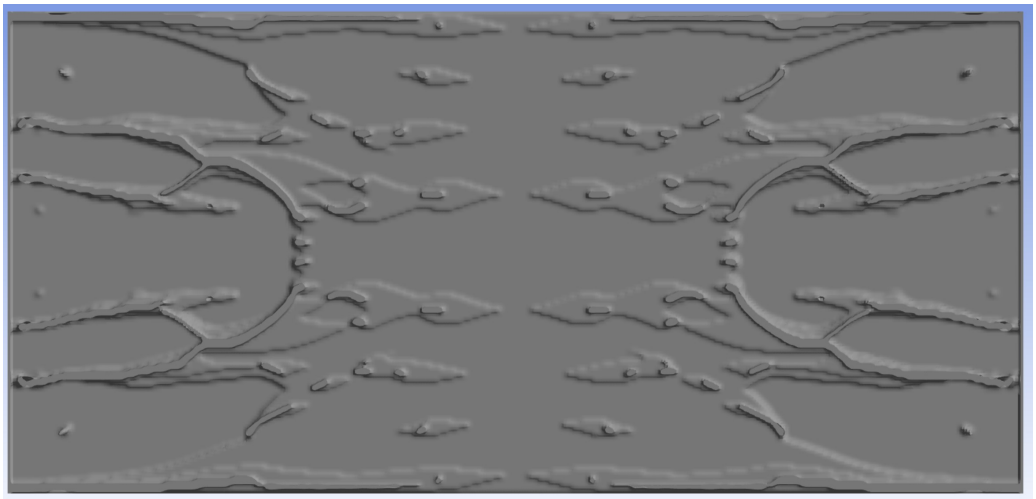


Figure 38: A mid-plane cross section.

The last simulation with two hinged edge supports uses a mass constraint of 20% of the total possible mass. This required 47 iterations and 67 minutes. The final volume percentage is 20.111 %. The displacement, normal stress, shear stress, convergence rate, and core geometry are visible in Figures 39, 40, 41, 42, and 43. The maximum displacement has increased significantly compared to previous simulations and has a maximum of 0.80 mm. Additionally, this simulation required more iterations, and Figure 42 shows that in the last 20 iterations, the solution changed less than 1 % per iteration. Finally, the core geometry has less material in the middle of the panel compared to the result with a 30% mass constraint. Furthermore, some cylindrical shapes have appeared in the middle of the panel that connect both outside layers. This was not present in the previous simulation. Finally, a distortion is visible in the figures displaying the shear and normal stress. However, the distortion is not present at the bottom of the panel.

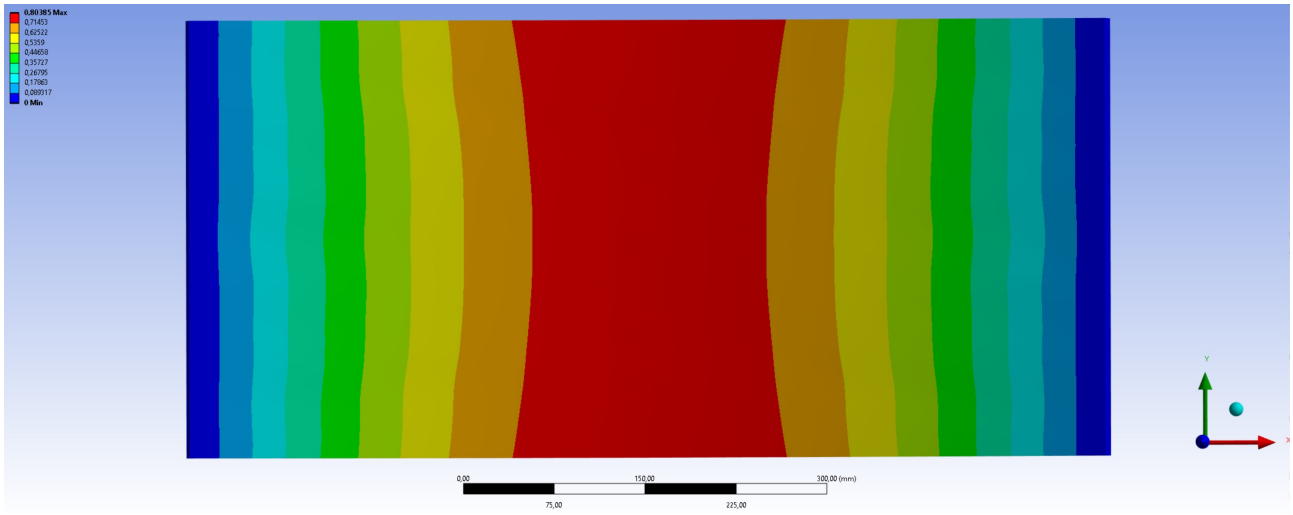


Figure 39: Top view of the displacement (mm).

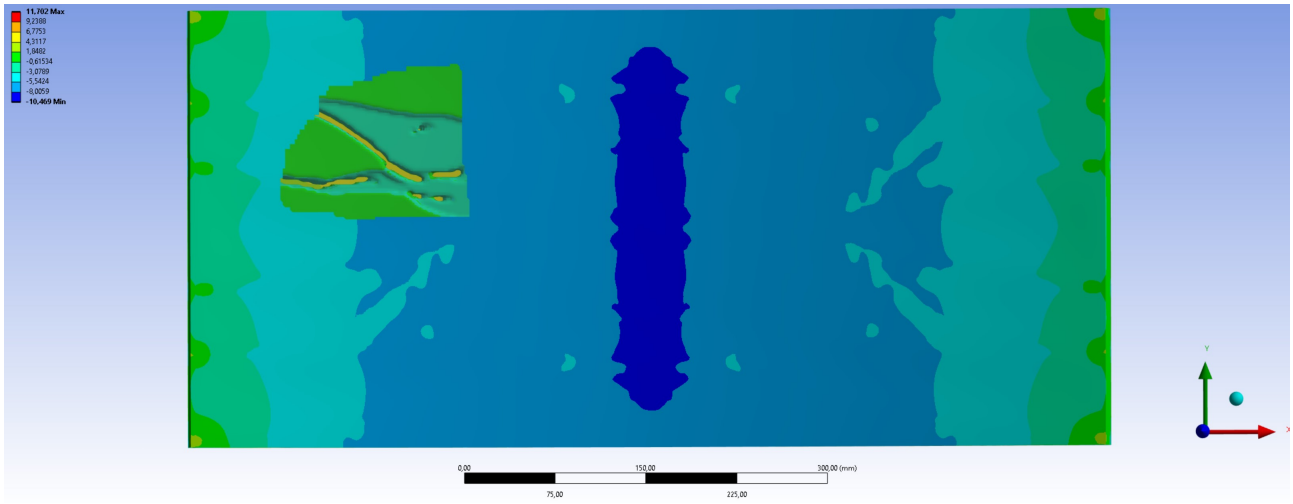


Figure 40: Top view of the normal stress (N/mm^2) in x-direction.

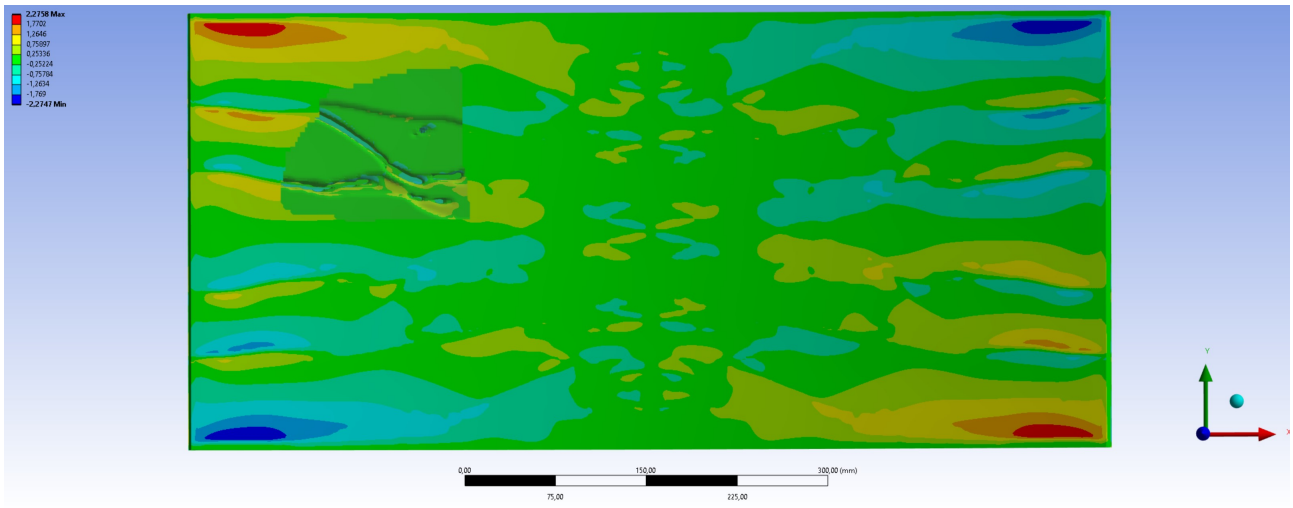


Figure 41: Top view of the shear stress (N/mm^2) in xy-direction.

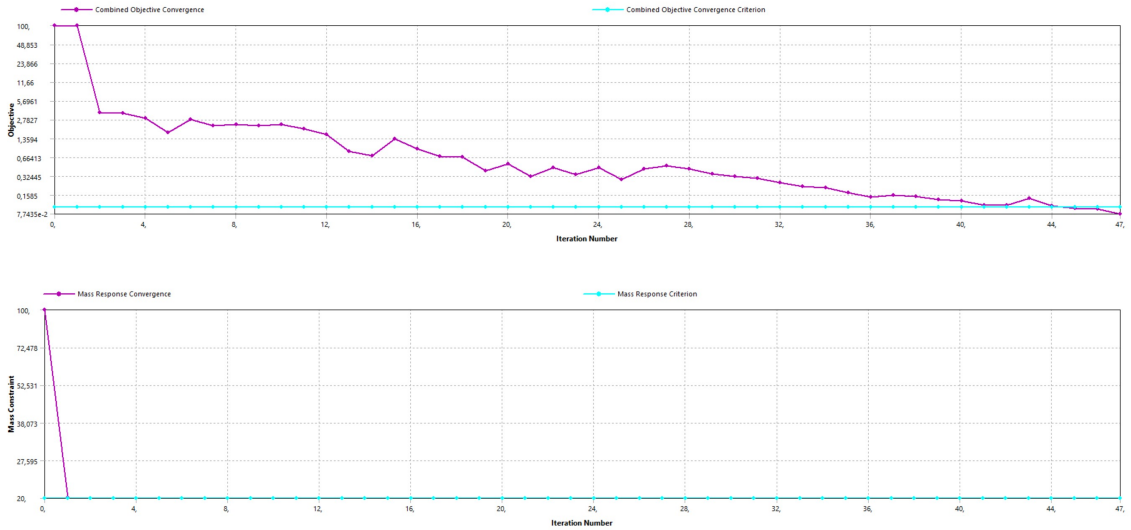


Figure 42: The convergence rate as well as the fulfillment of the mass constraint.

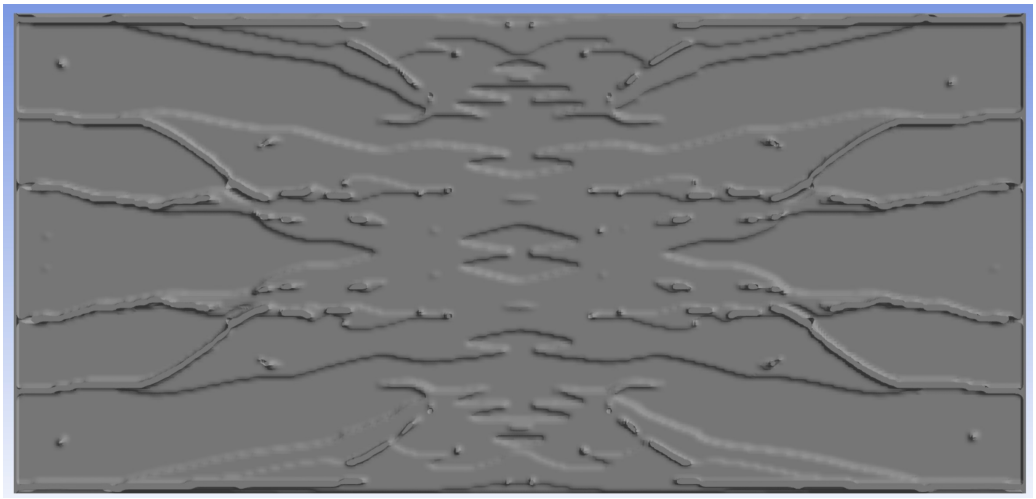


Figure 43: A mid-plane cross section.

8.3 Results of facade element with four hinge supports

First, the case of a solid core is examined. The maximum displacement is equal to 0.00489 mm, which is much smaller than the simulations with two supports along the faces. The normal stress and shear stress have an absolute maximum of 0.69 and 0.119 N/mm^2 respectively. The results are visible in Figures 44, 45, and 46. In Figure 46 a well-known pattern of the shear stress is depicted.

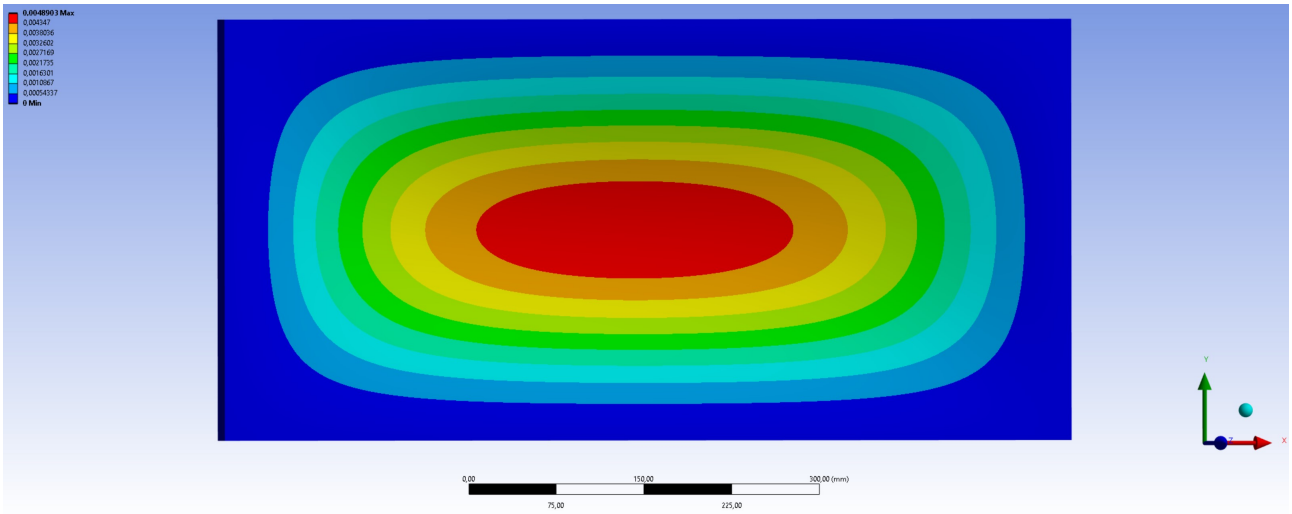


Figure 44: Top view of the displacement (mm) in case of a solid core.

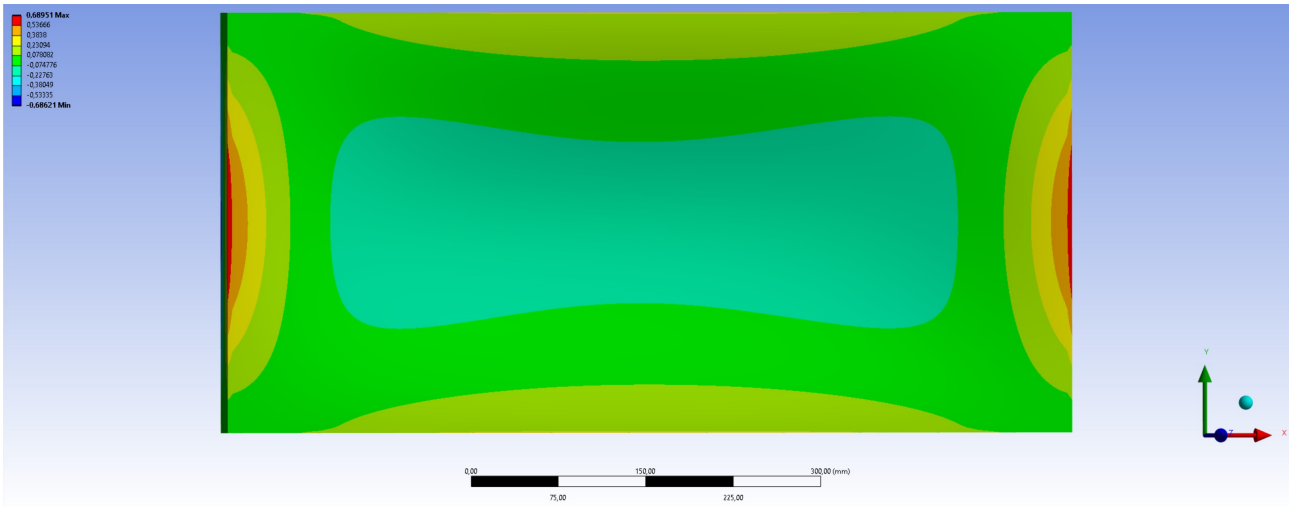


Figure 45: Top view of the normal stress (N/mm^2) in x-direction in case of a solid core.

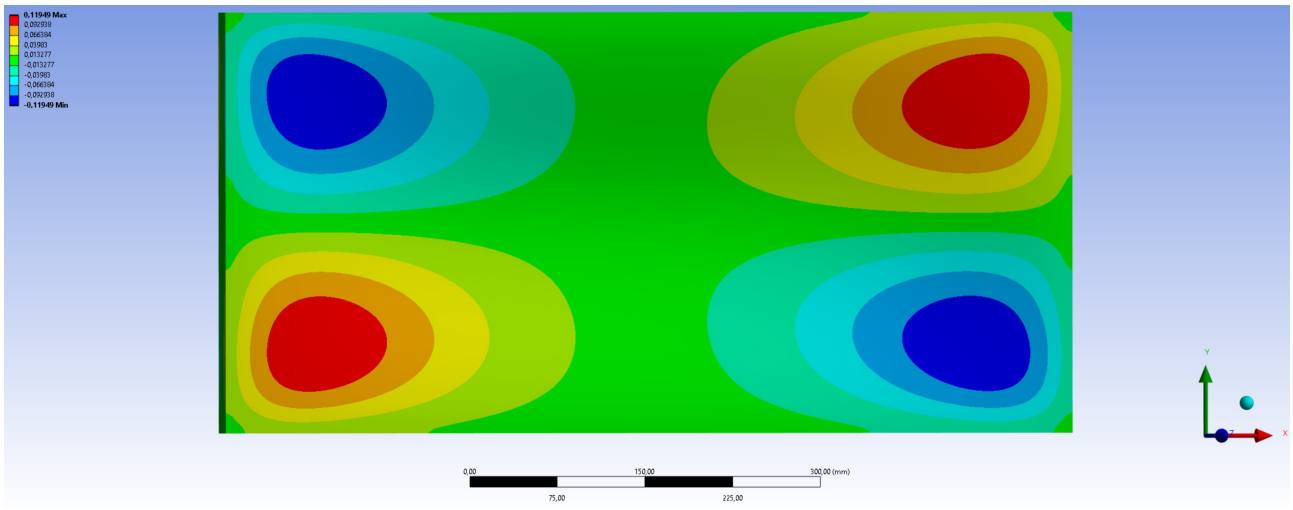


Figure 46: Top view of the shear stress (N/mm^2) in xy-direction in case of a solid core.

The first simulation uses a mass constraint of 40 % of the total possible mass. Figure 47 shows that the deformation has increased to 0.00657 mm. Despite the small increase in normal stress displayed in Figure 45, the shear stress has almost doubled compared to the base case, which is shown in Figure 49. Subsequently, in Figure 50, information is given related to the optimisation program, which shows that 42 iterations were required and it took 75 minutes. Finally, the core structure is visible in Figure 51. This clearly shows the yield line of the plate and how the core is situated around it. The yield line is the line along which the plate will crack and is commonly used in the design of reinforced concrete plates [43].

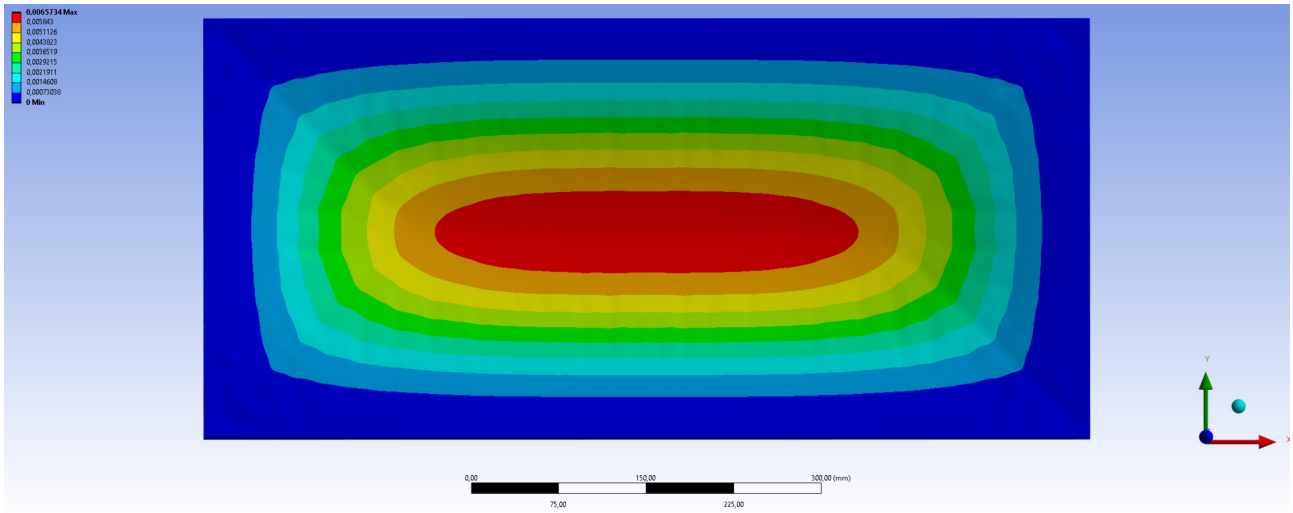


Figure 47: Top view of the displacement (mm).

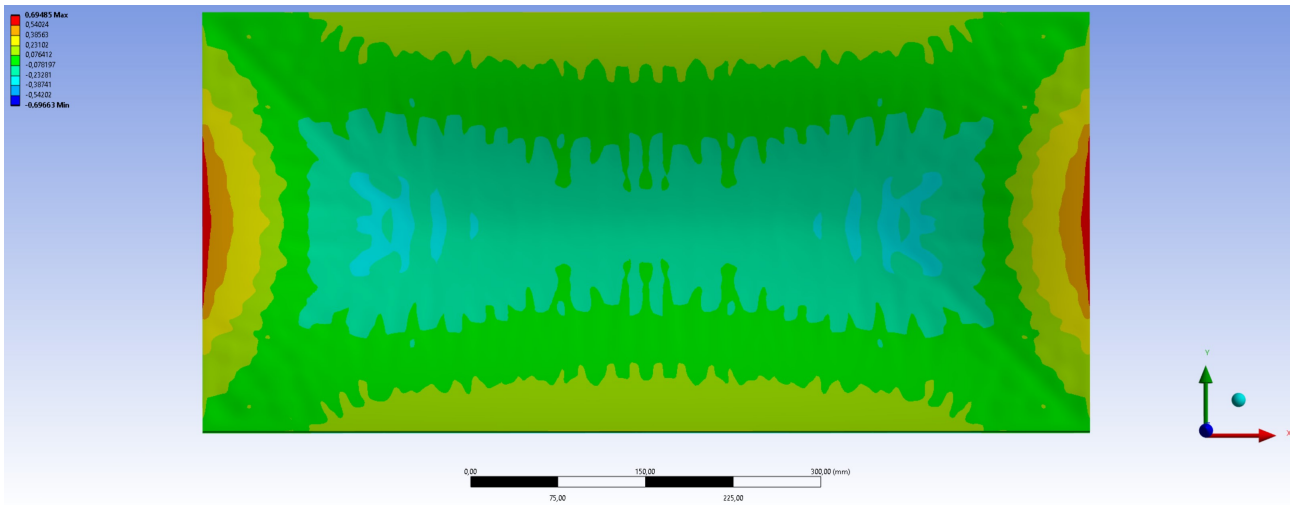


Figure 48: Top view of the normal stress (N/mm^2) in x-direction.

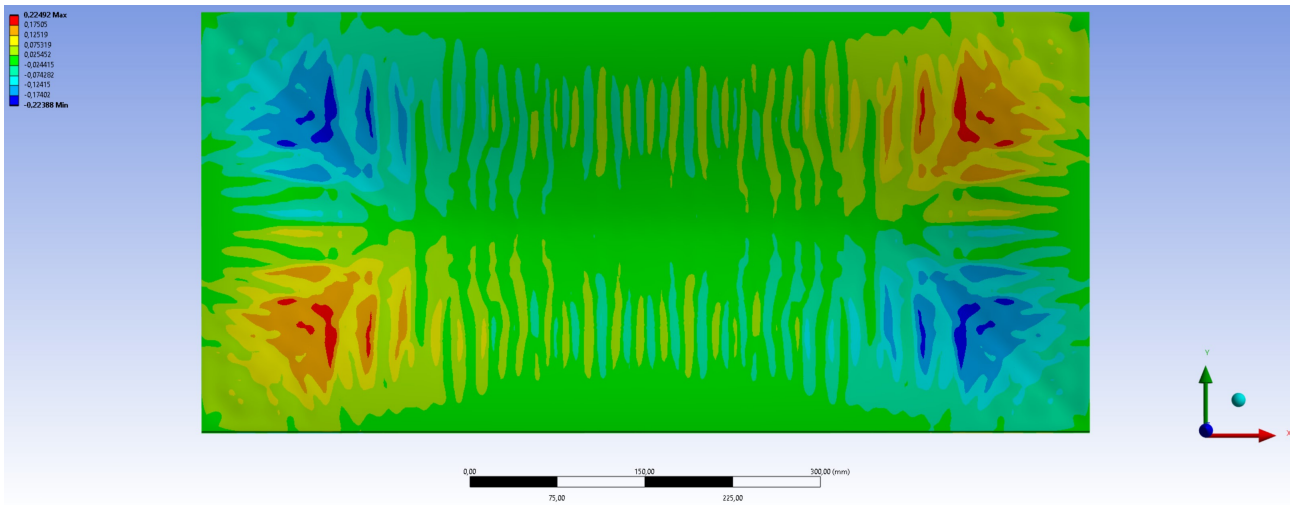


Figure 49: Top view of the shear stress (N/mm^2) in xy-direction.

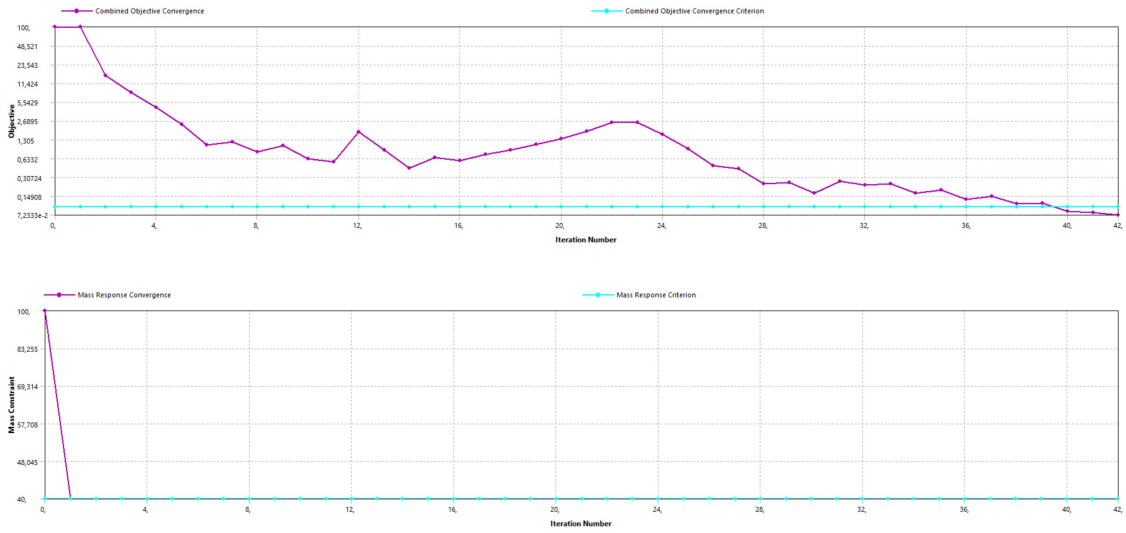


Figure 50: The convergence rate as well as the fulfillment of the mass constraint.

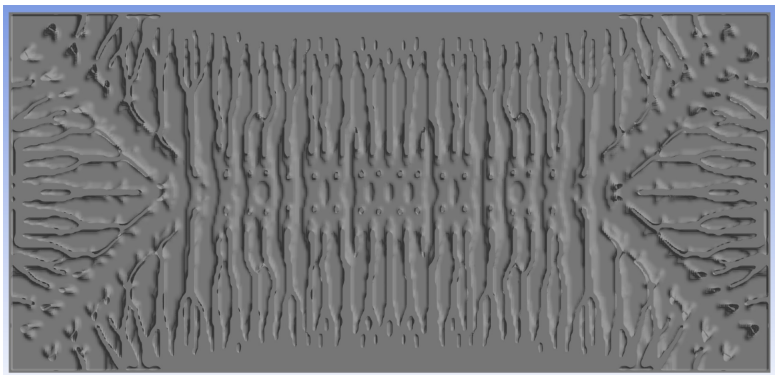


Figure 51: A mid-plane cross section.

The second simulation uses a mass constraint of 30 % of the total possible mass. The deformation, normal, and shear stress are visible in Figures 52, 53, and 54. While in the base case the elliptical contour lines are smooth, in this case the lines show local effects due to the core structure. In Figure 55, the convergence rate shows a very similar pattern compared to the previous simulation. At last, in Figure 56 the core structure is displayed, which looks very similar to the last simulation but has fewer lines running from the edges to the yield lines.

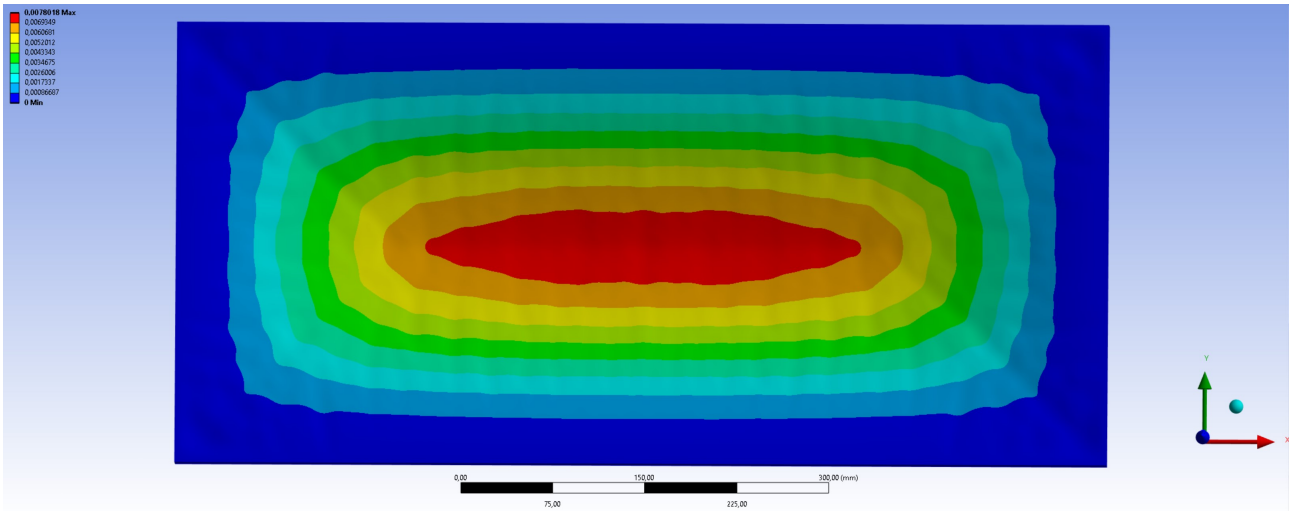


Figure 52: Top view of the displacement (mm).

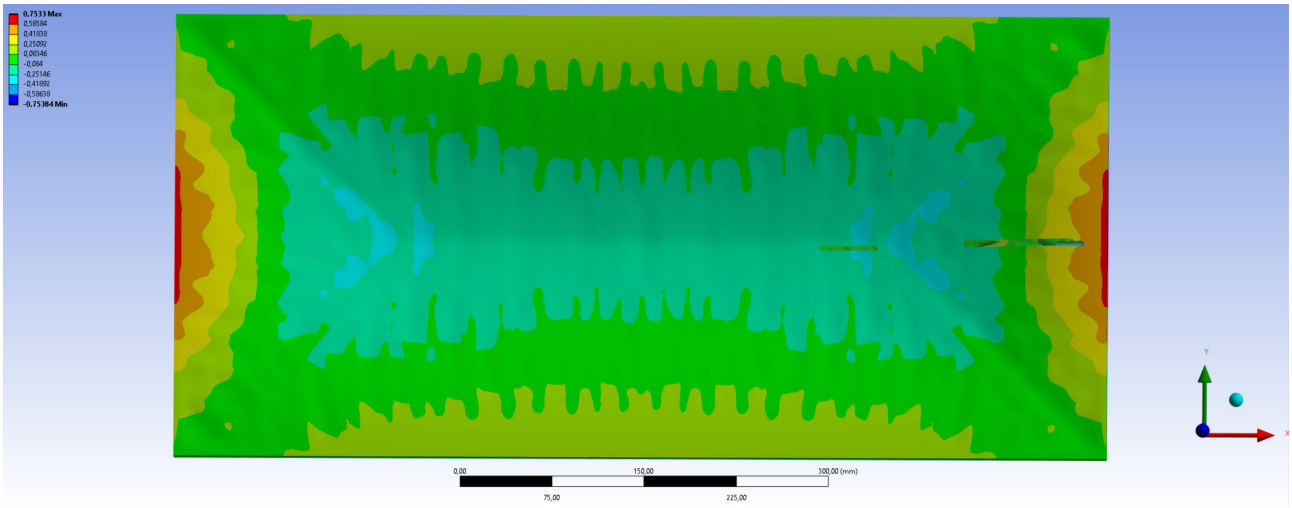


Figure 53: Side view, of the middle of the panel, of the normal stress (N/mm^2) in x-direction.

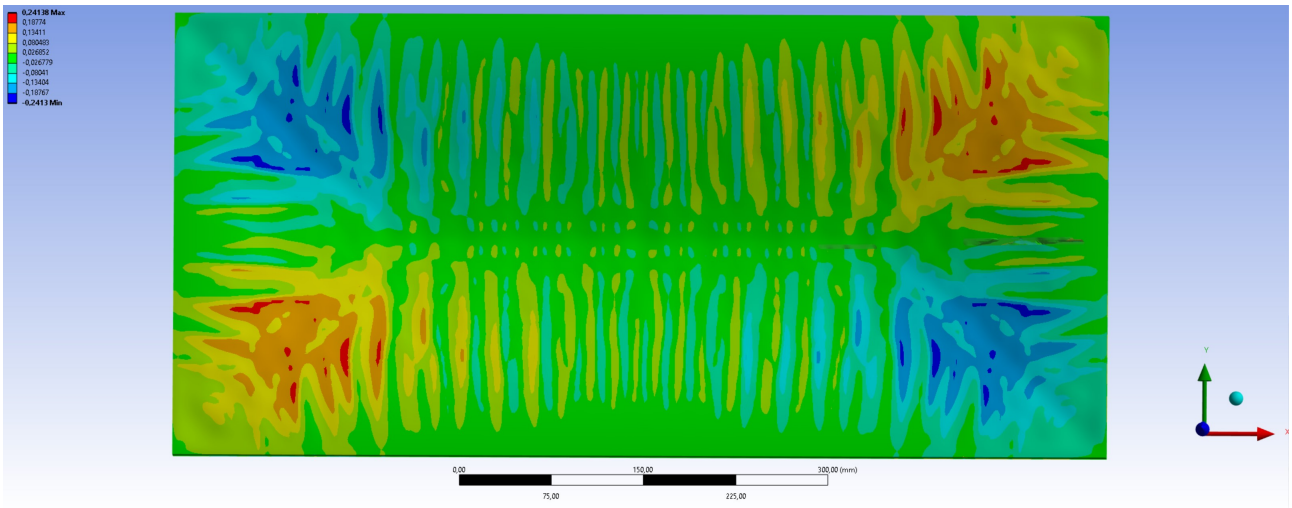


Figure 54: Top view of the shear stress (N/mm^2) in xy-direction.

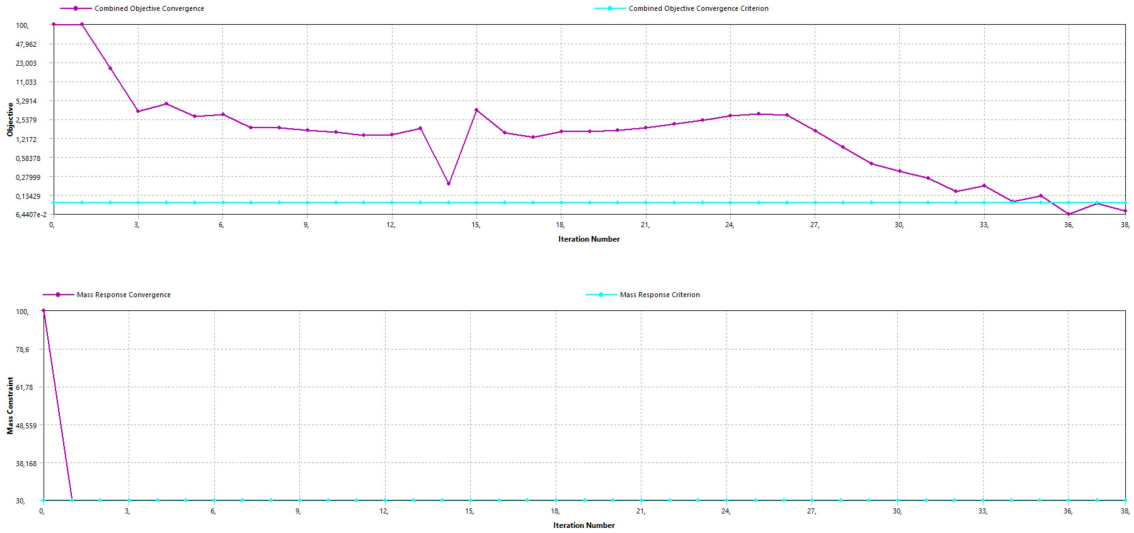


Figure 55: The convergence rate as well as the fulfillment of the mass constraint.

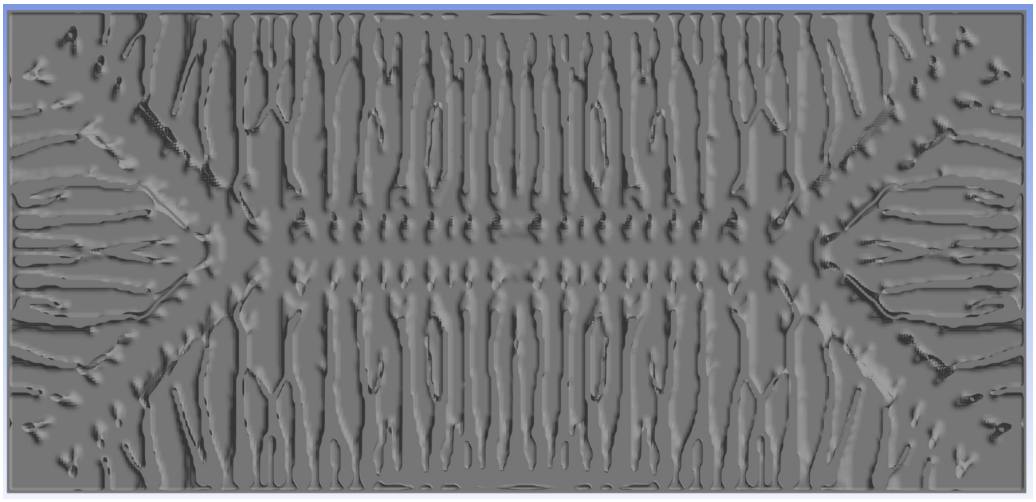


Figure 56: A mid-plane cross section.

The last simulation uses a mass constraint of 20 % of the total possible mass, and the final volume percentage is 20.134. The deformations, normal and shear stress, have increased clearly compared to the mass constraint of 30 % and are shown in Figures 57, 40, and 41. This simulation required 50 iterations and took almost 2 hours, as shown in Figure 60. Finally, the core structure is depicted in Figure 43. This still has the same general shape as the simulation, with a mass constraint of 40 and 30 %, however, the number of lines between the yield lines and the edges has further decreased.

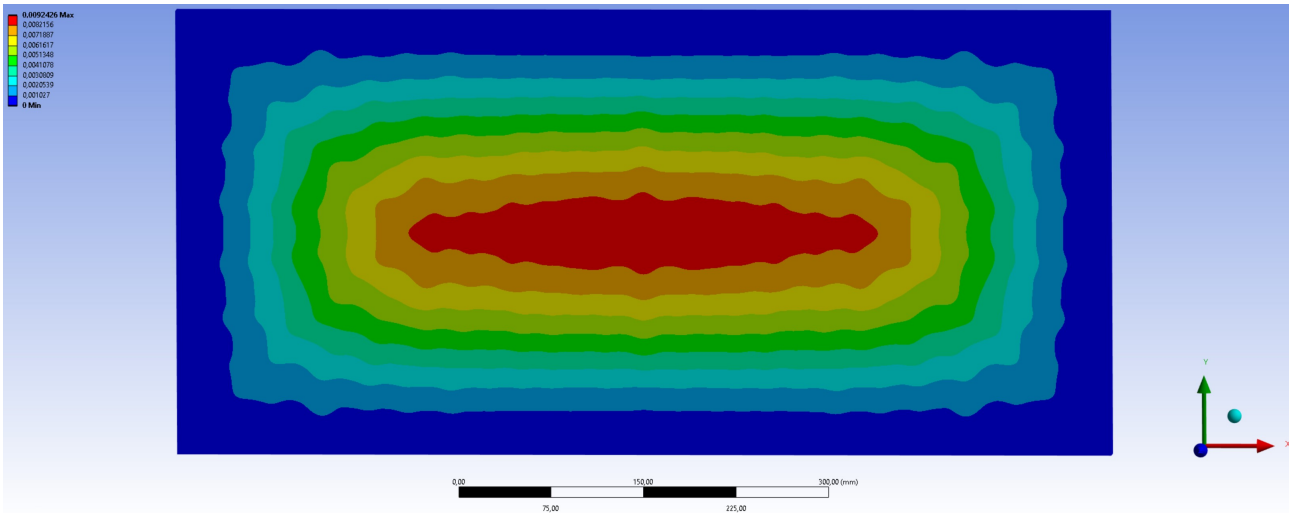


Figure 57: Top view of the displacement (mm).

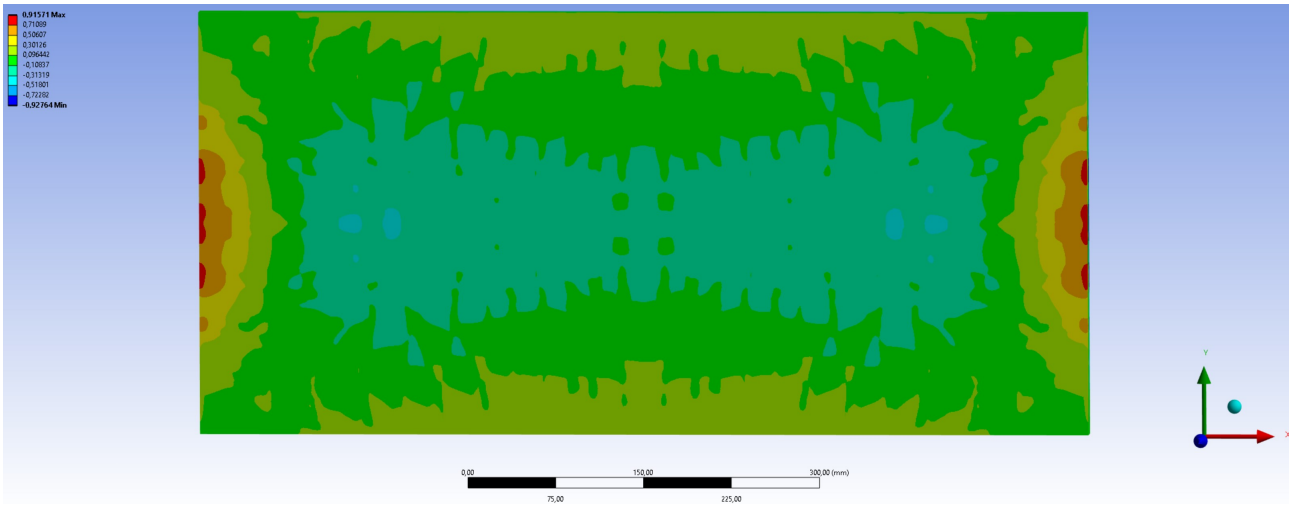


Figure 58: Side view, of the middle of the panel, of the normal stress (N/mm^2) in x-direction.

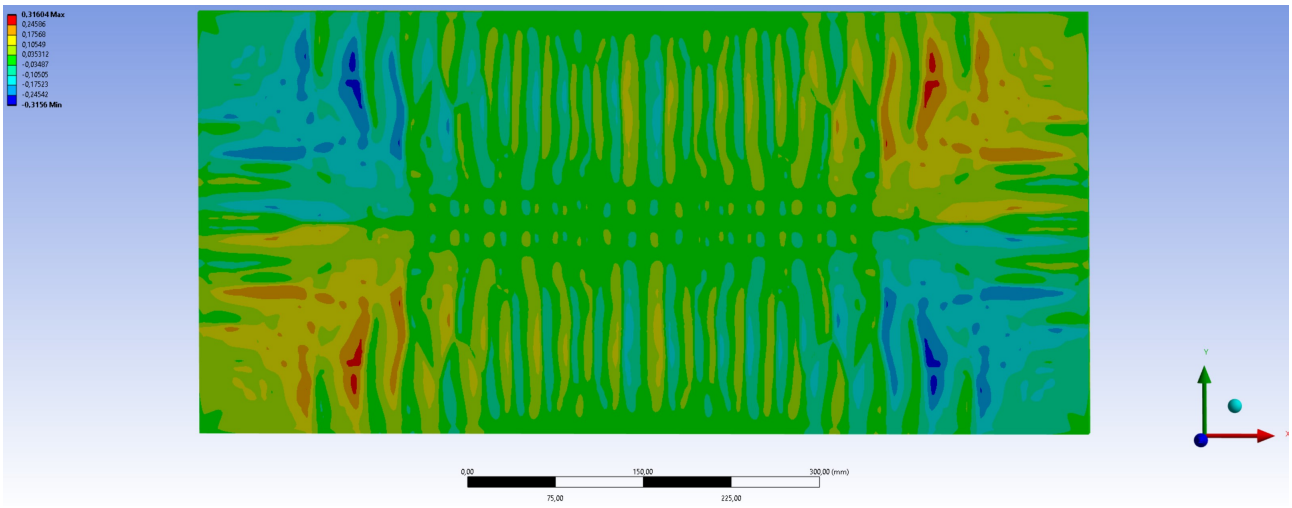


Figure 59: Top view of the shear stress (N/mm^2) in xy-direction.

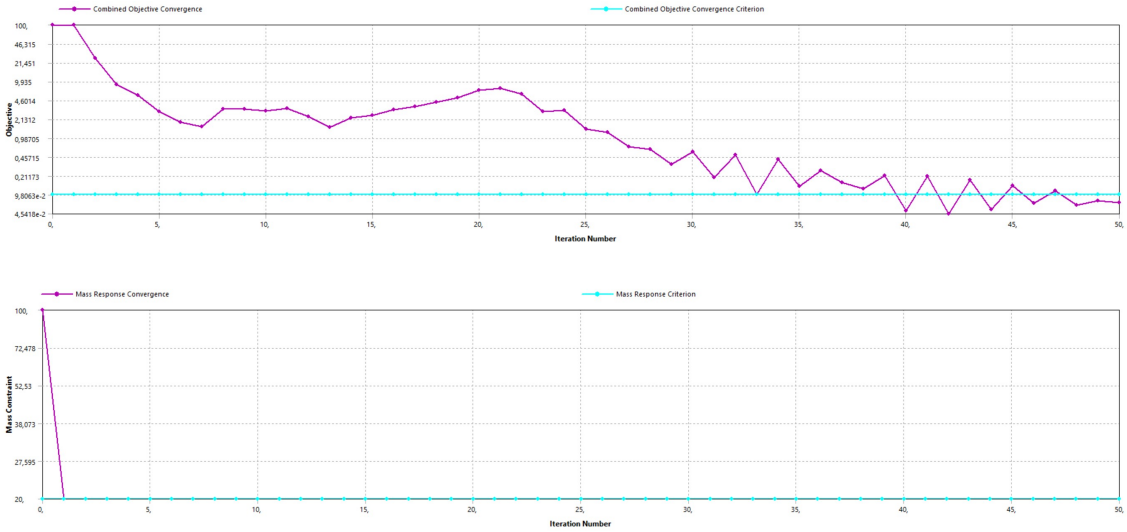


Figure 60: The convergence rate as well as the fulfillment of the mass constraint.

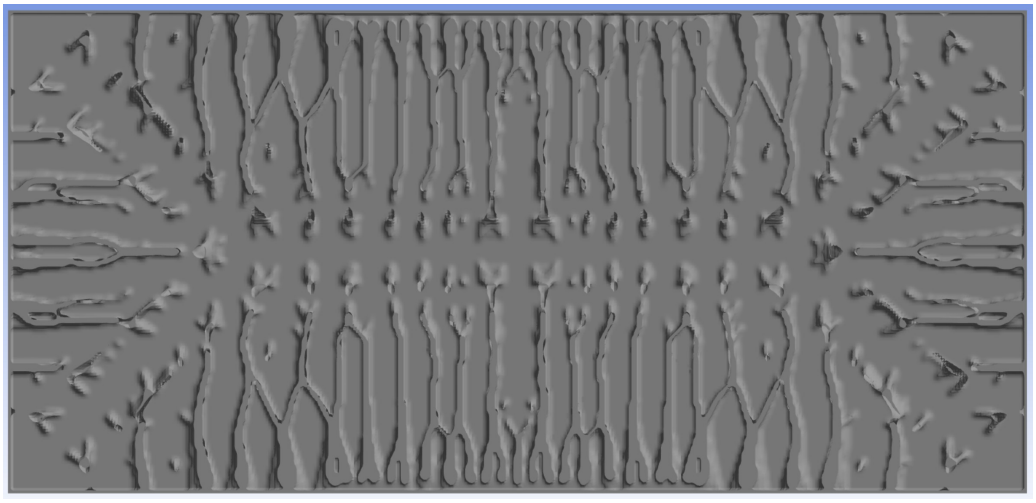


Figure 61: A mid-plane cross section.

8.4 Conclusion

The results of the simulations are summarised in Tables 19 and 20. The tables show the maximum deflection, normal stress, and shear stress, as well as the relative values. The relative values give an easier insight into the relationship between the different simulations. In the tests with two hinge supports, the maximum displacement is increased by 54 % with a core volume of 20 % of the total possible mass. The normal stress has increased by 97 %, but a very large increase took place between 30 % and 20 % of the core volume. The shear stress has increased the most, and with a mass constraint of 20 % the shear stress is 320 % higher. This shows the shear stress increases mainly as the core volume decreases. Thus, a core structure that can withstand shear stress is required. Furthermore, the core geometry in the simulations shows lines connected to both outside layers only near the support, which is the same location of the maximum shear stress. Also, in the middle of the facade panel, the core only has cylindrical connections between both outside layers. Most of the core material in the middle is situated near the outside layers to increase the local thickness of these layers and therefore reduce the normal stress.

In the case of four hinge supports, the maximum deflection increased by 89 % compared to the base case. Interestingly, the maximum normal stress only increased by 34 %, especially compared to the case with two hinge supports; this increase is very minimal. At last, the shear stress maximum increased by 166 %, and shows the largest increase. Therefore, if the facade panel has four hinge supports, a core that can withstand shear stress is a requirement. Another interesting aspect of the core geometry is that the yield lines are clearly visible in the core structure and play an important role in the design.

Furthermore, the computation time is longer in the case of four hinge supports, and in general, a lower core volume requires more iterations to find an optimal solution. The computation time increases with more supports as more finite elements are constrained. Additionally, the simulations required almost 16 GB of RAM, which is the maximum available RAM on the laptop. Thus, a decrease in element size or a larger facade panel would require more RAM and, therefore, another device.

Finally, both problems are symmetrical; therefore, one would expect a symmetrical core geometry. In the case of two and four hinge supports simulations, the core structure does show symmetry. This can also be seen in the deformation and normal and shear stress patterns. Subsequently, in the case of two supports, a large increase in deflection and stress is visible when the core volume is decreased from 30 to 20 %. This is not visible in the case of four supports, which means, and is expected, that the core volume of a four-sided supported plate can be smaller compared to a two-sided supported panel before large effects occur from deformations and stresses. Additionally, in both cases, lines appear in the core that connect both outside layers near the supports, while further in the middle of the panel or on the yield line, only cylinders connect both outside layers.

	100% core volume	40% core volume	30% core volume	20% core volume
Max. Deflection $[mm]$	0.513	0.615	0.678	0.80
Deflection [relative]	1.00	1.20	1.32	1.54
Abs. max. normal stress $[N/mm^2]$	5.95	6.47	7.16	11.7
Normal stress [relative]	1.00	1.09	1.20	1.97
Abs. max. shear stress $[N/mm^2]$	0.543	1.59	2.09	2.28
Shear stress [relative]	1.00	2.93	3.85	4.20
Iterations required		28	26	47
Computation time [minutes]		36	37	69

Table 19: Overview results of simulations with two hinge supports.

	100% core volume	40% core volume	30% core volume	20% core volume
Max. Deflection $[mm]$	0.00489	0.00657	0.00780	0.00924
Deflection [relative]	1.00	1.34	1.60	1.89
Abs. max. normal stress $[N/mm^2]$	0.690	0.697	0.754	0.928
Normal stress [relative]	1.00	1.01	1.09	1.34
Abs. max. shear stress $[N/mm^2]$	0.119	0.225	0.241	0.316
Shear stress [relative]	1.00	1.89	2.03	2.66
Iterations required		42	38	50
Computation time [minutes]		75	73	104

Table 20: Overview results of simulations with four hinge supports.

9 Multi-criteria optimisation models

An optimisation model tries to find the optimal or near-optimal solution for complex problems that have a decision-making component. To explain the theory behind the optimisation model and the multiple options available in the domain, a short summary is provided, which first presents relatively easy problems and continues to more and more complex problems.

The creation of an optimisation model can be divided into four parts. The first part is to define the problem that has to be solved, for example, deciding which car to buy. The second part is to translate a real-life problem into a mathematical model. This step requires the creator to make assumptions and simplifications, which determine the complexity of the optimisation model. This step presents, for example, the variables, objective function, and constraints. The third step is the choice of an optimisation method, for example, using a genetic algorithm. At last, the fourth step presents the solution or solutions of the model, which can be presented in different ways, as will be explained in the subsequent paragraphs [28]. It is important to interpret the results within the context of the method and model used.

9.1 Linear and nonlinear models

The relatively simple method would be linear programming, which uses a linear objective function, and linear constraints, and continuous variables. This limits the complexity, but could provide the user with some first insights into the problem, and a simple method requires less computation time. Subsequently, quadratic programming can be the desired method, which uses a quadratic objective function and linear constraints. This increases the complexity and possibly bears a better resemblance to the real situation. Both LP and QP are convex problems, which implies that every local minimum is a global minimum. If the feasible region is bounded and convex, there is always a solution, and the solution can be multiple. This is an important advantage of LP and QP. To search for an optimal solution, a gradient-based method can be used. This iterative method uses the derivative of the objective and constraint functions to find the optimal solution [26].

The next method discussed is mixed integer linear programming, or MILP, which has the capability to use integer variables alongside continuous variables. This has the advantage of using variables, such as a binary option or variables that cannot be divided. The addition of integer variables requires a search method that is capable of accounting for them. Additionally, non-linear programming, or NLP, can be used to describe more complex problems or better depict the real problem in the mathematical model. This means that both the objective function and constraints can be nonlinear or linear. However, a significant difference from the previously discussed methods is that NLPs are usually non-convex problems. This can steer the solution to a local minimum or saddle point, so it is less efficient to find a global optimum compared to LP and QP. A possible search method could be gradient-based methods or non-gradient based methods. Non-gradient-based methods do not use derivatives. In order to know if a solution is optimal, the Karush Kuhn Tucker, or KKT, conditions are used [26].

The next logical step is to incorporate multiple objectives into the model because, in real life, normally more than one objective is important for the optimisation, especially conflicting objectives. For example, maximising profits while minimising emissions in case of sustainable business. The model can have continuous as well as discrete variables, linear or nonlinear objectives, and possible linear or nonlinear constraints. The challenge with multiple opposing objectives is that ample solutions exist, dependent on the relative importance of each objective. Thus, the user has to incorporate their preference when making a decision. The preference inclusion is divided into two parts: the a priori methods and the a posteriori methods. The a priori approach include the preference within one objective function; in other words, the preference has to be defined before the solution is known. In contrast, the a posteriori approach maps all solutions, and the user can make a decision afterwards, for example, with a Pareto front [28].

9.2 A priori approaches

The a priori approaches are abundant and can be distinguished into five classes, which are based on weight, related to defined goals, reference vectors, preference relations, or utility functions. All of these approaches are correct, and the choice between them depends on the reality the mathematical model tries to represent [28].

A well-known approach is the weighted objective function. The weights resemble the relative importance of each objective to the other objectives. The formula is shown in Equation (34), where w represents the weight and $\sum_{i=1}^{n} w_i = 1$, f is an objective, x is the variable, and n is the number of objectives. Furthermore, it is important to use normalisation of the objectives because, for example, objective one has a value between 10 and 100 and objective two has a value between 1 and 10. This implicitly means objective one determines the

solution, while this may not be desired. Additionally, reference vectors can be used, which are the mathematical equivalent of weight but give the preferential direction of the objective.

$$Objective = min_x \sum_{i=1}^{n} w_i f_i(x) \tag{34}$$

Subsequently, goal attainment can be used, which means minimising the largest difference between the value of each objective and the goal value of each objective. This is represented in Equation (35), where $f*$ represents the goal of the objective. The method is also known as the min-max method.

$$min_x max_i \frac{f_i(x) - f_i(x)*}{w_i}, i = 1..n \tag{35}$$

Next, preference relations can be used, which use preference functions that sort objectives based on the specified preference to translate objectives. Finally, utility functions rank solutions instead of objectives [28].

9.3 A posteriori approaches

The a posteriori approaches display all solutions with the use of a weighted objective function to calculate the different solutions that arise with all possible weight distributions. These solutions can be plotted on the Pareto front, and then the user can make a decision based on them. The Pareto front consists of optimal solutions originating from different weights. Especially in the case of two objectives, a 2D graph displays the relationship between the different weights and the optimal solution related to them. However, as the number of objectives increases, it becomes increasingly difficult to easily interpret the results. Another disadvantage is the required computation time to compute all possible solutions [28].

9.4 (Meta)Heuristics models

In order to solve the most complex problem, such as a large number of variables, non-trivial objective functions, complex constraints, or computation time is limited, (meta)heuristics are used. Heuristics are methods that, in a reasonable time, find a near-optimal solution, and meta-heuristics are "a set of intelligent strategies to enhance the efficiency of heuristic procedures" [18].

The (meta)heuristics methods can be classified in different manners. First, exploration versus exploitation. Exploration means the process of exploring unknown areas by expanding the search area. While exploitation is focused on areas containing high-quality solutions to have optimal convergence. This division is also called global versus local searching. Another classification is single- versus population-based methods. A single-based method uses one solution at a given time and steers this solution to find the near-optimal solution. In contrast, a population-based method uses multiple solutions simultaneously to iterate to a near-optimal solution. Furthermore, the methods can be divided into memory-intensive and memory-less methods. For example, storing the explored solutions or not, and iterative versus greedy. Iterative methods compute a complete solution in each iteration, while greedy methods begin with an empty solution and build the solution through the iterations. Finally, deterministic versus stochastic methods. A deterministic method implies that the same result will be obtained if the same initial solution is used. The stochastic method will give different results after each simulation while the initial solution remains equal [22]. Thus, through the use of the above-mentioned classifications, the different (meta)heuristic methods can be classified and compared. The choice between the methods depends on multiple preferences, for example, computation time, memory usage, and the accuracy of the solution.

9.4.1 Genetic Algorithm

The genetic algorithm is a (meta)heuristic model based on the theory of evolution developed by Charles Darwin. The algorithm starts with an initial population in the initial generation, which is a randomly generated population. A population is a specified number of individuals, where each individual is a possible solution to the problem. Thus, a combination of possible values for each variable. Additionally, within a generation, the different individuals are ranked based on the fitness score, which represents the objectives. This ranking is used to generate the subsequent generation. The creation is done through transferring the fittest individuals, cross-over between fit parents, and randomly generated new individuals. Parents are the individuals of a previous generation. The user can specify the percentage each of these three groups represents in the new generation. The central idea is that each generation results in more optimised individuals, therefore finding the near-optimal

solution. The process can be terminated in case a certain fitness score is acquired, the fitness score is not improved during the last number of generations, or after a maximum number of iterations. This algorithm, shown in Figure 62, is classified as an explorative, memory-less, iterative, and population-based method [58].

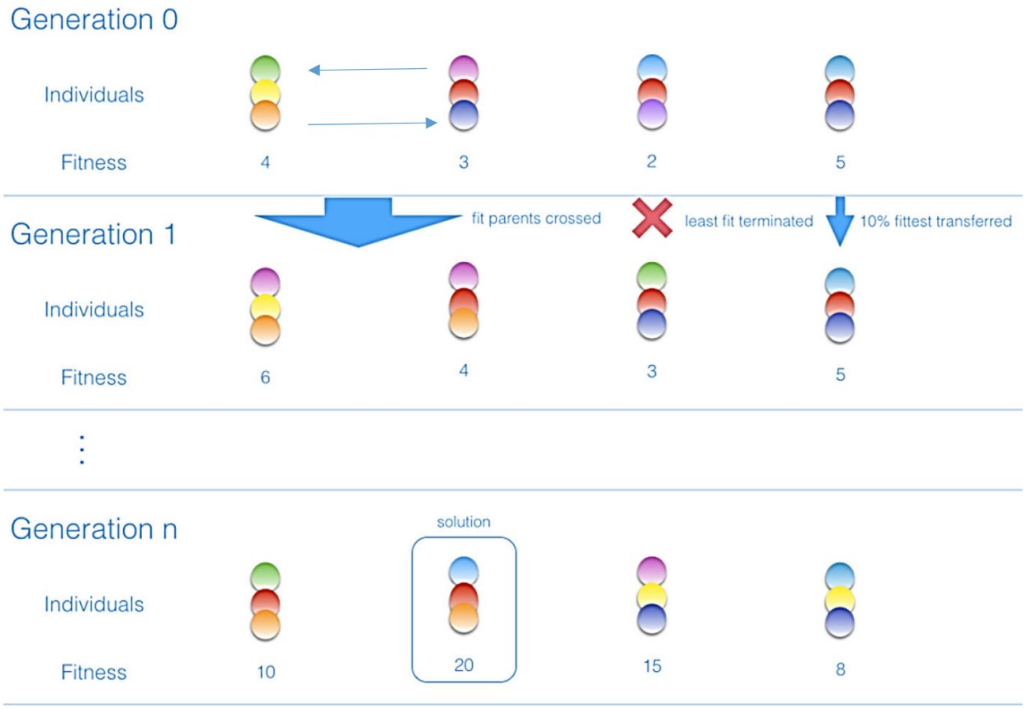


Figure 62: Principle of genetic algorithm [58].

9.5 Optimisation model

The previous paragraphs present different possible models to find the optimal solution. The choice between these models depends on the multiple factors that are summarised in the table below. The distinction has been made through the different components involved in an optimisation model, and each component influences which models can be used or are desired to be used.

Design variables	Continuous versus discrete
Objective function	linear or non-linear, single- or multi-objective
Constraints	linear or non-linear, equality versus inequality
Optimisation model	exact models such as LP, QP, NLP, and MILP or inexact models for example (meta)heuristics
Solution	Feasibility, optimality, and accuracy

Table 21: Classifications of optimisation models

Furthermore, if a (meta)heuristic model is used, a choice has to be made between all the different algorithms. New algorithms are always under construction. The distinction can be made between using state-of-the-art optimisation methods or commercially available optimisation methods. The state-of-the-art methods have the advantage of finding a method that is tailor-made for the defined problem. However, it required good programming skills and could require extensive searching through the existing literature. The other option is the use of available methods, which are implemented in software or programming languages. The advantage is the usability and availability of the methods. Although the number of methods is limited, and the available options may not be optimal for the specific problem.

To conclude, the choice has been made to adapt the genetic algorithm as the optimisation method because of the following reasons: the algorithm is widely available and implemented in commercial software, such as Rhino. The implementation in Rhino is the main reason, because the members of the committee have experience with Rhino, which makes it easy to implement. Additionally, parts of the total optimisation model have already been created by some of the members of the committee and integrated with the additive manufacturing capabilities. Also, the variables are discrete, and the problem is non-linear; therefore, a heuristic model has to be used.

10 Multi-criteria optimisation model

The previous chapter discusses the optimal geometries of the core under different loading and supporting conditions. However, in regard to the goal of creating a multi-criteria model within a certain time period, the choice has been made to focus on a pre-determined geometry as the core with some variables. The chosen core geometry is a honeycomb pattern because of its efficient shape in shear dominant situations [65, 74], which corresponds with the main situation of the core of a sandwich panel where the outside layers are predominately loaded in bending. Furthermore, the homogeneousness of an additive manufactured layer is an important part of its structural behaviour and overall strength [61]. Consequently, the main constraint of the core design is the possibility of printing the complete layer in a continuous process. Additionally, the created model optimises the core geometry of a facade panel in a certain application; therefore, the overall dimensions, such as width, height, and thickness, are presumed to be known upfront. Thus, these dimensions are not included as variables, and all variables are based solely on the core geometry. After establishing the parameters and overall core design, the subsequent part of this chapter is dedicated to explaining, in the following order, the variables, constraints, and objectives.

10.1 Parameters

For the sake of convenience, an exhaustive list of all the input parameters, variables, constraints, and objectives is presented in Tables 22 and 23. This overview shows the symbols, which are used throughout the subsequent chapter; the classification, which is the function of the parameter in the model; and a short definition with the accompanying unit.

Symbol	Classification	Definition
W	Input parameter (geometry)	The width of the facade panel $[mm]$
H	Input parameter (geometry)	The height of the facade panel $[mm]$
T_c	Input parameter (geometry)	The thickness of the facade panel core $[mm]$
T_g	Input parameter (geometry)	The thickness of the glass panel $[mm]$
L_w	Input parameter (geometry)	The width of the lines in the facade panel core $[mm]$
L_h	Input parameter (geometry)	The individual layer height core $[mm]$
R_{panel}	Input parameter (geometry)	The rotation of the honeycomb pattern [-]
λ_{air}	Input parameter (insulation)	The thermal conductivity of air $[W/mK]$
λ_{core}	Input parameter (insulation)	The thermal conductivity of the core material $[W/mK]$
λ_{glass}	Input parameter (insulation)	The thermal conductivity of glass $[W/mK]$
α_{in}	Input parameter (insulation)	Heat transmission coefficient inside $[W/m^2K]$
α_{out}	Input parameter (insulation)	Heat transmission coefficient outside $[W/m^2K]$
W_T	Input parameter (optimisation)	The weight of the transparency objective [-]
W_I	Input parameter (optimisation)	The weight of the insulation objective [-]
W_D	Input parameter (optimisation)	The weight of the daylight objective [-]
$Load$	Input parameter (structural)	The surface load applied the top glass layer $[kN/m^2]$
$TB_{support}$	Input parameter (structural)	The presence or absence of top and bottom supports [-]
$LR_{support}$	Input parameter (structural)	The presence or absence of left and right supports [-]
$Young_{glass}$	Input parameter (structural)	The Young's Modulus of glass $[kN/cm^2]$
$Shear_{glass}$	Input parameter (structural)	The Shear modulus of glass $[kN/cm^2]$
$Weight_{glass}$	Input parameter (structural)	The specific weight of glass $[kN/m^3]$
Ts_{glass}	Input parameter (structural)	The tensile strength of glass $[kN/cm^2]$
Cs_{glass}	Input parameter (structural)	The compressive strength of glass $[kN/cm^2]$
$Young_{core}$	Input parameter (structural)	The Young's Modulus of the core $[kN/cm^2]$
$Shear_{core}$	Input parameter (structural)	The Shear modulus of the core $[kN/cm^2]$
$Weight_{core}$	Input parameter (structural)	The specific weight of the core $[kN/m^3]$
Ts_{core}	Input parameter (structural)	The tensile strength of the core $[kN/cm^2]$
Cs_{core}	Input parameter (structural)	The compressive strength of the core $[kN/cm^2]$
Or_{facade}	Input parameter (daylight)	The orientation of the facade [-]

Table 22: Overview of all the models' input parameters.

Symbol	Classification	Definition
N_{hh}	Variable	The number of hexagons along the height [-]
N_{hw}	Variable	The number of half hexagons along the width [-]
$Indiv_H$	Variable	The change in individual height hexagons [-]
P_H	Variable	The pattern of the individual height modification [-]
A_c	Variable	The angle of the core lines relative to the glass layer [°]
RP_{incl}	Variable	The location of the inclination reference point [-]
$N_{h_{min}}$	Constraint	The minimum number of hexagons along the height [-]
$N_{h_{max}}$	Constraint	The maximum number of hexagons along the height [-]
$N_{w_{min}}$	Constraint	The minimum number of hexagons along the width [-]
$N_{w_{max}}$	Constraint	The maximum number of hexagons along the width [-]
$Indiv_{H_{min}}$	Constraint	The minimum change in individual height hexagons [-]
$Indiv_{H_{max}}$	Constraint	The maximum change in individual height hexagons [-]
$A_{c_{min}}$	Constraint	The minimum angle of the core lines [°]
$A_{c_{max}}$	Constraint	The maximum angle of the core lines [°]
$RP_{incl_{min}}$	Constraint	The minimum value for RP_{incl} [-]
$RP_{incl_{max}}$	Constraint	The maximum value of RP_{incl} [-]
S_{min}	Constraint	The fulfilment of the structural limit states [-]
O_I	Objective	The insulation objective [-]
O_T	Objective	The transparency objective [-]
O_D	Objective	The daylight objective [-]
F_{final}	Fitness function	The optimisation function comprised of the normalised objectives and S_{min} [-]

Table 23: Overview of all the models' variables, constraints, and objectives.

10.2 Variables and Constraints

The variables are a combination of possible changes within a honeycomb pattern, while the ability to print a layer continuously remains. The variables can be grouped into changing the pattern in 2D and changing it in 3D. The first group consists of four variables, and the latter group consists of two.

First, the variables that transform the core structure in 3D are discussed. Each line segment is rotated by a maximum of A_c around a reference point defined by RP_{incl}. The $A_{c_{min}}$ is equal to zero degrees, which means no inclination, and the $A_{c_{max}}$ is 45 degrees. The upper limit is the maximum feasible inclination without the need for temporary supports during the printing process. The distance between the line segment and the reference point along one axis determines the rotation in the vertical direction, where the angle of the line segment with the maximum distance equals to A_c. Moreover, this angle lies between zero and A_c degrees. The location of the reference point can be at the bottom, top, or in the middle of the facade. This is represented as a single value, where zero represents the location at the bottom, half is the location in the middle of the facade, and one is the location at the top of the facade. Thus, $RP_{incl_{min}}$ has to be larger than or equal to zero, and $RP_{incl_{max}}$ has to be smaller than or equal to one. The variables are shown in Equations (36) and (37).

One of the variables that alters the geometry in 2D is N_{hw}, which is the number of half hexagons along the width. Half hexagons, instead of complete hexagons, are chosen because it allows the pattern to be asymmetrical if N_{wh} is uneven. Additionally, the upper and lower limits are defined as $N_{w_{min}}$ and $N_{w_{max}}$. These boundaries are chosen freely by the user; however, it is advised to use at least a lower limit of three, and the upper limit boundary should consider the geometry of the panel. For example, in theory, a specific N_{wh} value can result in a hexagon that has no hole in the middle and becomes solid. This exact value depends mainly on L_w, W, and H. Thus, $N_{w_{max}}$ should be chosen with caution. Subsequently, the variable N_{hh}, which is the number of hexagons along the height, modifies the geometry as well. The upper and lower limits are described by $N_{h_{min}}$ and $N_{h_{max}}$. The advice or consideration in the decision-making process of picking the boundary values is the same as the previously discussed variable. The variables are depicted in Equations (38) and (39).

The next variable, $Indiv_H$, is related to the individual height of the straight parts of the hexagons, the lines with twice the thickness, highlighted in Figure 63. The double thickness is the result of the constraint that one layer should be printed continuously. This figure also shows the variables N_{wh} and N_{hh}, and the input parameter L_w. The boundaries $Indiv_{H_{max}}$ and $Indiv_{H_{min}}$ must at least be between one and minus one. The inclusion of positive and negative values is introduced to affect the effect of P_H, introduced in the next paragraph. As an illustration, if $Indiv_H$ is 0.5, it will result in line length alterations between roughly 50 and 150 % of their original length.

The length adjustment domain is defined by $Indiv_H$, while the change of each individual line is determined

through the variable P_H. This variable is chosen to be a sine function with three different domains and is influenced by the change of the $Indiv_H$ sign, which flips the function. The patterns are shown in Figure 64, which are sine functions with a domain of zero to π, zero to 2π, and zero to 3π. The P_H variable is equal to 1, 2, or 3, which represents the before-mentioned domains. The black line in Figure 64, represents the center line of the vertical extremes. The patterns are chosen for their symmetrical, asymmetrical, and uncomplicated shapes. The horizontal axis in the graph can be seen as the direction of the facade in which these individual lines lie. For example, an individual line in the center of the facade lies in the middle of the black horizontal line. Next, the vertical axis represents the change in line length, where a point above the black line has an increased line length and a point below the black line has a decreased line length. The top of the graph, on the vertical axis, depicts the area of the facade where the line length is increased by approximately the absolute value of $Indiv_H$, and the bottom represents the area with a line length reduction of more or less the absolute value of $Indiv_H$.

The exact adjustment of line length cannot be simply determined upfront while fulfilling the constraint to keep an unvarying facade height or invariable sum of these line lengths. This can be caused by either one of the following two reasons: first, the initial lines sometimes have a different length, as the lines at the borders are smaller than those in the center of the facade. Second, the integral of the blue or orange line around the black line can give a nonzero value. This complicates the modification process. In order to consistently meet the constraint regarding the total length, each line is multiplied by an identical number, somewhere around one. As the extent of the influence of this number on the length modification is ambiguous, it is admonished to conservatively limit the domain of $Indiv_H$ to ± 0.6. The variables are displayed in Equations (40) and (41).

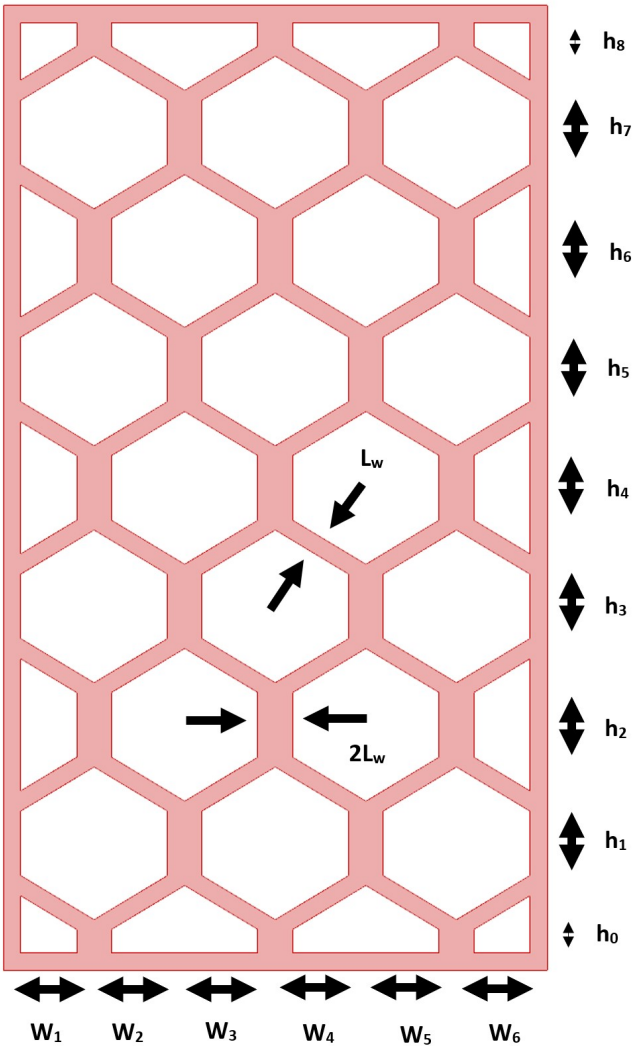


Figure 63: The geometrical features highlighted, where w represents the half hexagon widths and h the individual lines influences by $Indiv_H$ and P_H.

$$A_{c_{min}} \leq A_c \leq A_{c_{max}}, \quad \text{where } 0 \leq A_{c_{min}} < A_{c_{max}} \leq 45 \tag{36}$$

$$RP_{incl_{min}} \leq RP_{incl} \leq RP_{incl_{max}}, \quad \text{where } 0 \leq RP_{incl_{min}} < RP_{incl_{max}} \leq 1 \tag{37}$$

$$N_{h_{min}} \leq N_{hh} \leq N_{h_{max}}, \quad \text{where } N_{h_{min}} \geq 3 \tag{38}$$

$$N_{w_{min}} \leq N_{hw} \leq N_{w_{max}}, \quad \text{where } N_{w_{min}} \geq 3 \tag{39}$$

$$Indiv_{H_{min}} \leq Indiv_H \leq Indiv_{H_{max}}, \quad \text{where } -0.6 \leq Indiv_{H_{min}} < Indiv_{H_{max}} \leq 0.6 \tag{40}$$

$$P_H = 1, 2, 3 \tag{41}$$

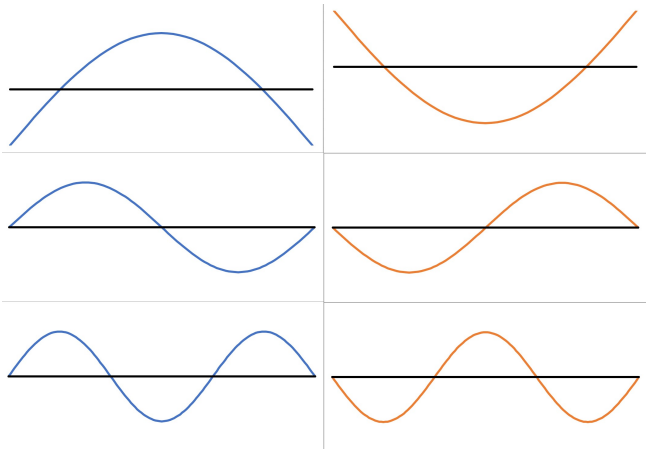


Figure 64: Possible length-altering patterns, where a positive or negative $Indiv_H$ results in the blue or orange lines, respectively.

10.3 Objectives and optimisation

The model uses multiple objectives, which are transparency, insulation, and daylight control. The insulation value is calculated through Equation (23). Next, the transparency value originates from the ability to see through the facade panel from a perpendicular viewpoint. The daylight control value stems from the ratio between the total incoming sunlight, direct and reflected, at the winter solstice and the proportion of direct sunlight and the total sunlight at the summer solstice. This number increases when a geometry is used that allows for ample total sunlight during the winter and minimises the direct sunlight during the summer. This is beneficial in temperate climates as heat transportation during the summer is limited, which reduces the usage of air conditioning while allowing for sufficient light, which is advantageous to indoor living conditions.

Additionally, the used optimisation algorithm is the genetic algorithm because the variables are discrete and the problem is highly nonlinear. Within the genetic algorithm, multiple options can be used, for example, the population, percentage of crossover in the new generation, percentage of maintaining the fittest individuals from the last generation to the new generation, maximum stagnant generations, and a possible population boost in the first generation. These options affect the efficiency of the algorithm to discover the optimal solution. The optimal values for these options depend on the problem and can be found through trial and error.

The final fitness function determines the most optimal geometry and is given in the expression below, Equation (42). The fraction of each objective results in a value between zero and 1 in every situation. This guarantees the correct functionality of the weights. Thus, the optimal solution has the maximum fitness score. Moreover, the facade panel should be structurally safe; therefore, a constraint S_{min} is added to the fitness function with a penalisation term. The equation of S_{min} is shown in Equation (43. If the actual deflection is smaller than the maximum allowed deflection and the stress remains below the stress limit, the function is equal to zero. If either or both of them do not fulfill the above-mentioned condition, the function has a negative value. This negative value is multiplied by the penalisation term to prevent designs that do not comply with the structural requirement from becoming an optimal solution.

$$F_{final} = W_T \cdot \frac{O_T - min(O_T)}{max(O_T) - min(O_T)} + W_I \cdot \frac{O_I - min(O_I)}{max(O_I) - min(O_I)} + W_D \cdot \frac{O_D - min(O_D)}{max(O_D) - min(O_D)} + 100,000 \cdot S_{min} \tag{42}$$

$$S_{min} = min(Max_{deflection} - Deflection, 0) + min(1 - UnityCheck_{stress}, 0) \tag{43}$$

11 Explanation Rhino optimisation model

This chapter explains the fundamentals of the Rhino model. It provides a general understanding of the different components and steps used in the optimisation model. Nevertheless, a helicopter view is applied, but in Appendix H, the model components are described in detail.

In general, the steps made in the Rhino model can be divided into three groups: the required input, which contains the variables as well; the creation of the panel, which is mainly comprised of creating the core; and the optimisation process, including the optimisation criteria and constraints. The difference between variables and the other input is their relationship with the optimisation process; while the variables are functioning as the components used to optimise the geometry, the other inputs keep their initial value. Thus, in theory, all inputs can be chosen as variables; however, as this is not the case in the current selection of variables, a distinction is made. This basic division is displayed in Figure 65. Moreover, this division is used to further explain these three parts. First, the required inputs are described, followed by separating the creation of the geometry into multiple steps, and at last, the different parts of the optimisation process are explained.

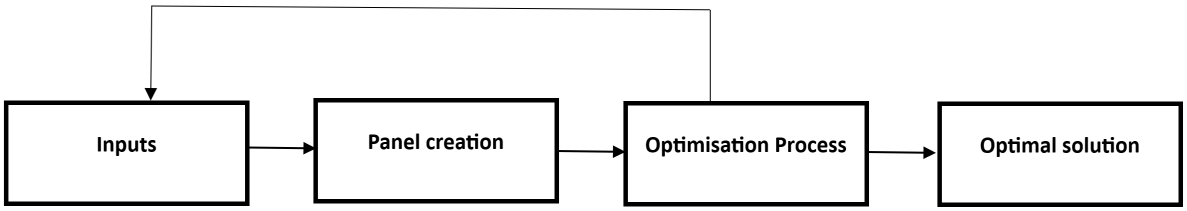


Figure 65: Basic components of the Rhino model.

11.1 Defining the required inputs

The inputs can be distinguished into three parts, see Figure 67. The first part is the variables, which are the chosen inputs by the user that function both as the required input to create the geometry, to define the optimisation criteria or constraints, and as variables in the optimisation process. For example, as discussed in the previous chapter, all the variables defined are required inputs to generate the geometry, such as the number of hexagons vertically and horizontally. Additionally, a variable can be comprised of different core materials, and the optimisation process will discover the most optimal material for a specific situation. Thus, a variable can be any of the required inputs. Second, the other inputs are grouped into geometrical inputs and optimisation inputs. The geometrical input consists of, for example, the dimensions of the panel and basic parameters defining the design of the core geometry. The optimisation input is related to all the applied optimisation criteria and constraints, as well as the chosen optimisation procedure. In this case, the inputs for the optimisation procedure are the weights of all the criteria. Another example are the insulation properties required for the insulation criterion. An overview of all these inputs is given in the previous chapter.

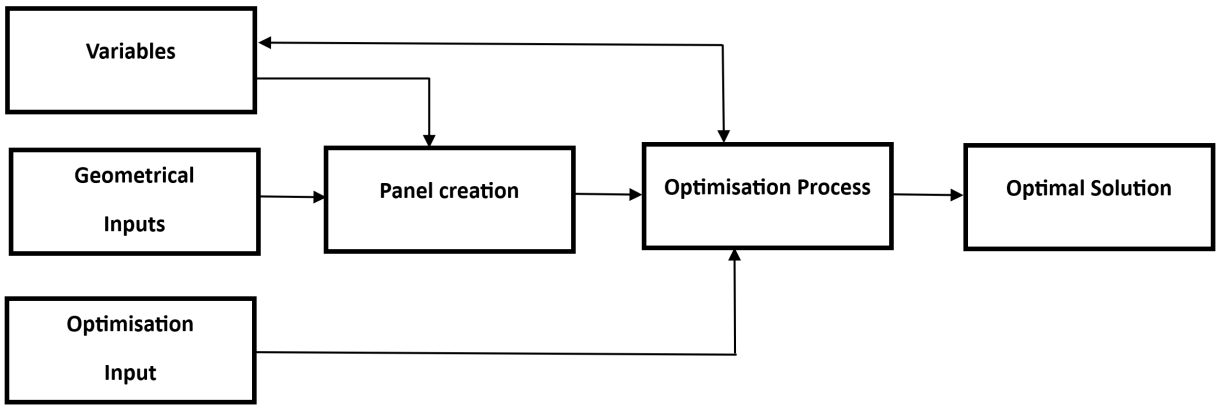


Figure 66: Basic input sub-components of the Rhino model.

11.2 Panel creation

The creation of the panel is almost completely dedicated to the core geometry, and the glass panels can be easily generated. Therefore, all the discussed steps, shown in Figure 68, are only relevant to the core geometry. The first step is to create a 2D honeycomb pattern with the predefined inputs, which is performed by one of the existing applications in Grasshopper. An example of the gathered end result is displayed in Figure 68a. Subsequently, the line thickness has to be added to represent the line thickness used during the printing process of the core. Moreover, due to the choice of being able to print each layer in a continuous manner, some of the lines have twice the thickness. The thickness is added by dividing the honeycomb pattern into straight lines. For example, each hexagon consists of six straight lines. Then, the vertical lines, except for the boundaries, are duplicated, and a small movement in the opposite direction is imposed to resemble the line thickness. The obtained result is displayed in Figure 68b.

The third step is altering the hexagon sizes, which is related to the variables $Indiv_H$ and P_H. The exact process of alteration is explained in the previous chapter. Within Grasshopper, all vertical lines are grouped based on their vertical position in the pattern; thereafter, based on the two variables, the lengths of these lines are changed, and all the lines are reconnected to create a new honeycomb pattern. This result is shown in Figure 68c. Also, the change in hexagon size can be seen by comparing it with the pattern in Figure 68b.

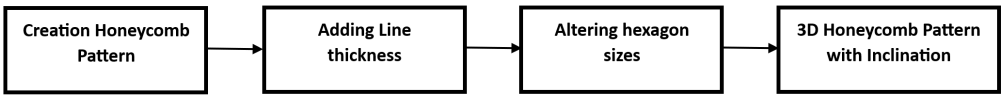
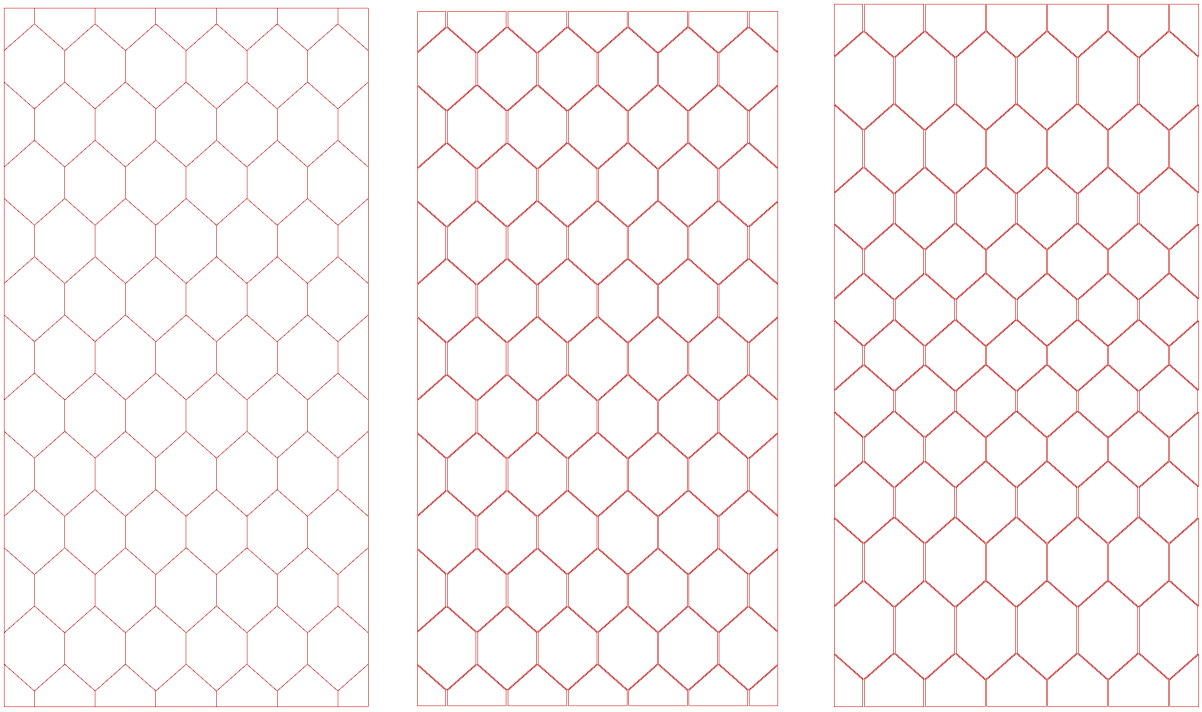


Figure 67: Basic panel creation sub-components of the Rhino model.



(a) Creation honeycomb pattern (b) Adding Line thickness (c) Altering hexagon sizes

Figure 68: An visual representation of the first three steps in the panel creation.

The final step in creating the core geometry is to turn a 2D pattern into a 3D pattern with possible inclination. This step uses the variables RP_{incl} and A_c to define the magnitude of the inclination and the pattern of inclination. The applied procedure, in general, first determines the change in location of each individual line as a result of the imposed inclination along the thickness. The second step is to create a 3D pattern made of lines, while some small changes are made to the line lengths to ensure all lines are connected. Then, based on the lines, a 3D representation is created, which can be seen in Figure 69.

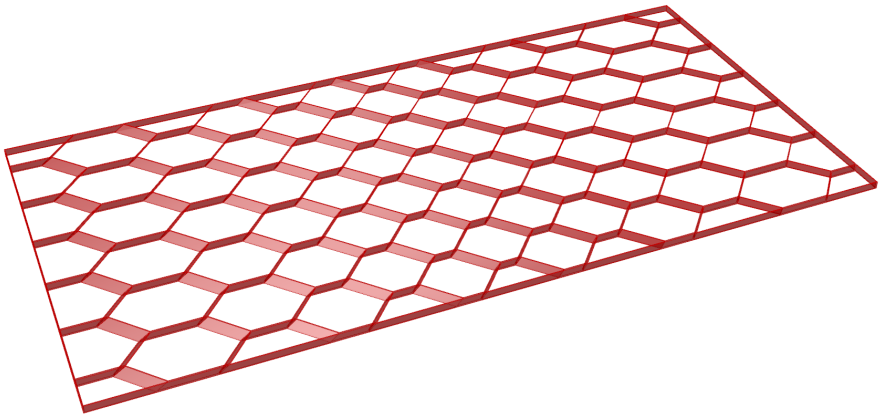


Figure 69: The visual representation of the end result of the panel creation

11.3 Optimisation Process

The final part is comprised of all the relevant components of the optimisation process, as shown in Figure 70. First, the criteria are defined, and the value of their objective is determined. The complexity of defining criteria depends on the chosen criteria. For example, the insulation and transparency criteria are easily calculated. The transparency criterion is computed by simply obtaining the area of the panel, which can be seen through in case someone stands directly in front of it. Next, the insulation criterion is based on the formula defined in Chapter 4, which can easily be calculated with the defined inputs and created geometry. However, the daylight criterion requires additional steps. First, the daylight scenario has to be created, which means that the correct secondary intercardinal direction from the inputs is used. Furthermore, the area that captures the incoming sunlight has to be created. This is set to be just behind the panel, and the small gap between the panel and area is introduced to allow reflection light to reach the areas that would otherwise be in the shadow of the core geometry. Then, the solar radiation is calculated with and without reflection. This requires the use of two different Ladybug components. Ladybug is a library within Grasshopper. Thereafter, the value for the daylight criterion is computed.

In addition to the different criteria, constraints have to be included. The structural constraints require a reconfiguration of the core geometry because the structural finite element model uses plate elements. Therefore, the midsections of all the lines within the core are needed. After completion, the structural model is defined with the Karamba3D library, which defines the boundary conditions, material properties, loading condition, mesh size, and calculates the maximum deflection and Unity Check. At last, the output from all criteria and the constraint is used to determine the fitness value, and a genetic algorithm application performs the optimisation process.

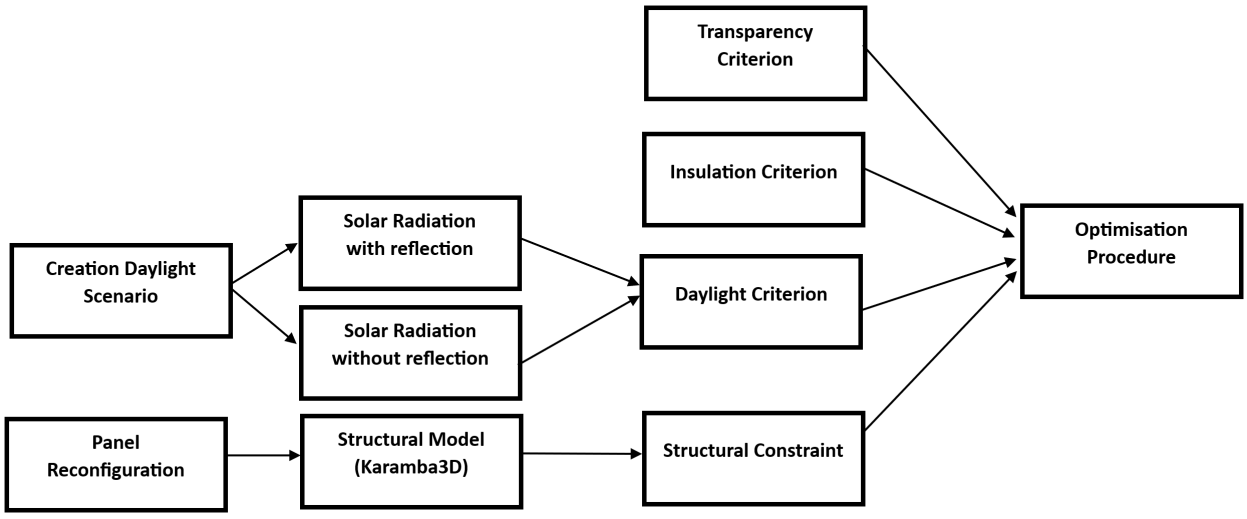


Figure 70: Basic optimisation sub-components of the Rhino model.

12 Validation Karamba3D

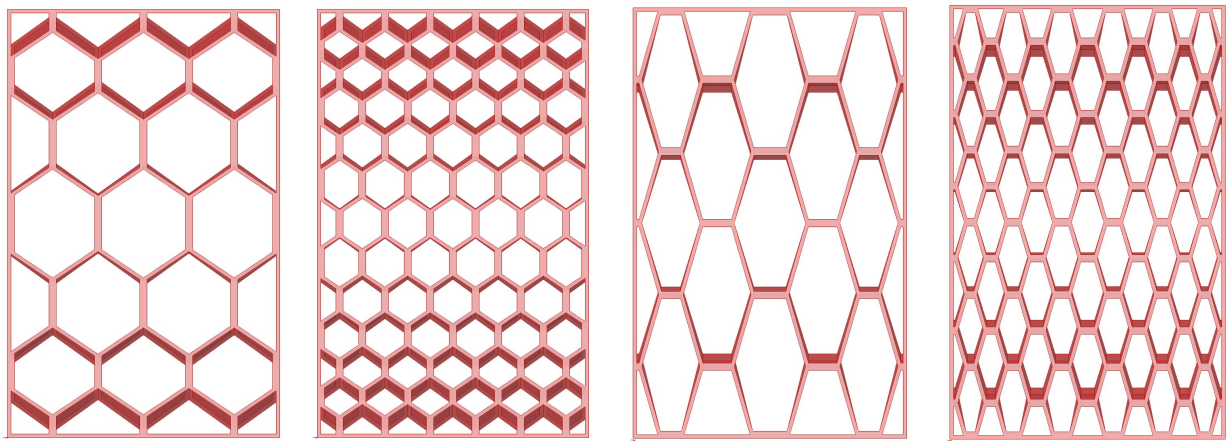
A validation of the structural model Karamba3D, used in the optimisation model, is compared with a FEA program called DIANA. Karamba3D is a parametric engineering tool designed to make quick analyses that are mainly used in the preliminary design stage. The available elements are shell and beam elements, and the optimisation model uses shell elements to represent glass sheets as well as the core geometry. As a result of this element representation, two DIANA models are used, namely a FEA with solid elements and one with shell elements. First, the core geometries used in the validation are discussed, and then the three different models are explained extensively. Thereafter, the results are discussed, and, finally, the models are compared and conclusions drawn.

12.1 Alternative core geometries used for validation

The validation uses four different core geometries. Two of them correspond with the geometries used in the physical experiments, and two additional geometries are used to understand the effect of increasing the number of hexagons over the width and height. The values for the variables per geometry are shown in Table 24, and displayed in Figure 71. All geometries have a height of 710 mm, a width of 450 mm, a thickness of 18 mm, and a line width of 6 mm. The glass sheets have the same width and height as the core and are each 1 mm thick.

	Pattern 1	Pattern 2	Pattern 3	Pattern 4
N_{hh} [-]	6	12	6	12
N_{vh} [-]	6	12	6	12
$Indiv_H$ [-]	0.7	0.5	0.7	0.5
P_H [-]	1	1	1	1
A_c [°]	40	40	40	40
RP_{incl} [-]	0.5	0.5	0.5	0.5
Rotated	No	No	Yes	Yes

Table 24: The characteristics of the different geometries with respect to the variables and orientation.

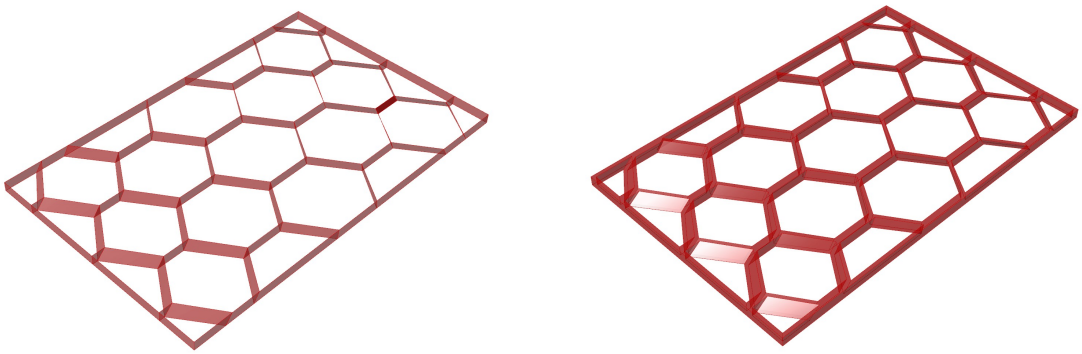


(a) Pattern 1 (b) Pattern 2 (c) Pattern 3 (d) Pattern 4

Figure 71: A front view of the four core geometries

12.2 Model characteristics

The model characteristics of all three models are discussed, such as the finite elements, supports, and loading conditions. The first model that will be discussed is the Karamba3D model, followed by the DIANA model with shell elements, and finally the DIANA model with solid elements. The basic geometries that are used in the models are shown in Figure 72. Pattern 1 is used as a visualisation to show the differences between the two representations. The geometry used for shell elements consists of the center lines of the solid model. However, around the left and right boundaries, the double line-width lines, the most outer center line is used. This is due to retaining a smooth rectangular line around the boundary. Thereby, this small deviation from the true center line will probably have an insignificant effect on the results.



(a) (b)

Figure 72: The basic geometry used in the models with shell elements (a) and solid elements (b).

12.2.1 Karamba3D

The model creation in Karamba3D requires several steps and these steps, are used as a guideline for the structure of this paragraph. Additionally, the application has its own website, which is used as the main reference [6]. The Karamba3D version is 2.2.0.15, and Von Mises stresses are used as this type of stress is the only one available in Karamba3D.

The first step is to create a mesh from the core geometry, which is displayed in Figure 72 (a). Multiple meshes are used to compare them based on maximum deflection, Unity-Check of the Von-Mises stresses, and computation time. The latter is an important aspect of the optimisation process. In Karamba3D, the mesh is defined by the spacing between the nodes, where the nodes at the boundary are used as the basis for the supports and are used to create nodes at the glass sheets with the same x and y coordinates in order to assure proper connection between the glass sheets and the core. The meshes are differentiated in their spacing in the width and height planes as well as in the number of nodes along the thickness or depth. The glass sheets have one element over the thickness and the same element size. The different models with their respective names, element sizes, and number of nodes along the thickness of the core are shown in Table 25, and two examples of the meshes are displayed in Figure 73.

Model name	A0	A1	A2	A3	A4	B0	B1	B2	B3	B4	C0	C1	C2	C3	C4
Element size [mm]	30	15	10	5	2	30	15	10	5	2	30	15	10	5	2
Elements along thickness core	5	5	5	5	5	7	7	7	7	7	9	9	9	9	9

Table 25: Applied meshes in the Karamba3D models

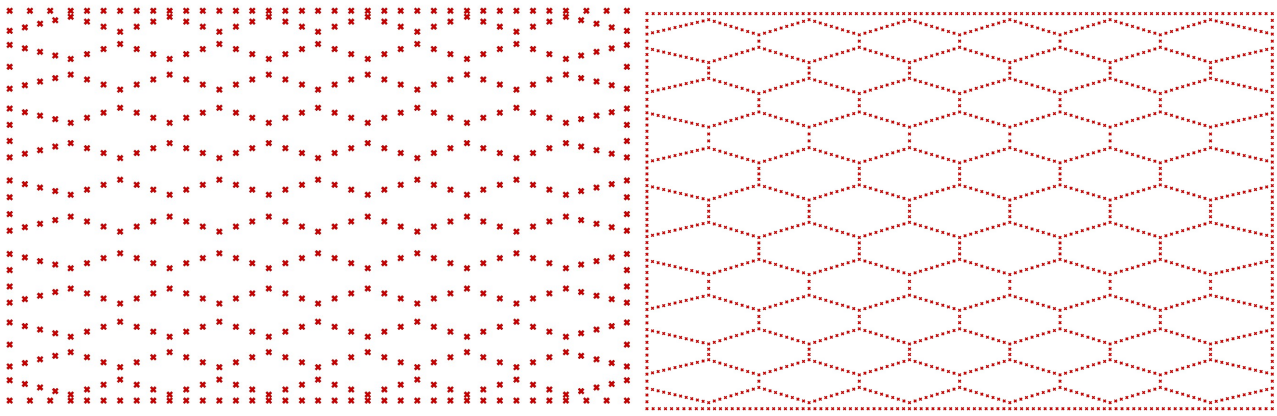


Figure 73: Visualisation of Pattern 4 with an element size of 30 and 10 millimetre.

The next step is to define the material properties of the two distinct materials, which are based on the physical experiments. Isotropic material is assumed for simplicity, and tensile and compressive strength are needed for the Unity-check, which requires the following input:

- Young's modulus of 70 and 2.02 of glass and the core respectively [GPa],
- In-plane shear modulus of 30 and 0.90 [GPa],
- Transverse shear modulus of 30 and 0.90 [GPa],
- Specific weight of 2480 and 1270 [kg/m^3],
- Tensile strength of 40 and 50 [MPa],
- Compressive strength of 400 and 50 [MPa].

After the establishment of the material properties, the cross sections can be defined. The cross section used is the constant shell cross section, where the height is equal to the line width, double line width, or glass sheet thickness. This shell type is chosen because of its constant thickness and material properties.

Subsequently, the shell elements are created with the mesh and cross section as their input. Karamba3D will automatically merge nodes if they are situated within the defined limit distance. Only one shell, no parallel shells, represents the vertical sides of the hexagons and glass sheets. In this model, the limit distance is chosen to be sufficiently small to prevent this process. The creation of shell elements is done separately for both glass layers and the core geometry. The shell elements resemble TRIC elements, which are flat elements and neglect transverse shear deformations [14, 15]. Also, an option is provided to include or exclude the bending behaviour, thus reassembling a shell or membrane element. In this model, bending behaviour is included.

Furthermore, a constant area load is applied to the top glass layer. The load is equal to 1.04945 kN/m^2, which in the entirety of the applied load is equal to a load of 1.00 kN/m^2 in the solid element model. This discrepancy emerged because of the use of center lines in the model, which uses shell elements; consequently, the area is slightly smaller. Finally, the area load is applied as point loads on the nodes, and the load due to the self-weight of the structure is ignored.

Moreover, the supports are situated at the nodes described previously. In this model, the shortest edges of the lower glass panel are supported. One of the supports prevents translation in the x, y, and z directions, while the other only inhibits translation in the x and z directions. Thus, it can be seen as a hinged support and a rolling hinged support.

After the creation of the loads, support, and shell elements, the model is assembled and analysed based on first-order theory for small deflections. Finally, the results provided are the maximum deflection and an utilisation factor for all elements. The utilisation factor is the ratio between the Von Mises stresses and the yield stress; for example, a value of 1 implies both stresses are equal. It is based on Eurocode 3, and buckling is not considered.

12.2.2 DIANA model with shell elements

DIANA version 10.5 is used [7]. The settings are set to a linear mesh order and quad elements. The core geometry is imported from Rhino with a step file. Next, the glass sheets are created in DIANA, which are polygon sheets.

Subsequently, shape properties are added, which are the element class, material properties, and geometry. The element class is flat shells or regular curved shells. The flat shells are, in essence, a combination of the plane stress and plate bending elements. The curved shell elements are defined by an 'isoparametric degeneratid-solid approach' using the hypotheses of 'straight-normals' and 'zero-normal-stress' [8]. The characteristics of the elements are shown in Figure 74. The use of regular curved shells is limited to a small number of polygon sheets in the core if the hexagons are rotated, as in patterns 3 and 4. The sheets that have an assigned element class of regular curved shells are the sheets that are part of a hexagon and connected to either the top or bottom of the core geometry, as displayed in Figure 71. Additionally, the material properties are added. Linear isotropic properties are used, which require the following input:

- Young's modulus of 70,000 and 2,020 of glass and the core respectively [N/mm^2],
- Poisson's ratio of 0.21 and 0.4 [-].

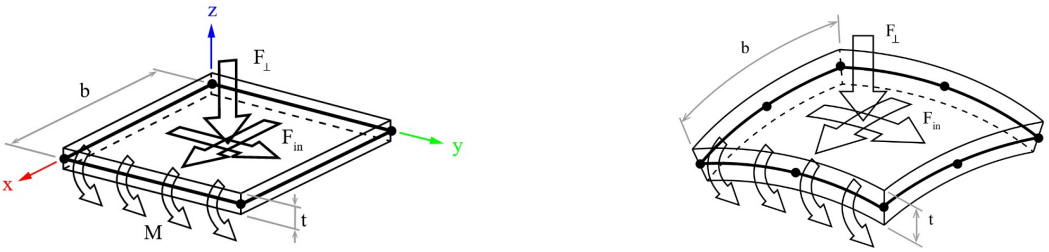


Figure 74: The characteristics of a flat shell and a regular curved shell element [8].

Then, the element geometry is defined, which constists solely of setting the thickness of the core and glass elements to 6 or 12 and 2 mm, respectively. In Figure 71, it is visible that the lines in the core have different thicknesses; hence, the sheets in the core are either 6 or 12 mm thick. Furthermore, the load is added as a distributed force along the face of the upper glass panel with a magnitude of 1.04945 kN/m^2. Then, the edge supports are added at the shortest edges of the bottom glass sheet, where one support fixes all three translations and the other one fixes two translations and allows translation in the direction of the longest edge. Finally, the mesh is generated with an element size of 6, 3, and 2 mm. A static linear analysis is performed, and an overview of the used finite elements is displayed in the table below.

Finite elements type (isoparametric)	Triangular flat/curved shell (T15SF / T15SH)	Quadrilateral flat/curved shell (Q20SF / Q20SH)
DOFs (per node)	u_x, u_y, u_z, ϕ_x, ϕ_y	u_x, u_y, u_z, ϕ_x, ϕ_y
Interpolation scheme	Linear	Linear
Integration scheme	1	2x2
Shape dimension	2D	2D
Topological dimension	2D	2D
Stress components	σ_{xx}, σ_{yy} σ_{xy}, σ_{xz}, σ_{yx} σ_{yz}, σ_{zx}, σ_{zy}	σ_{xx}, σ_{yy} σ_{xy}, σ_{xz}, σ_{yx} σ_{yz}, σ_{zx}, σ_{zy}
Shear deformation	Yes	Yes

Table 26: Overview of characteristics of shell elements in Diana.

12.2.3 DIANA model with solid elements

In the model with solid elements, the core geometry is also imported as a step file from Rhino, and the two glass sheets are generated as blocks. The default mesh order is linear, and the mesh type is hexahedrons. Moreover, the shape properties are added, which have similar material properties as the shells. However, the element class is equal to structural solids, and the geometry does not require any inputs. The characteristics of solid elements are shown in Figure 75. Then, the load is added as a distributed load along the face of the upper glass sheet with a value of 1.00 kN/m^2. Subsequently, the supports are created, which are defined as edge supports along the shortest edges of the lower glass sheet. One support fixes all translations, and the other only allows for translation in the direction of the longest edge. Finally, the mesh is generated, with three different mesh sizes and default options. The used mesh sizes are 6, 3, and 2 mm. An overview of the finite element is presented in the table below. Again, a static linear analysis is used.

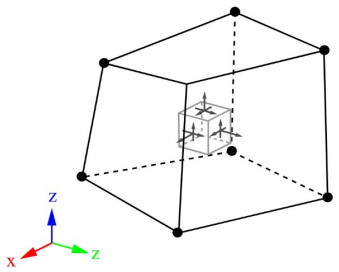


Figure 75: Characteristics of solid elements [8].

Finite Elements type (Isoparametric solid)	Tetrahedron (TE12L)	Pyramid (PY15L)	Wedge (TP18L	Brick (HX24L)
DOFs (per node)	12 u_x, u_y, u_z	15 u_x, u_y, u_z	18 u_x, u_y, u_z	24 u_x, u_y, u_z
Interpolation scheme	Linear	Linear	Linear	Linear
Integration scheme	1	5	1x2	2x2x2
Shape dimension	3D	3D	3D	3D
Topological dimension	3D	3D	3D	3D
Stress components	σ_{xx}, σ_{yy}, σ_{zz} σ_{xy}, σ_{xz}, σ_{yx} σ_{yz}, σ_{zx}, σ_{zy}	σ_{xx}, σ_{yy}, σ_{zz} σ_{xy}, σ_{xz}, σ_{yx} σ_{yz}, σ_{zx}, σ_{zy}	σ_{xx}, σ_{yy}, σ_{zz} σ_{xy}, σ_{xz}, σ_{yx} σ_{yz}, σ_{zx}, σ_{zy}	σ_{xx}, σ_{yy}, σ_{zz} σ_{xy}, σ_{xz}, σ_{yx} σ_{yz}, σ_{zx}, σ_{zy}
Shear deformation	Yes	Yes	Yes	Yes

Table 27: Overview of characteristics of solid elements in Diana.

12.3 Results

This section will discuss all the results from Karamba3D, Diana with shell elements, and Diana with solid elements. The results are discussed in the before-stated order. The results will provide an overview of the maximum deflections and the Unity check based on Von-Mises stresses. Additionally, the Karamba3D results also contain the computation time for each of the models, because in an optimisation model, the computation time is an important aspect.

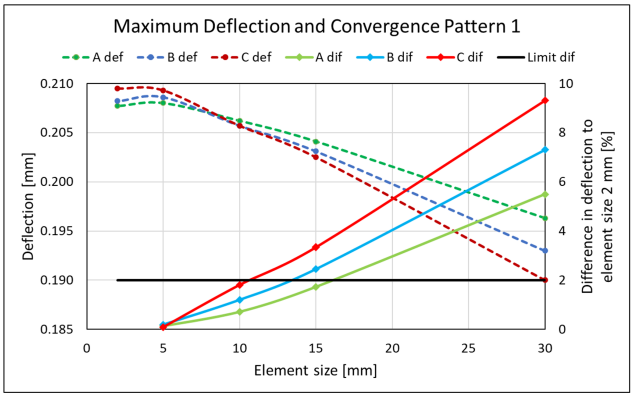
12.3.1 Results Karamba3D

The Karamba3D results, which show the maximum deflection, convergence behaviour in the deflection, and the Unity Check, are displayed in Figures 76, 77, 78, and 79. The graph displaying the maximum deflection has two vertical axes. The left vertical axis shows the maximum deflection of the different models used in Karamba3D, and the dotted lines belong to the left vertical axis. Next, the right vertical axis shows the difference in deflection, expressed in percentage, with respect to the deflection of the relative model using an element size of 2 mm. For example, in Pattern 1, Model A with elements of 2 mm has a maximum deflection of 0.2077 mm, and Model A with elements of 5 mm has a maximum deflection of 0.2080 mm. Thus, a difference of 0.0003 mm, which is 0.14 % of the maximum deflection of Model A with elements of 2 mm. This process is performed for all models in every pattern. The results are shown by the continuous lines, which belong to the right vertical axis. Moreover, a black line represents the arbitrary limit that equals to 2 %. If the model with a certain element size lies below the limit, it is concluded that the mesh is adequately refined for the purpose of the optimisation model.

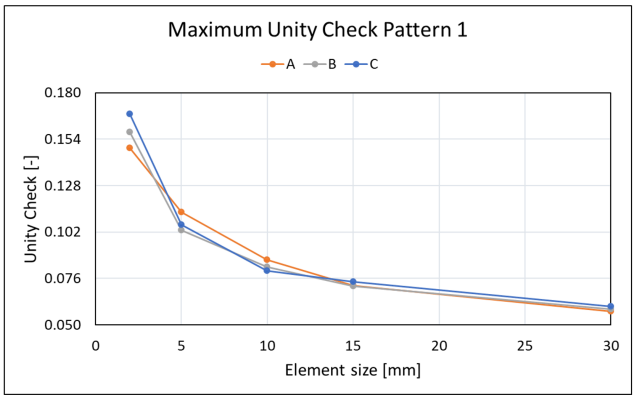
The first observation is that in all four patterns, the models converge to the same maximum deflection, which is depicted by the dotted lines. However, Models A, B, and C show clear differences in their maximum deflection in cases where their element size is equal to or larger than 15 mm. Furthermore, as there are more elements along the thickness of the core, as in Model C, the difference between the maximum deflection at an element size of 2 and 30 mm increases. This is clearly visible in the graphs, as model A has the smallest difference, followed by model B, and model C has the largest difference. Thus, the number of elements over the thickness of the core has an effect on the convergence speed, or, in other words, the slope of the line. This process is even more obvious when displayed with the continuous lines, where the red line is always above the blue line and the blue line is always above the green line, which represent models C, B, and A, respectively. Although in pattern 4, the blue and green lines are almost on top of each other. Another observation is that in all models and patterns, real convergence appears. Moreover, it appears the rotated geometry is slightly stiffer than the other geometry because patterns 3 and 4 have a marginally smaller maximum deflection than patterns 1 and 2.

Besides the maximum deflection, the figures also show the maximum Unity Check. The graphs show that as the element size is reduced, the magnitude of the Unity Check increases. This process can be explained by a phenomenon that occurs at, for example, corners where the stresses go to infinity as the element size decreases. This has been widely known and studied since the end of the last millennium. For example, a paper released in 1971 by England [33], describes the occurrence of stress singularities in linear elasticity, which occur at corner points or at locations with a change in boundary conditions. In all patterns, the four corners also have a change in boundary conditions, which would make them prone to stress singularities. Thus, the increase in stress is likely caused by stress singularities. This process occurs as the element size is around 5 or 10 mm, depending on the pattern. Furthermore, as the element size is less than 10 mm, large differences occur in the Unity Check

in models A, B, and C. This is particularly present in pattern 3 at an element size of 5 mm. Moreover, the stresses also appear to be lower in patterns 3 and 4 compared to patterns 1 and 2. Finally, tables of all the numbers in the graphs can be seen in Appendix F.

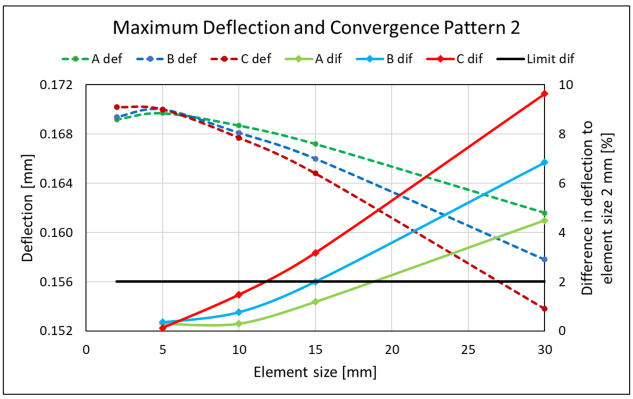


(a) Maximum deflection

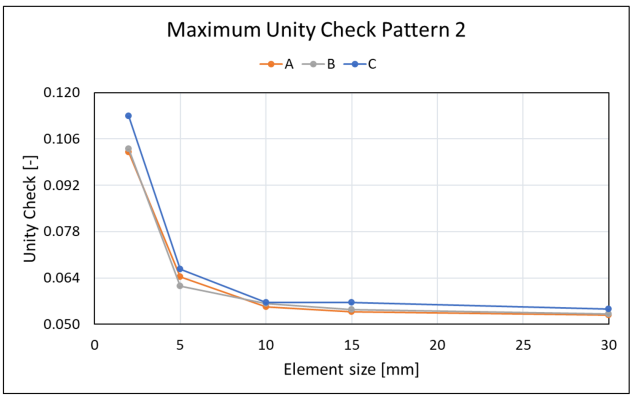


(b) Maximum Unity Check

Figure 76: An overview of the maximum deflections and Unity Checks per Karamba3D model of Pattern 1.

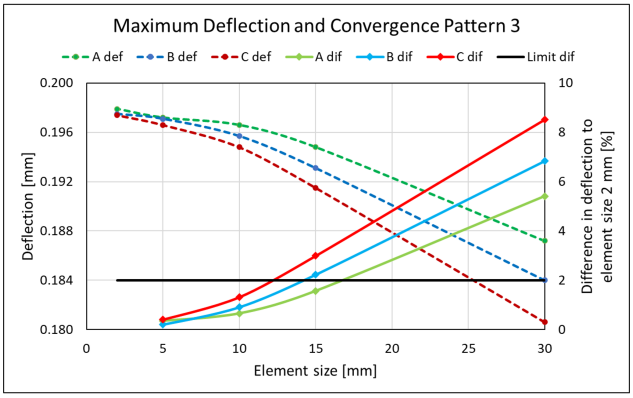


(a) Maximum deflection

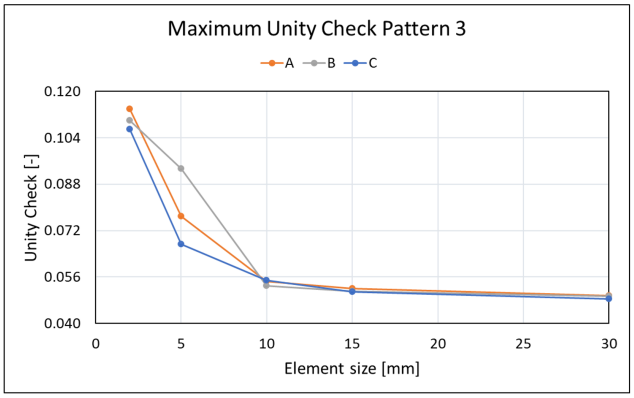


(b) Maximum Unity Check

Figure 77: An overview of the maximum deflections and Unity Checks per Karamba3D model of Pattern 2.

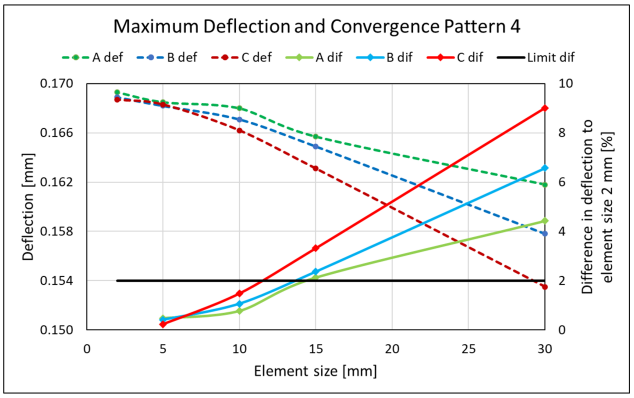


(a) Maximum deflection

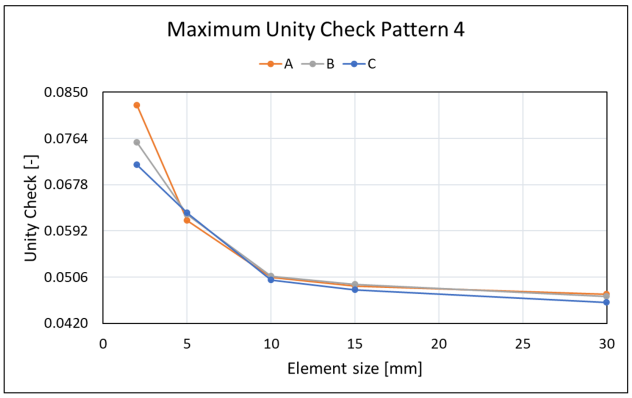


(b) Maximum Unity Check

Figure 78: An overview of the maximum deflections and Unity Checks per Karamba3D model of Pattern 3.



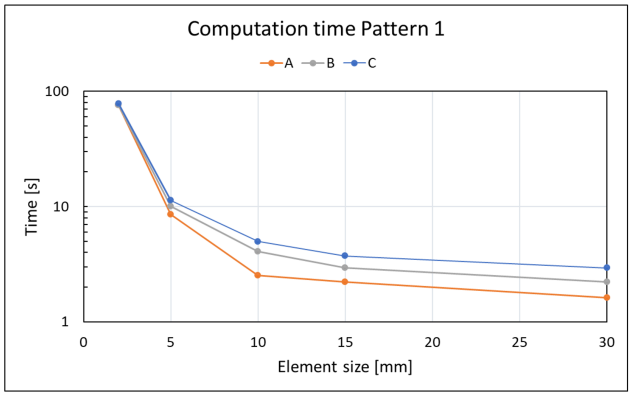
(a) Maximum deflection



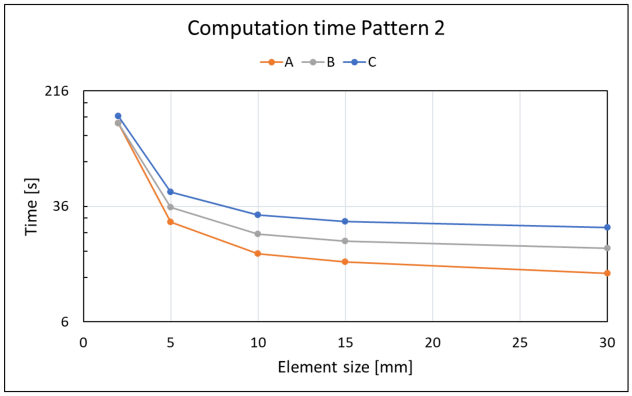
(b) Maximum Unity Check

Figure 79: An overview of the maximum deflections and Unity Checks per Karamba3D model of Pattern 4.

The computation times are shown in Figures 80 and 81, where a logarithmic vertical axis is used to emphasize the sharp increase in computation time as the element size reduces. In patterns 1 and 3, a logarithmic basis of 10 is used, and in patterns 2 and 4, a basis of 6. This implies that as the number of hexagons increases, the computation time increases. For example, model A4 has a calculation time of 76 and 130 seconds in patterns 1 and 2, respectively. Thereby, in general, as the number of elements over the thickness increases, the computation time increases, which is chiefly caused by the fact that the computation time increases as the number of elements increases.

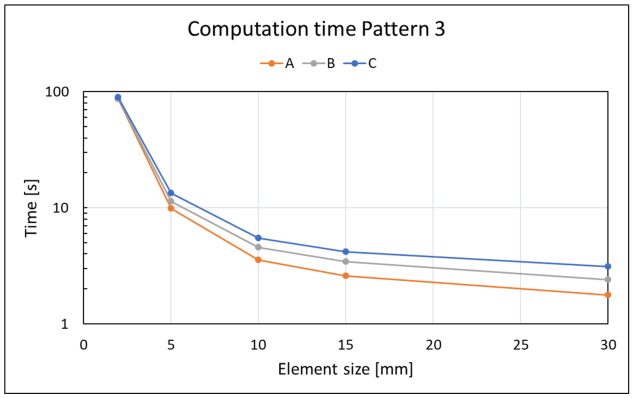


(a) Maximum deflection

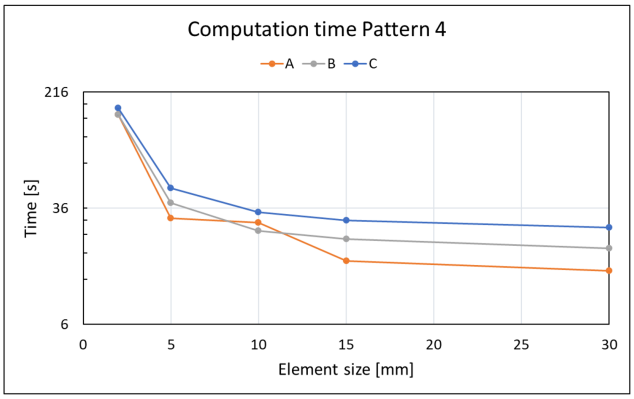


(b) Maximum Unity Check

Figure 80: An overview of the computational time per Karamba3D model of Pattern 1 and 2.



(a) Computation time Pattern 3



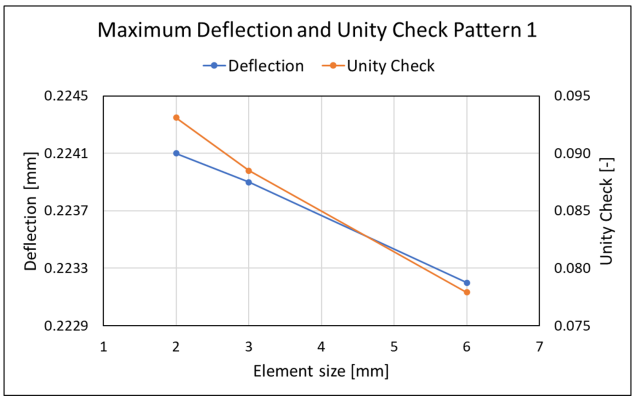
(b) Computation time Pattern 4

Figure 81: An overview of the computational time per Karamba3D model of Pattern 3 and 4.

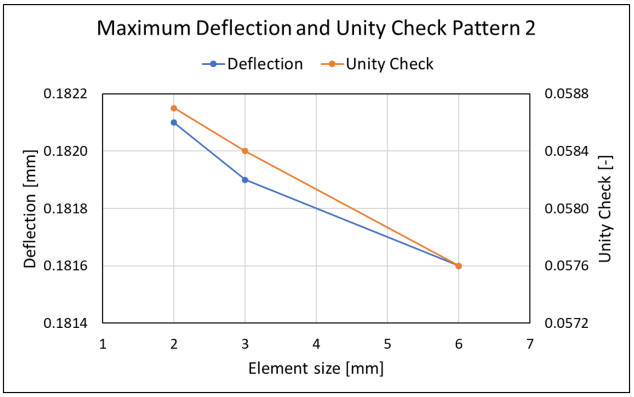
12.3.2 Results Diana with Shells elements

The results of Diana with shell elements are shown in Figures 82 and 83. These graphs display the maximum deflection and Unity Check for all patterns at different element sizes. In all cases, a reduction in the element size increases the maximum deflection and Unity Check, which means that the larger element sizes underestimate the stresses and deflections. Subsequently, as opposed to the observed behaviour in Karamab3D, the existence of a stress singularity does not appear clearly. Also, there is no strong evidence of convergence except from pattern 4; however, it has to be stated that the difference in the maximum deflection of an element size of 6 and 2 mm is between 0.44 and 0.09 %. Therefore, it can be concluded that the mesh refinement is sufficient. Nevertheless, the difference in Unity Check between an element size of 6 and 2 mm is between 2 and 28 %. Especially, pattern 3 has significant changes in its Unity Check as the element size reduces.

Additionally, when comparing the maximum deflection of patterns 1 and 3, pattern 1 appears to be less stiff, although the stresses are slightly smaller. The same observation is valid for patterns 2 and 4. It has to be stated that the maximum Unity Check appears to be in the bottom glass layer in all patterns. These occur either at the corners or at the connections with the core geometry. Furthermore, images of the stresses can be found in Appendix D, and the tables containing the numbers used in the figures are shown in Appendix F. The images contain the Von Mises stresses on the top and bottom sides of the panel, as well as the core geometry for all patterns and element sizes.

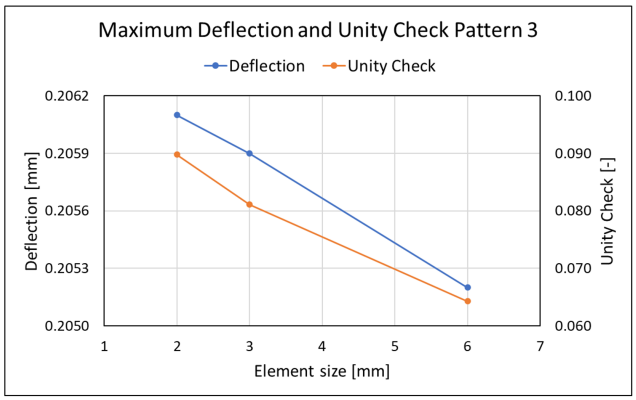


(a) Pattern 1

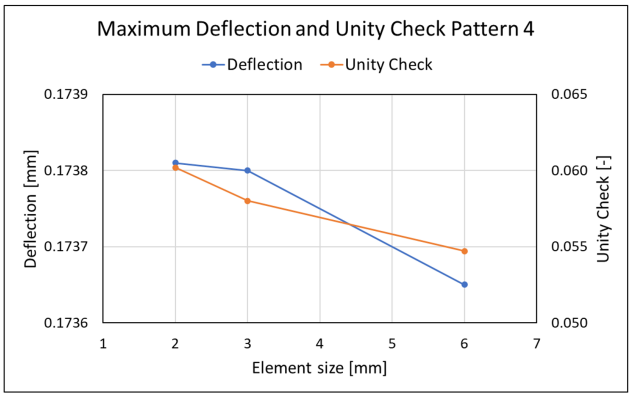


(b) Pattern 2

Figure 82: An overview of the maximum deflections and Unity Check per element size of Pattern 1 and 2.



(a) Pattern 3



(b) Pattern 4

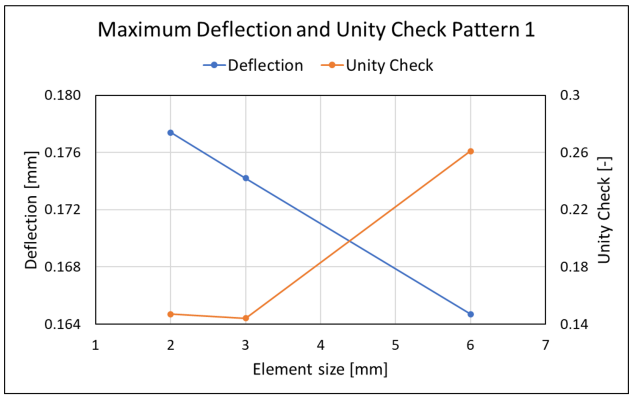
Figure 83: An overview of the maximum deflections and Unity Check per element size of Pattern 3 and 4.

12.3.3 Results Diana solid

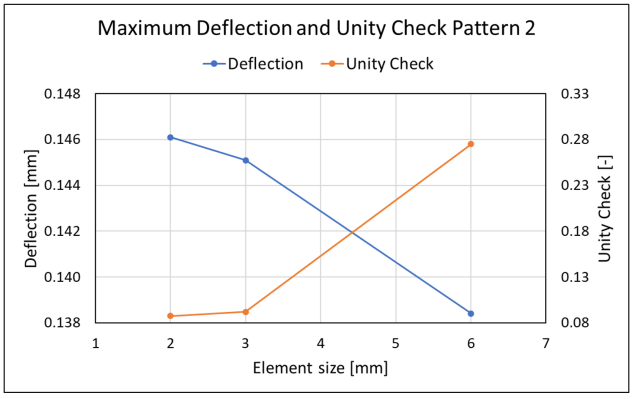
Figures 84 and 85 present the maximum deflection and Unity Check of the different patterns of Diana with solid elements. First, in all patterns, the maximum deflection increases as the element size decreases. While patterns 2 and 4 show convergence, patterns 1 and 3 do not. Additionally, the difference in deflection at an element size of 2 and 3 mm in patterns 1 and 3 is around 1.8 %, and the difference in deflection in patterns 2 and 4 is less

than 1 %. With regard to the arbitrary limit of 2 %, all patterns have sufficient mesh refinement at an element size of 2 mm. The difference in maximum deflection of an element size of 3 and 6 mm lies between 4 and 5 %. Furthermore, patterns 3 and 4 appear to be stiffer than patterns 1 and 2, which appeared as well in the Diana models with shell elements and the Karamba3D models.

Besides, the stresses tend to decrease as the element size is reduced, except in pattern 4. Next, the stresses seem to converge, which is especially clear in patterns 1 and 3. Subsequently, the difference in the Unity Check of an element size of 2 and 3 lies between 2 and 21 %, while the difference in the Unity Check of an element size of 3 and 6 is as large as 200 %. Thus, combined with the difference in maximum deflection, a mesh of 6 mm elements is not refined enough. The images of the stresses at the top and bottom of the panel as well as the core are shown in Appendix E, and the tables containing the maximum deflections and Unity Checks are presented in Appendix F.

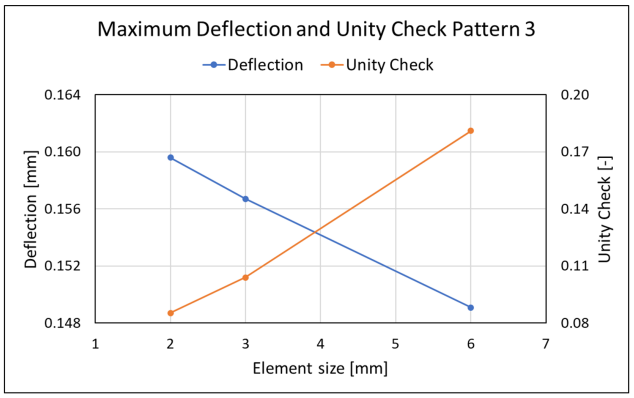


(a) Pattern 1

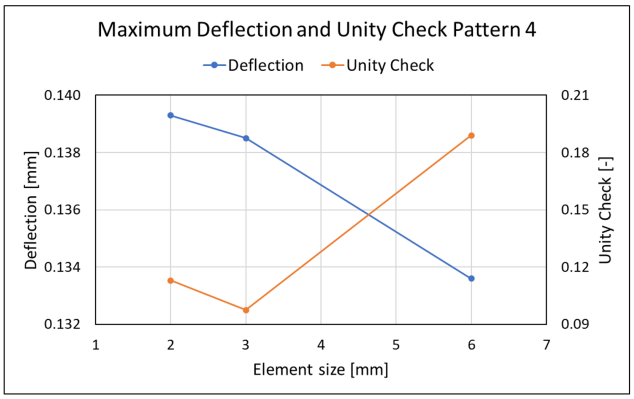


(b) Pattern 2

Figure 84: An overview of the maximum deflections and Unity Checks per element size of Pattern 1 and 2.



(a) Pattern 3



(b) Pattern 4

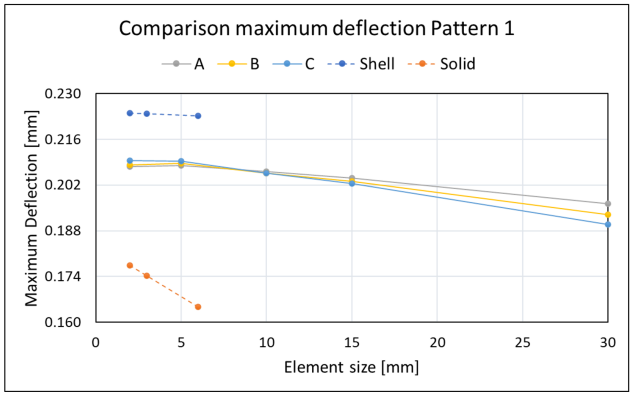
Figure 85: An overview of the maximum deflections and Unity Checks per element size of Pattern 3 and 4.

12.4 Comparison Karamba3D and Diana with shell and solid elements

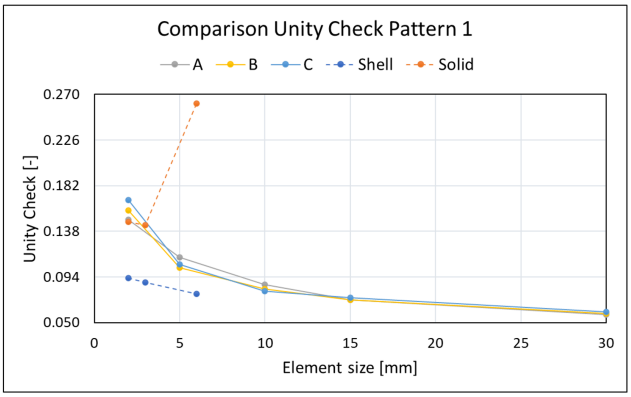
The Figures 86, 87, 88, and 89 show all the used models and their results with respect to the maximum deflection and Unity Check. First, the maximum deflection in all patterns in the Karamba3D models lies between the Diana results, where the Diana with shell elements has the highest maximum deflection and the Diana with solid elements has the lowest maximum deflection. Additionally, Karamba3D results in closer proximity to Diana with shell elements than Diana with solid elements. This makes sense as the Karamba3D models are also based on shell elements and should produce similar results. Moreover, as the Diana with solid elements is probably the most exact solution, the Karamba3D models always overestimate the maximum deflection. The overestimation is in the order of 16 to 24 %.

Furthermore, the Unity Check, in general, the Diana with shell elements provides the lower boundary in which the Unity check lies, and depending on the pattern, the Diana with solid elements provides the upper

boundary. This is especially the case for pattern 4, while in the other three patterns, the Unity Check of Diana with solid elements with an element size of 2 mm lies below the Karamba3D models. Additionally, in pattern 3, the Unity Check of Diana with shell elements is even higher than that of Diana with solid elements. Thus, in three patterns, the Karamba3D models slightly overestimate the stresses, and in one pattern, it underestimates the stresses in case it is assumed that the Diana with solid elements is the most exact solution.

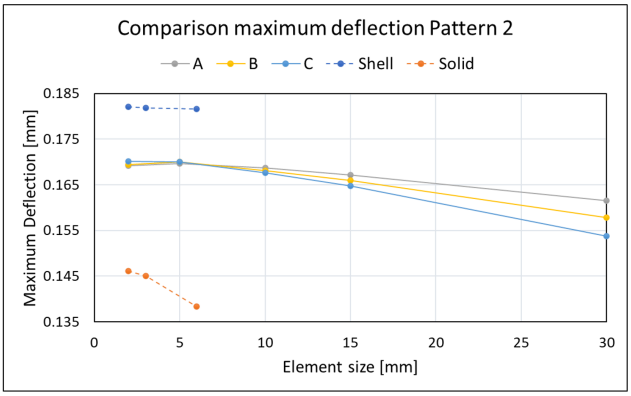


(a) Deflection

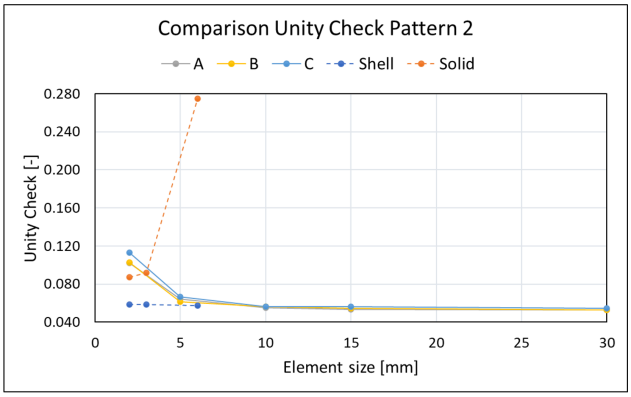


(b) Unity Check

Figure 86: A comparison of the maximum deflection and Unity Check of all the models of Pattern 1.

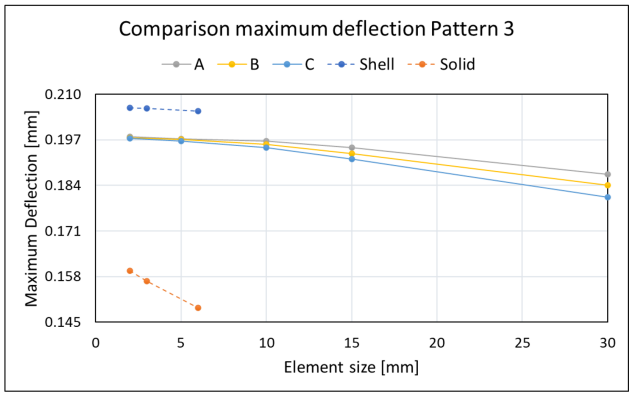


(a) Deflection

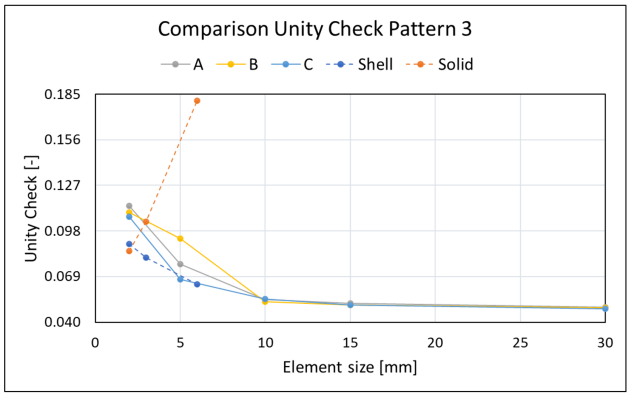


(b) Unity Check

Figure 87: A comparison of the maximum deflection and Unity Check of all the models of Pattern 2.

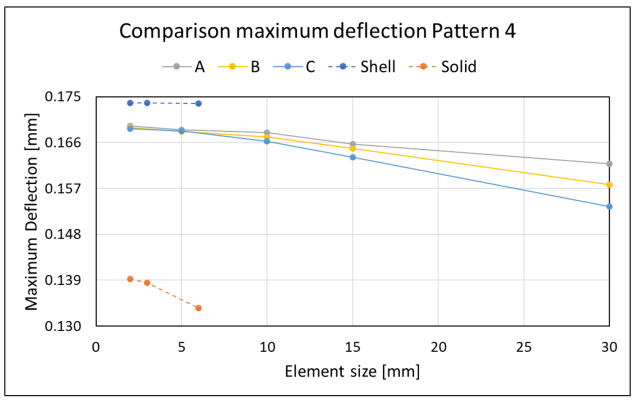


(a) Deflection

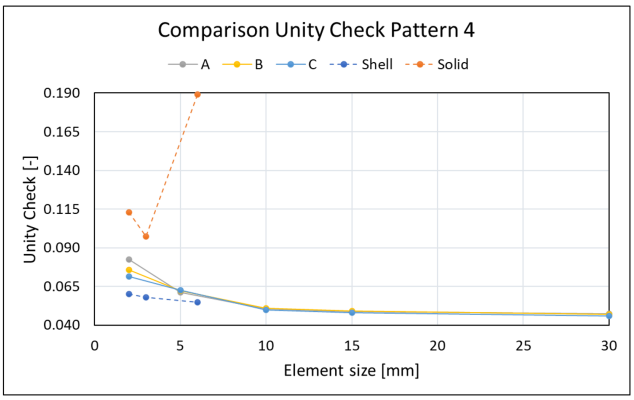


(b) Unity Check

Figure 88: A comparison of the maximum deflection and Unity Check of all the models of Pattern 3.



(a) Deflection



(b) Unity Check

Figure 89: A comparison of the maximum deflection and Unity Check of all the models of Pattern 4.

12.5 Conclusion

The overall behaviour of the Karamba3D models shows that clear convergence occurs in the maximum deflection, and although the different models, such as models A, B, and C, result in different maximum deflections with large element sizes, all are converging to roughly the same deflection. Thereby, an element size of at most 10 mm is required to limit the difference in maximum deflection with the model using 2 mm to 2 %. Additionally, the maximum deflection found in Karamba3D models is always smaller than that in Diana models with solid elements. Therefore, the maximum deflections obtained from Karamba3D can be used in the preliminary study without applying an additional safety factor.

Furthermore, Karamba3D seems to be prone to stress singularities, which is especially the case when the element size is smaller than 10 mm. Moreover, in cases where an element size of 10 mm or more is used in Karamba3D, the Unity Check always lies below the Diana models with solid elements. This difference is equal to 83, 58, 61, and 95 % in patterns 1, 2, 3, and 4, respectively. Therefore, a safety factor is applied, which doubles the obtained Unity Check from Karamba3D when using an element size of 10 mm. Beside the consideration of limiting the computation time, a smaller element size would increase the difference in the Unity check of the models A, B, and C, which is especially visible in pattern 3.

Subsequently, the use of shell elements in this geometry creates some problems. First, an equal thickness is assumed along the core, which is not completely true where two shell elements meet at a different angle. Second, situations with a nonzero inclination, the height of the shell elements, which is the thickness in a perpendicular direction, is not equal to the line width. Finally, the different orientations of the shell elements in the core and glass sheets create inconsistency in the transition between these two materials. As shown in Table 26, the stress σ_{zz} is not computed. However, when using a general coordinate system, where the z-direction is in the direction of the thickness, the shell elements in the glass layers do not compute σ_{zz}, while in the shell elements of the core, the general σ_{zz} is equal to the local σ_{xx} or σ_{yy}. Thus, inducing a discrepancy. Additionally, a similar problem arises with the rotational DOFs.

To conclude, although the use of shell elements is not optimal, there is no alternative available in Rhino. Therefore, Karamba3D will be used, and in this case, the element size is chosen to be equal to 10 mm, and a safety factor of 2 is applied at the Unity Checks.

13 Strength experiments of facade panels

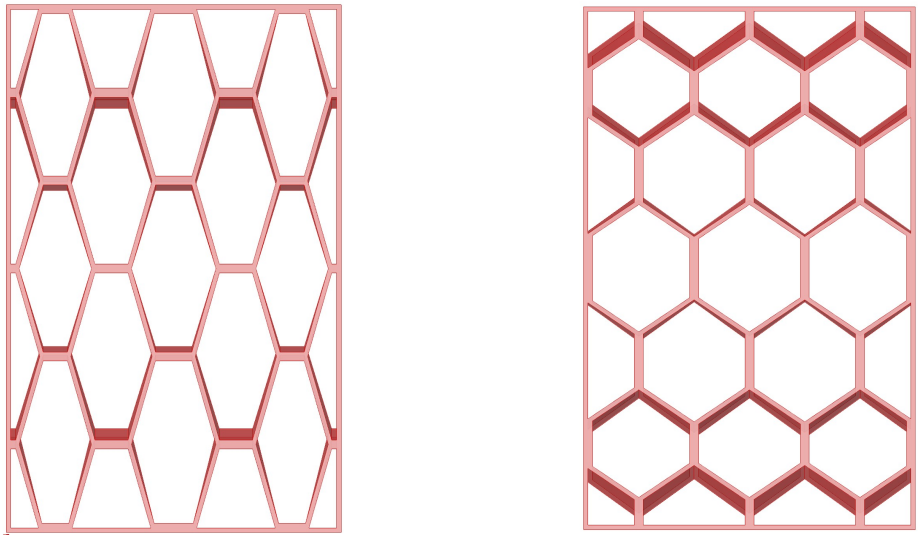
Experiments, which measure the deflection, have been done to have a more profound understanding of the behaviour of the facade element in static loading conditions in the linear-deformation domain. Two core geometries are used, and an area load is applied to them in order to represent the real loading condition by wind as close as possible. First, the materials used in the experiments are listed; then, the set-up of the experiments is discussed before the procedure is explained. Furthermore, the deformation results are discussed, and the results of a Finite Element Analysis of this structure are presented. Finally, the conclusions are drawn.

13.1 Materials

The used core geometries are presented in Figure 90. The geometries both have a line width of 6 mm, a layer height of 1.8 mm, and a total thickness of 18 mm, which corresponds to 10 layers. Additionally, both geometries have only one milled face. A milled face provides better connectivity between the glass and the core. Moreover, the core is approximately 710 by 450 mm.

The geometry in Figure 90b has six hexagons vertically and three horizontally. Also, the vertical lines have altered lengths, where lines in the center of the panel are elongated and lines at the edges are shortened. The last characteristic is the vertical inclination. The magnitude of the inclination is determined by the vertical distance between the individual line and the center of the core, with a maximum of 40 degrees.

The geometry in Figure 90a has six hexagons horizontally and three vertically. Also, the length of the horizontal lines is adjusted, such that the lines in the center are elongated and the lines at the sides are shortened. Next, the inclination is in the horizontal direction, which has a maximum angle of 40 degrees. The angle depends on the horizontal distance between the individual lines and the center of the core.



(a) Hexagons in the horizontal direction

(b) Hexagons in the vertical direction

Figure 90: The two core geometries used in the experiments

The core structure is connected through glue, with two glass sheets on either side. The glass sheets have the same dimension and are 1.1 mm thick. The material properties of the prestressed glass and the core are presented in Table 28.

Material Properties	Glass	Core
Material type	Falcon alumino-silicate glass	FGF PET-G
Density $[kg/m^3]$	2480	1270
Young's Modulus $[GPa]$	70	2.02
Poisson ratio [-]	0.21	0.4
Compressive strength $[MPa]$	800	50
Tensile strength $[MPa]$	800	50

Table 28: Overview material properties of the core and glass [2, 10].

13.2 Panel creation

The panel creation concerns the gluing of the glass layers to the core. The first step is to clean one of the future contact areas of the core with the glass with isopropanol. This will remove most of the surface contamination and improve the bonding. The next step is to manually apply the glue, LOCTITE AA3345, to this surface with a brush. After applying the glue to the whole surface, the polycarbonate sheet is placed on top. The glass sheet should be clean and contain the minimum amount of dust. After laying down the glass sheet, wooden blocks were situated along the edges to prevent any sideways movement of the glass during the curing process. Thereafter, the glue needs to be cured with UV light. First, the UV light is placed directly on top of the glass sheet for 45 seconds, which will provide the initial hardening of the glue. After this initial curing, the glue is cured for the second time in 30 seconds. This makes sure the glue is fully cured and the edges of the core can be cleaned with isopropanol, as some excess glue may have spread. The same process is followed on the other side. In case the side is not milled, extra attention is required to ensure sufficient glue is applied. The gluing process is shown in Figure 92, which shows the device emitting the UV light as well as the two wooden blocks that prevent any movement of the glass layer during the procedure. Additionally, Figure 91 shows the printing process of one of the core geometries at the University of Dresden, Germany. The finished panels are displayed in Figure 93.

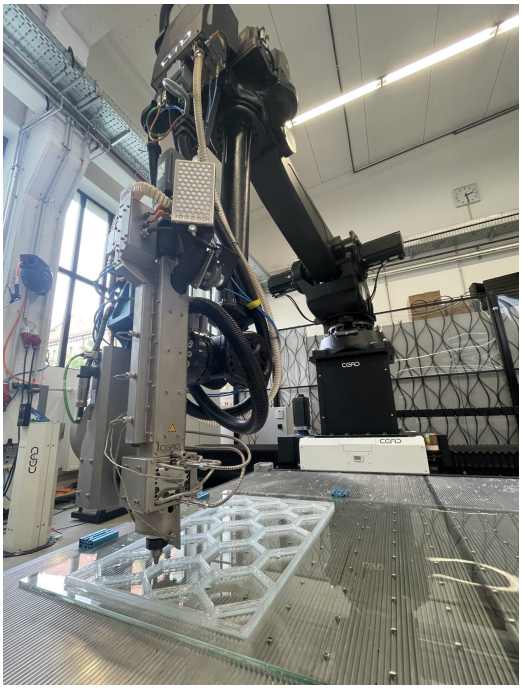
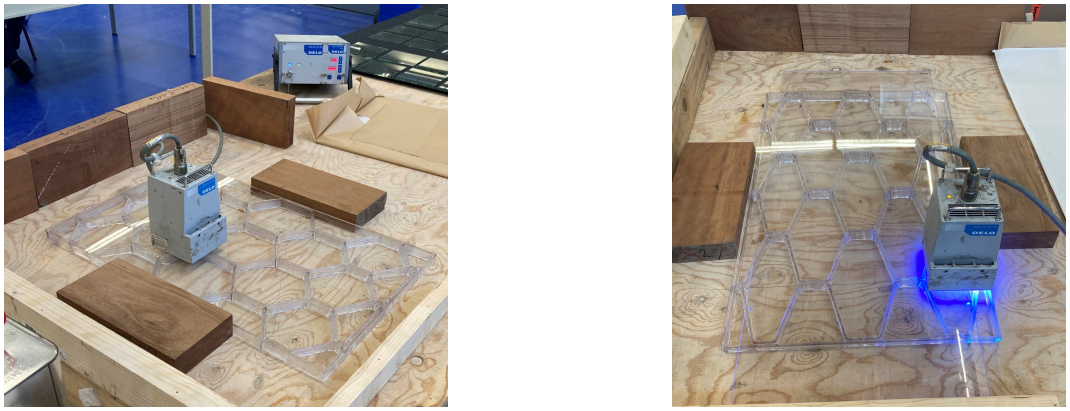


Figure 91: The printing process of the core panel.



(a) The glueing process from the side.

(b) The glueing process from above.

Figure 92: The glueing process.

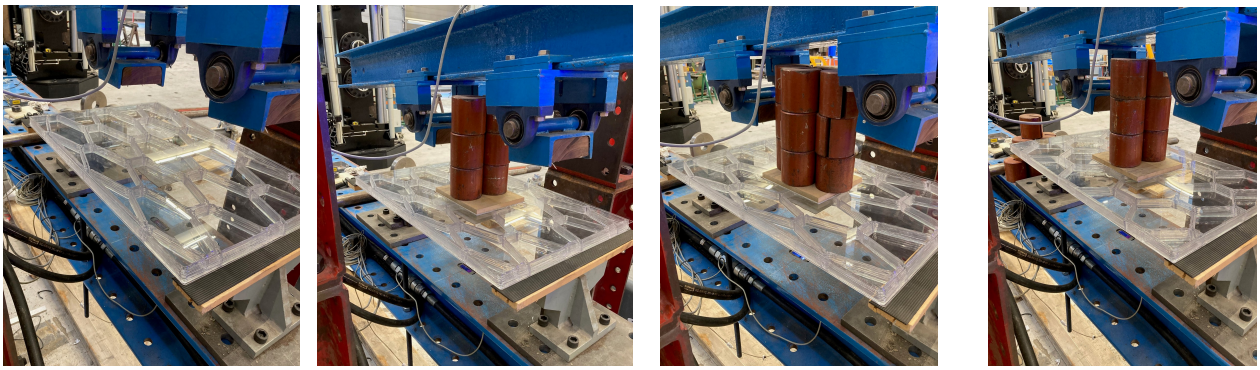
(a) The finished panel with hexagons in horizontal direction. (b) The finished panel with hexagons in vertical direction.

Figure 93: The final panels.

13.3 Procedure strength experiment

The strength experiments are a two-sided supported panel that is loaded in the center of the panel, as shown in Figure 94a. The supports allow for free rotation of the panel, are along the full length of the shortest side of the panel, and a distance of 67 cm is between the supports. The blue beam above is not part of the experiment. Under the center of the panel, two lasers are embedded in the structure, which can measure the distance to the structure above. As the glass structure is transparent, a piece of cardboard is taped to the bottom of the panel to ensure good readings. Also, the milled layer of the core is positioned at the bottom of the panel to ensure the best connection between the core and glass. Subsequently, the load is applied in the center of the panel, as shown in Figure 94b. The red cylinders weigh 4.00 kg each, and a piece of cardboard and wood are placed between the cylinders and the glass to prevent damage to the glass. The cardboard and wood have a size of 17 by 20 cm with its longest side placed parallel to the longest panel side, and weigh around 150 grams in total.

Furthermore, the temperature in the lab during the experiments is 24 degrees Celsius, and the panel is loaded in steps of 8.00 kg, which is equal to two red cylinders. The panels are loaded with no more than 48 kg. The load increments are performed in such a way that the first two red cylinders are placed beside each other, and in the subsequent two load increments, the two cylinders are placed on top of the previously placed cylinders. The result after three load increments is visible in Figure 94d. Then, two cylinders are added to obtain a diamond shape. In the following two load steps, each time two cylinders are stacked on top of them to reach the final situation shown in Figure 94c.



(a) (b) (c) (d)

Figure 94: An overview of the structural experiments.

13.4 Results experiments

The results of the experiment are shown in Figures 95 and 96. The deflection is the average of the two lasers, and the deflection of each laser is presented in Appendix I. First, the experiment with the rotated pattern, the figure shows that during the experiment, the load increments of 8.00 kg until a weight of 48 kg is reached. The dots on the deflection line show all the data points that were gathered during the experiment. This shows that as the weight is increased, the deflection changes rapidly, and thereafter, the deflection changes hardly, except for the last load increment. After a weight of 48 kg is reached, a small jump in deflection of 0.2 mm is visible, followed by a steady increase in deflection until the experiment finishes. This small increase is probably caused by a local failure in the connection between the core and the glass layer, and then the connection slowly weakens. Moreover, adding 8 kg on average increases the deflection by on average 1.1 mm.

The second experiment uses a non-rotated pattern, and during the first four load increments, it appears that the deflection remains fairly constant. However, as the load is increased to 40 kg, initially the deflection appears steady, but after a few seconds, a large jump in the deflection is seen. Simultaneously, a sharp noise was heard. This jump is probably also caused by a failure in the glue connection, although this is a larger failure compared to the orientated pattern. During the first four load steps, the deflection increases by roughly 1.2 mm, which is slightly more than the rotated pattern. Nevertheless, the not-orientated panel is slightly stiffer until a load of 16 kg, as its deflection equals 1.8 mm, while the rotated pattern has a deflection of 2.3 mm.

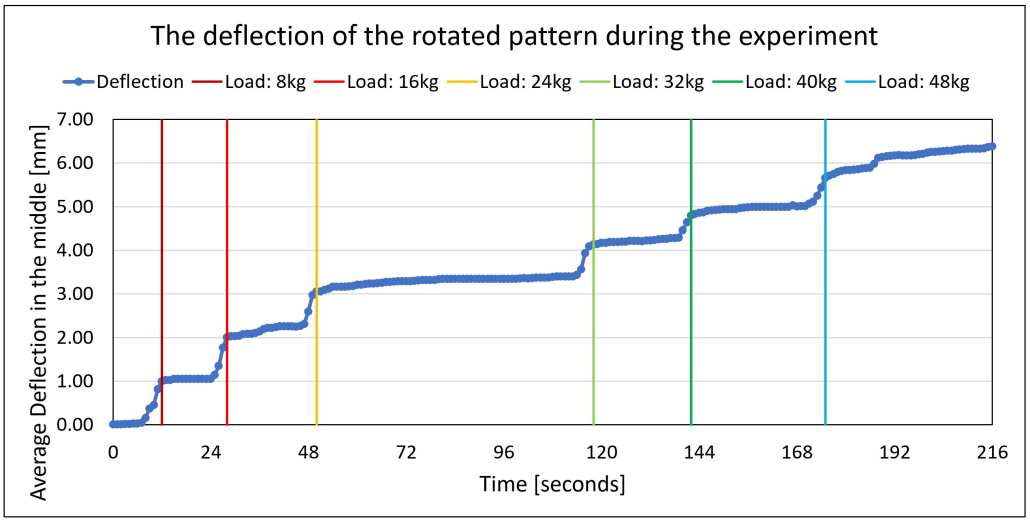


Figure 95: The deflection of the rotated pattern during the experiment.

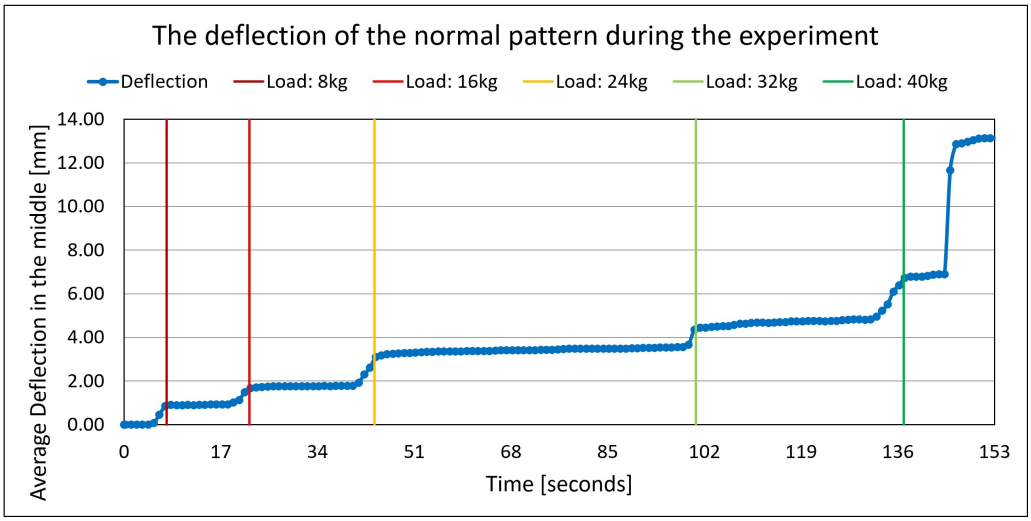


Figure 96: The deflection of the non-rotated pattern during the experiment.

Figure 97, gives a clear image of the behaviour of both panels during the test and how they compare. The deflections represent the deflection at the time between two subsequent load increments. For example, the deflection at 8 kg of the rotated pattern takes around 19 seconds. However, as the deflection is not constant after the last load step, the deflection equals the measured deflection a few seconds after applying the load. Thus, before the small or large jump in deflection in both panels. This provides the initial response of the panel, as if the load is only applied for a few seconds. The non-rotated panel shows a decrease in stiffness as the total load increases, especially, a stiff behaviour is observed until a load of 150 N with a stiffness of roughly 89 N/mm. Then, the stiffness reduces to 53 N/mm and remains relatively constant until a load of 300 N. Afterwards, the stiffness decreases further to 38 N/mm in the last load step. The average stiffness is 58 N/mm.

The rotated pattern appears to have an increase in stiffness as the load increases because the slope increases as the load exceeds 235 N. The stiffness with a load less than 235 N is equal to 70 N/mm and thereafter reaches 94 N/mm. The average stiffness lies around 80 N/mm. Thus, when comparing both patterns, the non-rotated panel shows a stiffer initial response, while the rotated pattern has a higher stiffness after reaching and surpassing a load of 235 N. Therefore, the difference in the final deflection at a load of 392 N (40 kg) is 1.8 mm.

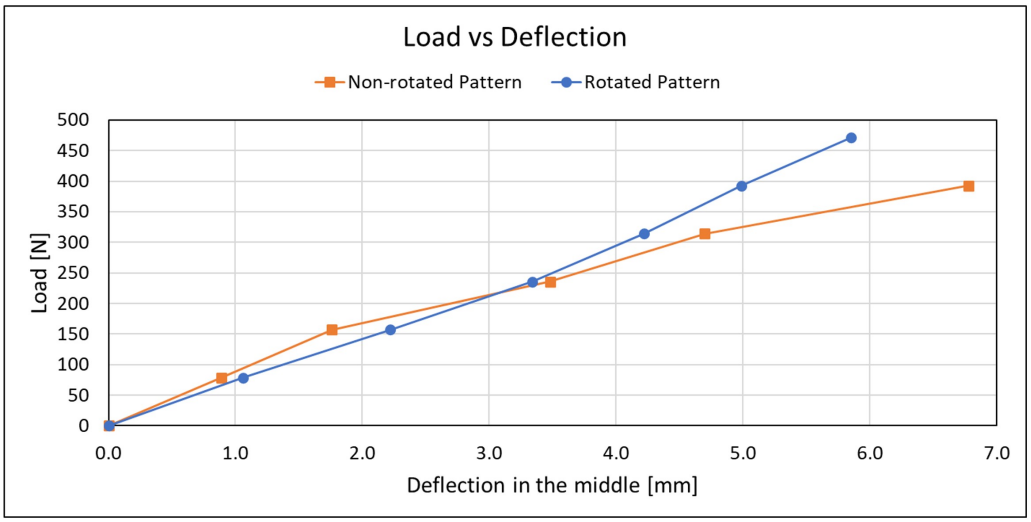


Figure 97: The comparison of the stiffness of both tested panels.

13.5 Results FEA

The Diana models use the same core geometry that is obtained from the Rhino model. Next, the glass layers are simply added as volumetric geometries. The material properties only require the E-modulus and Poisson's ratio, as they are both linear elastic. The glue connection is not modelled; thus, a perfect connection between the glass and core is assumed, which is a huge simplification. Furthermore, the loading conditions are simulated as a constant applied pressure over an area of 17 by 20 cm situated in the middle of the panel, and the gravity load is neglected. The wooden slab, which is present in the experiments, is not included in the Diana model because an uniform pressure is applied instead of concentrated forces due to the cylinders; thus, the function of the wooden slab to spread the load is no longer needed. This is an assumption that simplifies the finite element model; however, it is almost equal to the real situation.

Next, the support conditions, the panel is placed on two wooden beams over the complete width, and they have a 2 cm overlap at both ends. Thus, this support can only be loaded in compression, and it has no resistance to tension. Therefore, to closely resemble this, a boundary interface element is added. The boundary interface elements are placed between the lower glass panel and the supports at the location of both supports. These elements are non-linear elastic and have the same material properties as the glass. Additionally, the boundary interface elements have no tension with shear-stiffness reduction, which means that they prevent the supports from having tensile reaction forces. Finally, the supports, on one side, they prevent translation in all three directions while the other support allows for translation along the length of the panel and prevents it in the other two directions. In reality, the panel can indeed move parallel to the length direction as the wooden planks that provide the support can slide; however, there is some resistance. This resistance is assumed to be zero in the Diana model for simplicity purposes.

The applied mesh uses an element size of 2 mm. In the next paragraph it is discussed if this a sufficiently

small element size. Subsequently, a structural non-linear analysis is performed, which has ten load steps of 10 % of the maximum load used in the experiments. This equals 40 kg with the non-rotated panel and 48 kg with the rotated pattern. Also, the Newton-Raphson method is used with a maximum of 50 equilibrium iterations. During both analyses no more than 10 iterations and, on average, 1 iteration were required to have convergence in each load step.

The results of the analysis are displayed in Figure 98. This shows that in both cases, the deformation is linear, which should be the case as linear elastic elements are used in the panel. Also, the stresses in the glass and core remain well below the maximum stresses. Subsequently, the rotated pattern is stiffer than the non-rotated pattern, which also occurred during the experiments. The maximum global deflection is 0.582 and 0.658 mm for the non-rotated pattern and the rotated pattern, respectively. The tests have also been performed with 3 mm elements to know if a small enough element size is used. The maximum global displacements are 0.586 and 0.642 mm with the non-rotated and rotated pattern, respectively. This has absolute difference of 0.69 % and 2.43 %, which means that in case of the non-rotated pattern the mesh is small enough, and for the rotated pattern the mesh ideally should be smaller. Therefore, the Diana model with a rotated pattern has also been performed with 1.5 mm elements. This results in a maximum global deflection of 0.665 mm, which is 1.05 % larger than the maximum deflection with an element size of 2 mm. The stiffness of the rotated and non-rotated patterns is roughly 708 and 674 N/mm, respectively. Appendix I displays the boundary conditions, loading conditions, deflections, and principal tensile stresses. This shows that the area within the hexagon which is loaded by the external load, has large deflections and stresses. These stiffnesses are much larger than the stiffnesses found in the experiments, therefore, additional tests in Diana are conducted to take a closer look into possible causes of this difference.

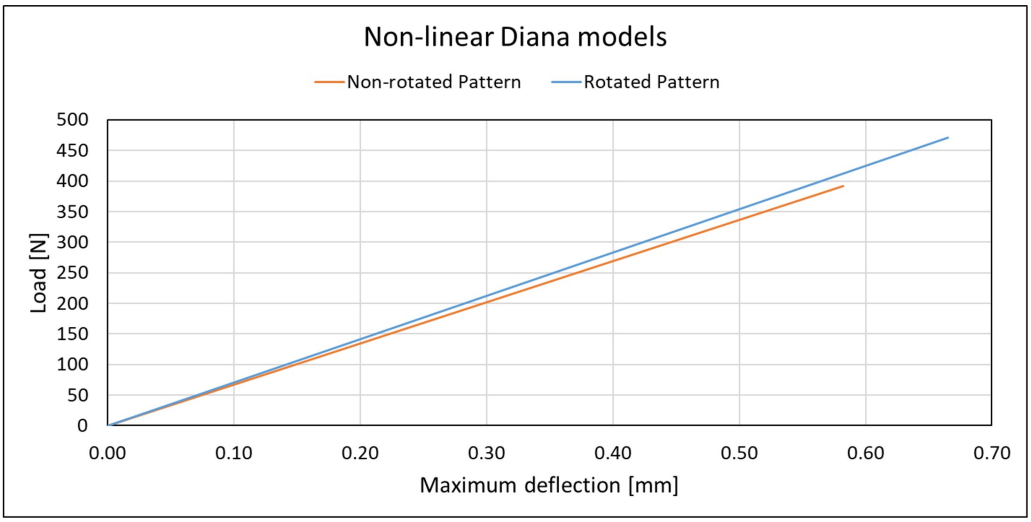
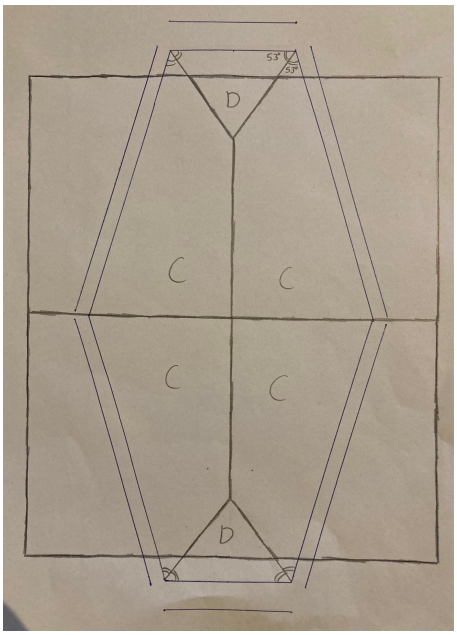


Figure 98: The Diana models of both experiments.

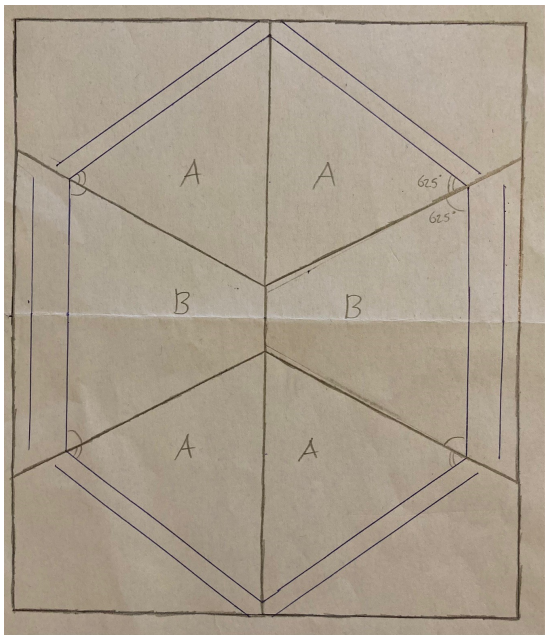
13.5.1 Additional tests related to the core

The first additional tests are related to the core itself and examines its stiffness. These tests only consider the core, therefore, the area load is converted to loads applied to the core directly. The loads are only applied to the hexagon in the middle of the panel, which is below the wooden plate in the experiments. The hexagon is divided into six parts, and it is assumed that the load on each part is an uniform pressure. The magnitude of the pressure is determined by the percentage of the uniform area load over the 17 by 20 cm plate is carried by each part. This percentage is obtained by dividing the plate as shown in Figure 99.

This division results in the following percentages of the total load: A is 16.56 %, B is 16.87 %, C is 23.67 %, and D is 2.66 %. These loads are applied as an uniform pressure over the area of the part. This results in the following pressures: 0.1302 N/mm^2, 0.06363 N/mm^2, 0.1690 N/mm^2, and 0.01976 N/mm^2, respectively. This tries to closely match the loading conditions while preventing to complicate the conversion by, for example, adding bending moments. The structural model of both tests is shown in Figures 322 and 323 in Appendix I.



(a) Dividing the area in six parts in the rotated pattern.



(b) Dividing the area in six parts in the non-rotated pattern.

Figure 99: The division of the area into six parts

Figure 100 shows the relationship between the load and deflection. It appears that the rotated pattern is stiffer than the non-rotated pattern, which is also found in the tests with the glass panels. Furthermore, the maximum deflection is around 80 mm, which is much larger than the deflection in the Diana models with the glass sheets and the experiments. This gives a stiffness of 4.79 N/mm and 6.01 N/mm for the non-rotated and rotated pattern, respectively. Additionally, the maximum deflection is smaller with the rotated pattern, while the applied load is larger. This behaviour was not displayed in the Diana models with glass. This is probably caused by the change in the loading condition. Moreover, these tests use an element size of 2 mm, and the same tests are performed with 3 mm elements. The absolute change is 1.5 % and 0.7 % in the rotated and non-rotated pattern, respectively. Thus, the rotated pattern is analysed again with 1.5 mm elements. This result has an absolute change of 0.3 % with the results of 3 mm elements. The stiffness is reduced to 5.99 N/mm. However, it has to be noted that the maximum stress of the material exceeds the stress shown in Table 28. The rotated pattern reaches the stress limit at a deflection of 43 mm and an applied load of roughly 260 N or 26.4 kg. The non-rotated pattern reaches the stress limit at 32 mm deflection and an applied load of around 160 N or 16 kg. This shows that the rotated panel is stronger than the non-rotated panel in this case.

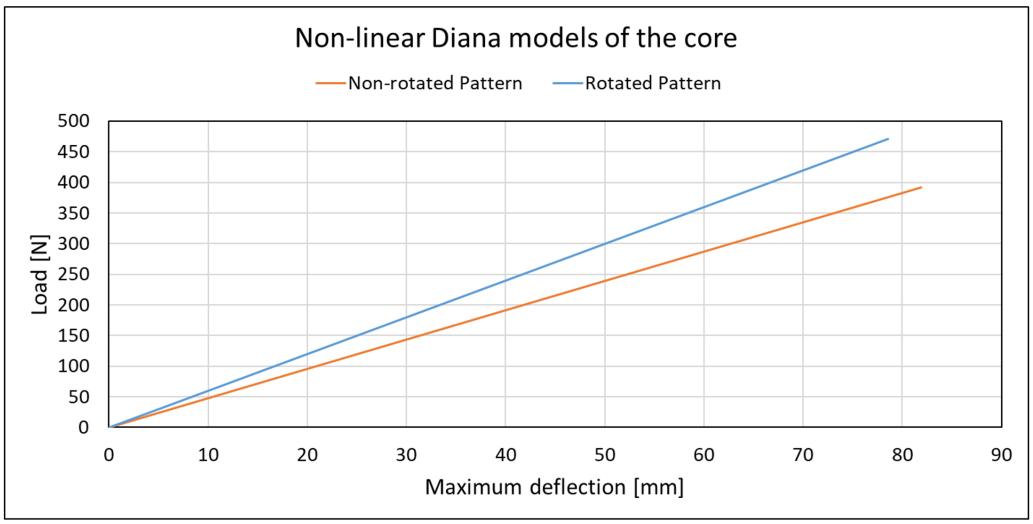


Figure 100: The maximum deflections of the cores.

Second, the boundary conditions are examined with the use of analytical calculations. These tests use a homogeneous glass plate, which has the same dimensions, boundary conditions, and loading conditions as the performed tests. However, it has a core that is completely made of glass, which results in a glass panel of 710 x 450 x 20 mm. The structural model uses an element size of 3 mm, and is displayed in Figure 331 Appendix I. Additionally, a Diana model with a line support is added for comparison. These models have a deflection of 0.132 mm at a total load of 48 kg or 470.9 N. Thus, the supports with the boundary interface elements and the line support, result in the same maximum deflection. The absolute difference in maximum deflection between 3 and 4 mm element size is less than 0.02 %. Moreover, analytical calculations based on the beam theory are used. The calculations use the forget-me-nots from structural mechanics. Equation (44), is a simply supported beam with a concentrated load in the center of the beam, and Equation (45), is a simply supported beam with an uniform pressure. The Diana model would have a deflection between these two models. The b is equal to 450 mm and the h is 20 mm.

The results are shown in Figure 101. This shows that the Diana model lies between the two analytical calculations, and it is much closer to the calculation based on the point load. The pressure in the Diana model is only applied to 200 mm, while the distance between the supports is 670 mm. Thus, it is expected to be closer to the situation with the point load.

$$w = \frac{FL^3}{48EI} \tag{44}$$

$$w = \frac{5qL^4}{384EI} \tag{45}$$

Where:

- w is the deflection in the center of the beam $[mm]$,
- F is the point load $[N]$,
- q is the uniform pressure $[N/mm]$,
- L is the length of the beam $[mm]$,
- E is the Young's modulus $[N/mm^2]$,
- I is the moment of inertia, in this case equal to $\frac{1}{12}bh^3$ $[mm^4]$.

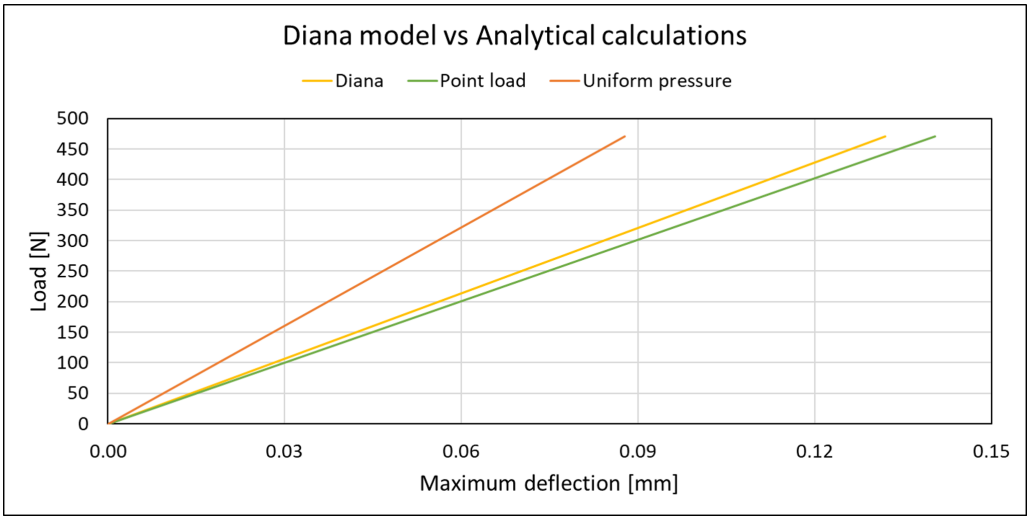


Figure 101: Comparison of the maximum deflection between Diana and Analytical calculations based on beam theory.

Third, the sandwich panel is compared to analytical calculations. The bending stiffness of the sandwich panel can be mainly attributed to the glass layers which are situated at the bottom and top. Therefore, the Diana results of the sandwich panel can be compared with an analytical calculation with a sort of I-beam. For simplicity, it is assumed that only the outer glass layers contribute to the moment of inertia and that the core

has no contribution to the moment of inertia. This means that only the Young's modulus of glass needs to be used. The moment of inertia can be calculated with Steiner's theorem or parallel axis theorem shown in Equation (46 [9]. The rule needs to be applied per glass layer which has an a of 9.5 mm, an A of 450 mm^2, and the I_c is equal to $\frac{1}{12} \cdot 450 \cdot 1^3$.

The results are shown in Figure 102. This shows that the order of magnitude of the maximum deflection is in line with the Diana models with the sandwich panels. These calculations give a stiffness of roughly 900 to 1450 N/mm. It has to be considered that these calculations are very simple and only give a really rough indication of the deflections.

$$I_p = I_c + a^2 A \tag{46}$$

Where:

- I_p is the moment of inertia along a parallel axis $[mm^4]$,
- I_c is the moment of inertia about its center of mass $[mm^4]$,
- a is the distance between the parallel axis and the center of mass $[mm]$,
- A is the area of the cross section $[mm^2]$.

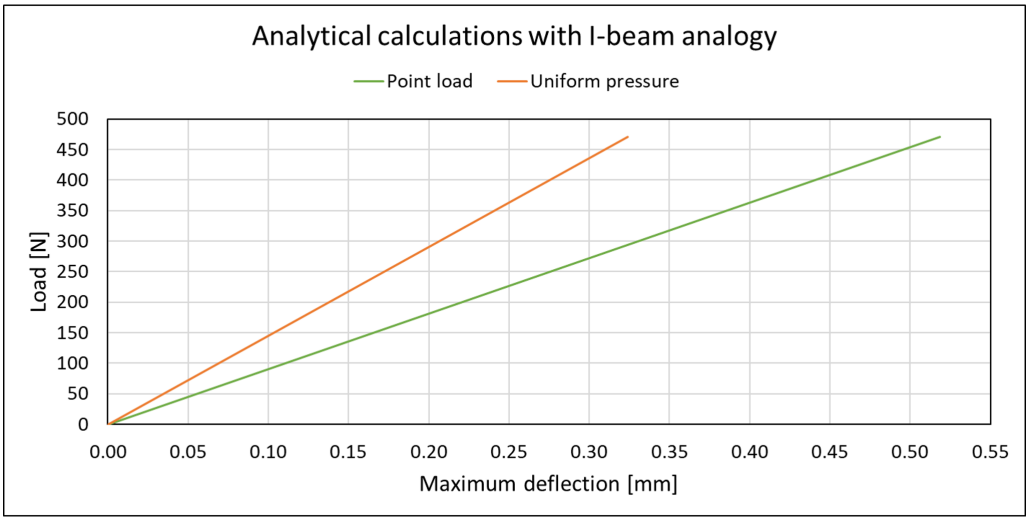


Figure 102: Load vs deflection of analytical calculations based on I-beam analogy.

13.6 Conclusion

Two different patterns are tested that only differ in their orientation of the honeycomb pattern. The experiments reveal that the rotated pattern can cope with a larger load than the non-rotated pattern, which reached a load of 48 and 40 kg, respectively. Next, the average stiffness of the rotated and non-rotated patterns is equal to 80 and 58 N/mm. Although these findings show that the rotated pattern is stiffer, a large uncertainty lies within the connection between the core and the glass layers. When comparing both panels, the implicit assumption is that both panels have the same connective strength in the glue layer, which is probably not the case. Furthermore, the non-rotated panel has a stiffer initial response to the load and then a significant reduction in its stiffness. This reduction is likely caused by a local failure in the glue. A more eminent failure in the connection is visible during the last added load of the non-rotated pattern.

The Diana models showed that the rotated pattern is responding stiffer than the non-rotated pattern, with a stiffness of about 708 and 674 N/mm respectively. This stiffness is approximately nine to twelve times as high as the observed stiffness in the experiments. When the maximum stiffness observed during the experiments is used, the stiffness in the Diana models is around 7.5 times as high.

Additional tests are performed to take a closer look at possible causes of the discrepancy. First, a Diana model with only the core is examined. This shows that the core itself would deflect by roughly 80 mm, which means that the core itself is not very stiff compared to the obtained results from the sandwich panel. Second, the boundary conditions are investigated. A solid glass panel of 710 x 450 x 20 is used. This reveals that the Diana model lies between the two analytical calculations based on the beam theory. Thus, the Diana models are in

line with the analytical calculations. Thereby, the Diana model which uses boundary interface elements and one that uses a line support, have the same maximum deflection. This indicates that the boundaries with boundary interface elements do not further stiffen the panel. Third, analytical calculations with a I-beam analogy are used to get a rough understanding of the order of magnitude of the expected deflections. This reveals that the order of magnitude of the deflections based on analytical calculations are in line with the Diana models of the sandwich panel.

The large difference between the Diana models and experiments is likely caused by the assumptions related to the glue layer. First, the glue layer, the connection between the glass and core is, in reality, an applied glue layer, while in the computer models the glue layer is not included and a perfect connection is assumed. Additionally, the core has one milled side and one side that is not. The milling process ensures a flat surface of the core where the adhesive is applied to enhance the connectivity between the glass and core. However, the other side has a concave surface, which reduces the area of the core that has a good connection with the glass [61]. The milled side is placed at the bottom of the panel. The analytical calculations with the I-beam analogy show if the sandwich panel would properly work as one united panel, the deflections should be much smaller than observed in the experiments. This can indicate that the assumption regarding the glue layer may be an important cause of the discrepancy.

Although the difference between the models may be partly caused by the glue layer, the difference in the representation of the boundary conditions and loading conditions can also play a part. However, the boundary conditions in the Diana model only have compressive reaction forces and no resistance to tension, which closely reassembles the boundary conditions in the experiment. Additionally, in the Diana models, the applied load is assumed to be a constant area load, while in reality, the load is concentrated in the placed cylinders. This may have a small effect on the deflection, but it is not likely to be the main reason because the added wooden plate between the panel and the cylinders helps to spread the load more evenly over the entirety of the wooden plate. Subsequently, the weight of the panel is disregarded in the computer model; however, the panels have a weight of 3.5 kg, which is no more than 10 % of the total applied load. Therefore, this could not be the main cause of the discrepancy.

14 Grid size determination daylight criterion

The daylight criterion is based on a calculated ratio between maximising the total sunlight at the winter solstice and minimising the direct sunlight during the summer solstice. These values are calculated with the use of the Ladybug library, where the incoming sunlight is measured through a mesh of sensors placed behind the panel. The ratio uses two calculation methods within Ladybug, where one only considers direct sunlight and the other considers direct and indirect sunlight. The goal of this chapter is to determine the relationship between the number of sensors used, the obtained sunlight values, and the computation time. Furthermore, a small study is conducted to understand the effect of small changes in the variables on the measured sunlight, as this determines the required accuracy needed to properly conduct an optimisation process with the shortest computation time.

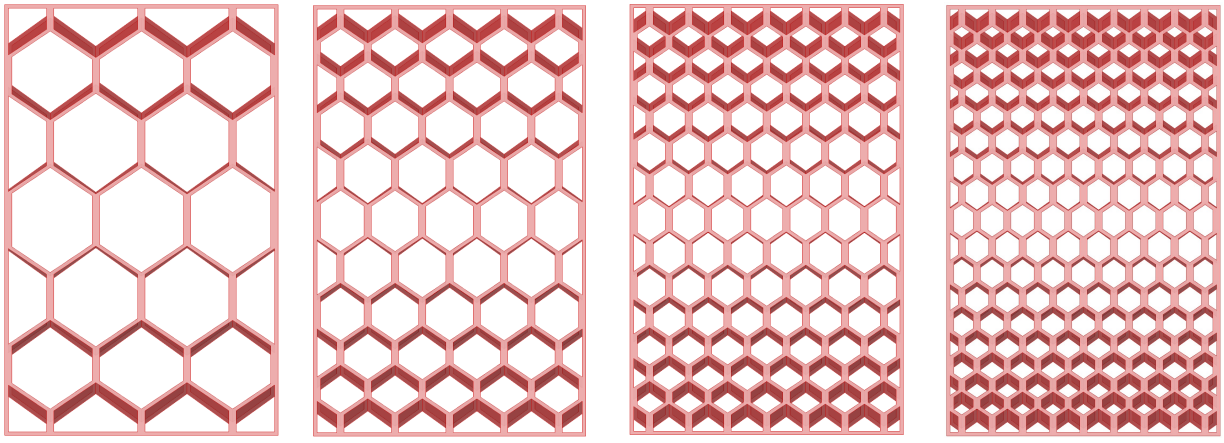
First, the different models used in this study are shown where their main difference is in the number of hexagons, because there is probably a relationship between the required number of sensors and the total number of hexagons in the core. Then, the effect of mesh changes is studied for the calculation method, which only considers direct sunlight. Subsequently, a similar process is performed for the calculation method, considering direct and indirect sunlight. Next, the effect of small changes within the variables on the calculated sunlight is examined. After gathering all this information, a conclusion is drawn, which determines the optimal mesh for both calculation methods.

14.1 Models

The determination of the mesh choice is based on four distinct core geometries, which are partly based on the normally orientated core geometry of the physical experiments. All three use the same value for the variables $Indiv_H$, P_H, A_c, and RP_{incl}, which are equal to 0.5, 1, 40, and 0.5, respectively. The differences between the models are based on the variables N_{hh} and N_{vh}, whose values are shown in Table 29. Additionally, the models are shown in Figure 103.

	Model 1	Model 2	Model 3	Model 4
N_{hh}	6	10	14	18
N_{vh}	6	10	14	18

Table 29: Overview of the different used models



(a) Model 1 (b) Model 2 (c) Model 3 (d) Model 4

Figure 103: A front view of the four core geometries

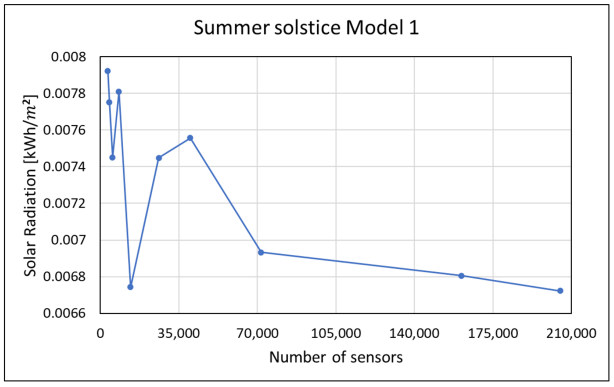
14.2 Daylight computation without reflection

The first calculation method does not use reflection, which means that only the direct sunlight during either the summer solstice or winter solstice is measured at the sensors. The applied grids are shown in Table 30, where the number of sensors lies between 3,200 and 205,000. For example, Grid 10 with 205,000 sensors means that as the panel is 710 by 450 mm, there is one sensor placed in the middle of an area equal to 1.56 mm^2, while in Grid 1, one sensor represents an area of 99.8 mm^2. Furthermore, the table presents the average computation time per grid. The computation time for each of the models can be found in Appendix G.

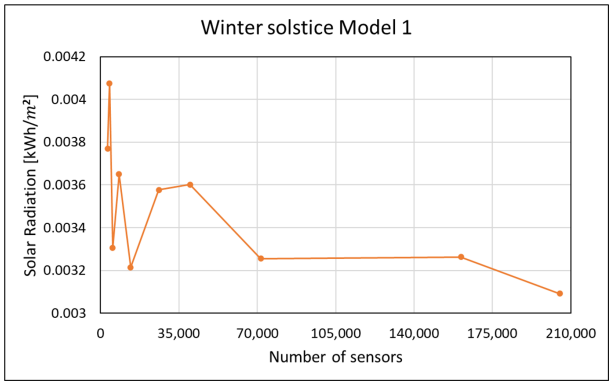
Thereafter, for each grid, the solar radiation at the summer and winter solstices is calculated and displayed in Figures 104, 105, and 106. This shows that every model shows relatively similar behaviour. As the number of sensors remains below 50,000, there is no clear indication of convergence. However, as the number of sensors exceeds 50,000 convergence occurs. For example, the maximum difference between the found solar radiation at 71,600 and 205,000 sensors is equal to 6 %, while the computation time increases by roughly 200 %.

	Number of sensors	Average computation time [s]
Grid 1	3,200	3.6
Grid 2	4,000	4.4
Grid 3	5,400	5.8
Grid 4	8,200	8.5
Grid 5	13,400	13.9
Grid 6	26,000	27.0
Grid 7	40,000	42.2
Grid 8	71,600	75.3
Grid 9	161,000	170
Grid 10	205,000	224

Table 30: Overview of used grid sizes.

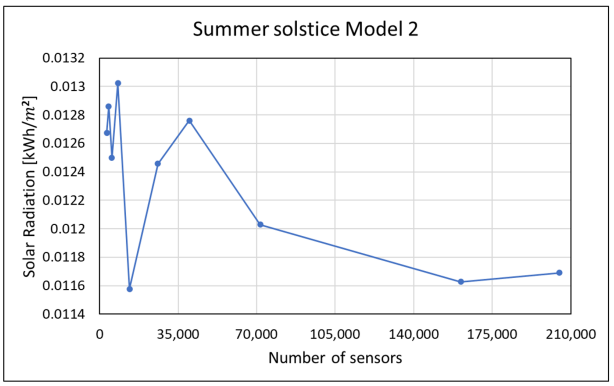


(a) Solar radiation at summer solstice

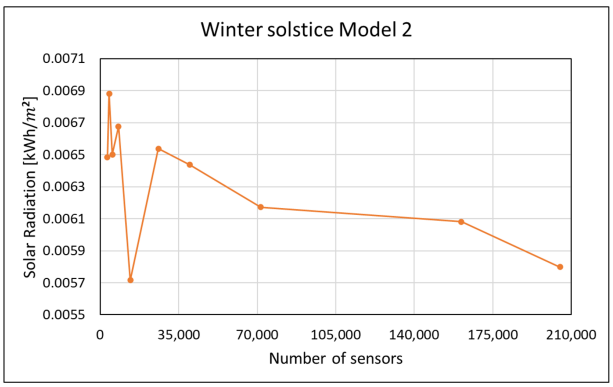


(b) Solar radiation at winter solstice

Figure 104: Relationship between number of sensors and solar radiation for Model 1

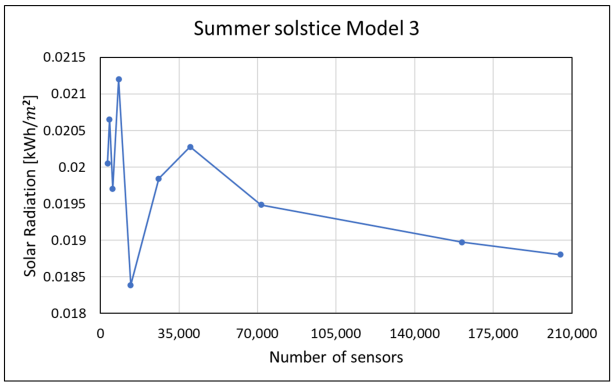


(a) Solar radiation at summer solstice

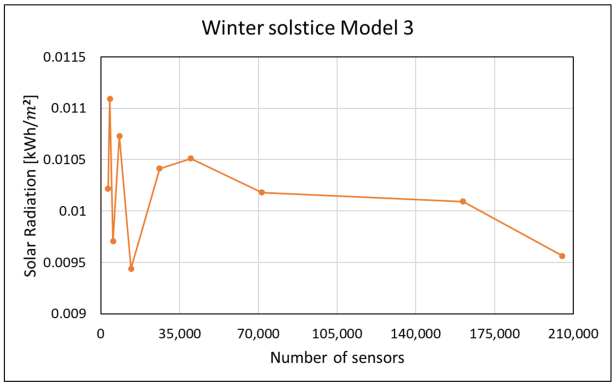


(b) Solar radiation at winter solstice

Figure 105: Relationship between number of sensors and solar radiation for Model 2

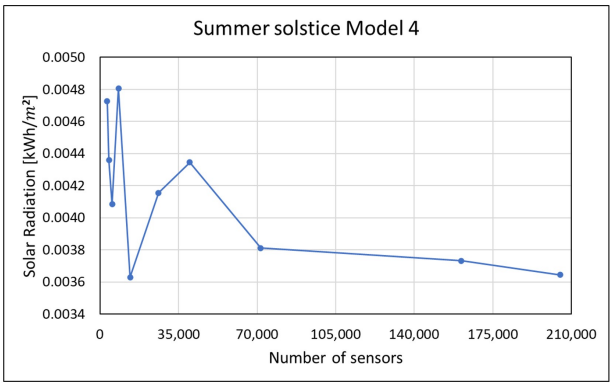


(a) Solar radiation at summer solstice

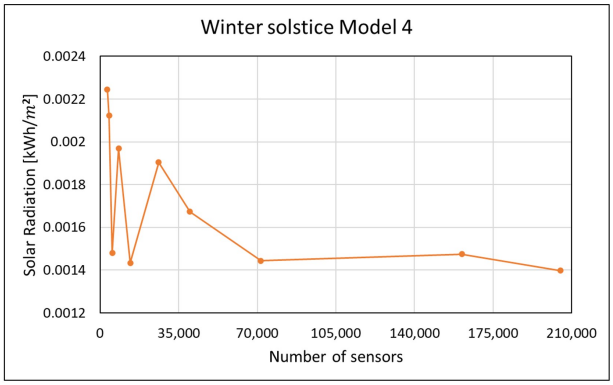


(b) Solar radiation at winter solstice

Figure 106: Relationship between number of sensors and solar radiation for Model 3



(a) Solar radiation at summer solstice



(b) Solar radiation at winter solstice

Figure 107: Relationship between number of sensors and solar radiation for Model 4

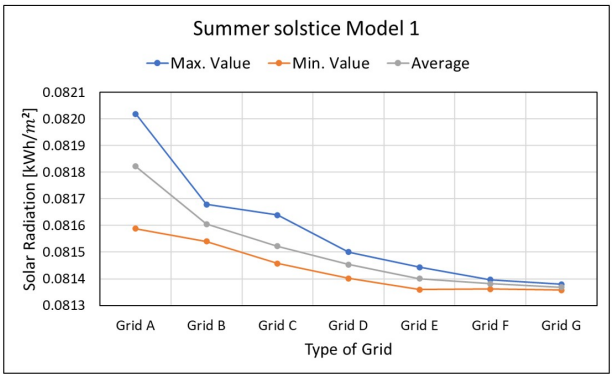
14.3 Daylight computation with reflection

The second part of the ratio is based on the daylight computation with reflection. This process uses a different Ladybug component, where only the number of sensors is a variable and all other aspects use the standard settings. The incorporating of reflection creates randomness in the model, which does not exist in the calculation without reflection. Thus, the randomness originates from the inclusion of reflection, which causes slightly different results if the same calculation is repeated. Therefore, for each of the grids stated in Table 31, the calculations are repeated thirty times. Although the number of repetitions is small, the findings give a basic understanding of the effects of the number of sensors and different numbers of hexagons in the geometry on the accuracy and magnitude of the random error.

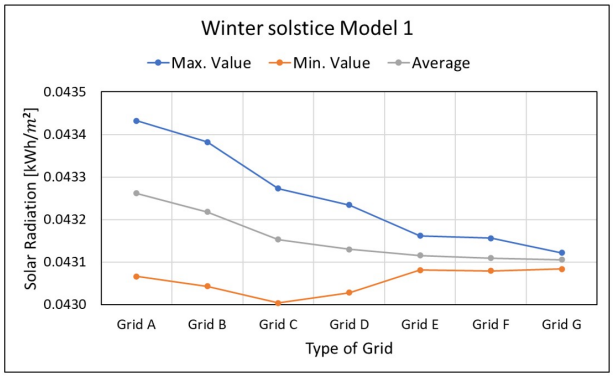
Figures 108, 109, and 110 provide some information on this randomness as well as the effect of an increase in sensors. For all models, the difference between the maximum and minimum diminishes as the number of sensors increases. Second, as the number of sensors increases the average results decrease and show convergence. Although there is convergence, the difference in the results is minimal, for example, the difference between Grid A and Grid G lies between 0.34 and 0.72 % while the computation time rapidly increases. Additionally, the maximum and minimum do not show clear convergence, which is probably caused by the small sample size.

	Grid A	Grid B	Grid C	Grid D	Grid E	Grid F	Grid G
Number of sensors	77	150	308	640	2183	4,928	19,824
Computation time Model 1 [s]	35.7	36.6	37.8	49.3	54	80.6	228
Computation time Model 2 [s]	40.3	44	45.8	54	65	98	276
Computation time Model 3 [s]	66.2	70.4	73.3	82.5	96	132	342
Computation time Model 4 [s]	144	156	162	168	192	222	456

Table 31: Overview of used grids and computation time

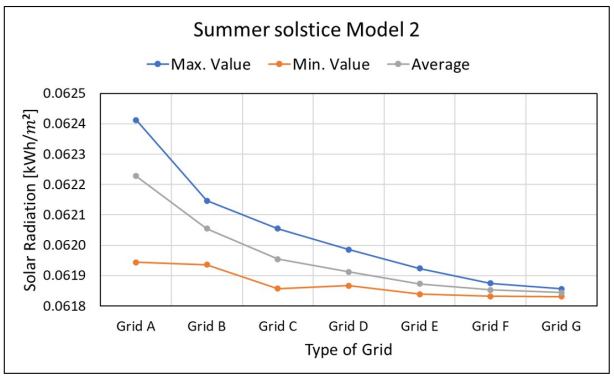


(a) Total solar radiation at summer solstice

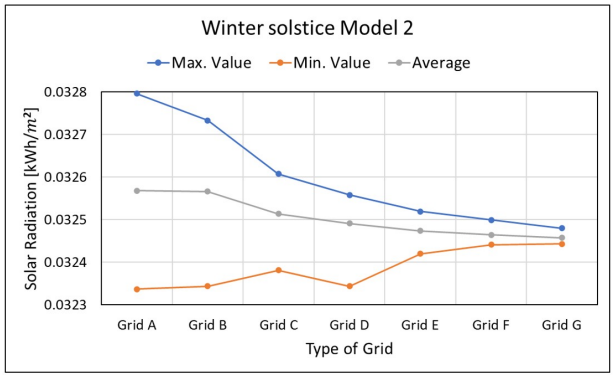


(b) Total solar radiation at winter solstice

Figure 108: Relationship between number of sensors and total solar radiation for Model 1

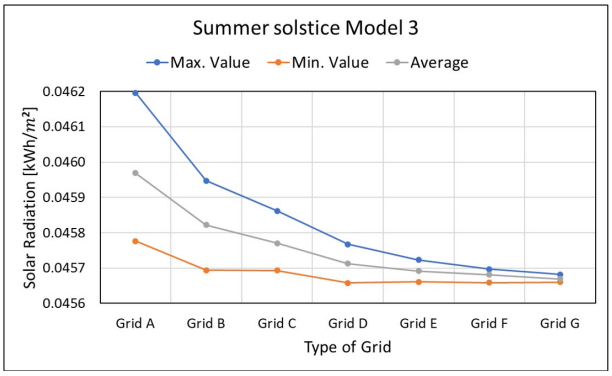


(a) Total solar radiation at summer solstice

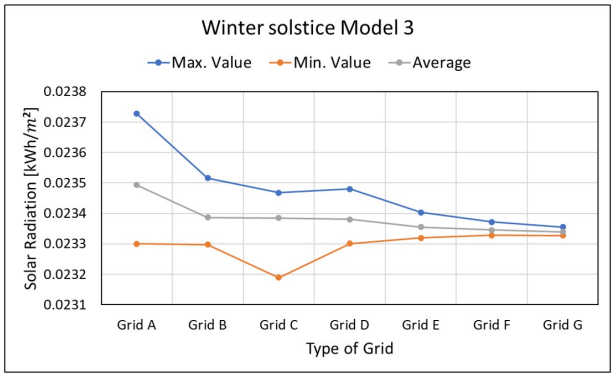


(b) Total solar radiation at winter solstice

Figure 109: Relationship between number of sensors and total solar radiation for Model 2

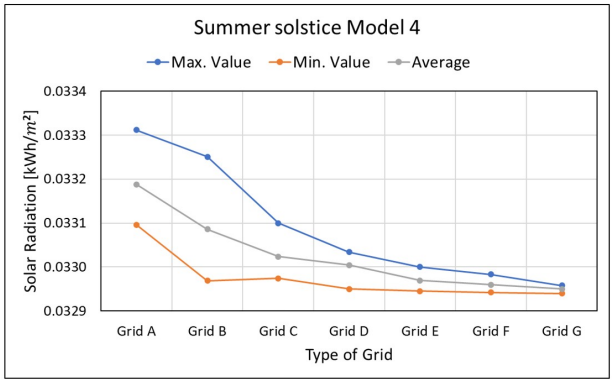


(a) Total solar radiation at summer solstice

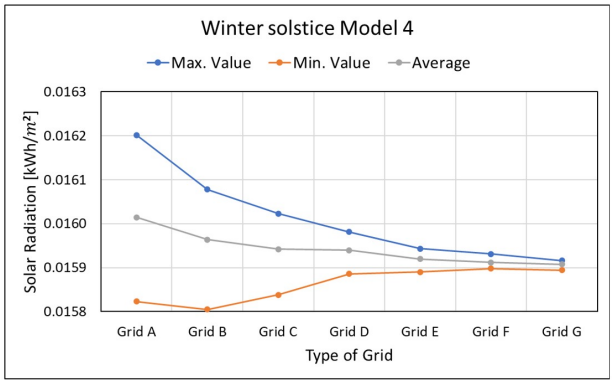


(b) Total solar radiation at winter solstice

Figure 110: Relationship between number of sensors and total solar radiation for Model 3



(a) Total solar radiation at summer solstice



(b) Total solar radiation at winter solstice

Figure 111: Relationship between number of sensors and total solar radiation for Model 4

Another important issue, as a result of inconsistent results, is drawing incorrect conclusions while comparing two relatively similar panels. For example, if two panels have an average difference of 2 % for the Daylight criterion, each average has a variance of 5 %, and the computation is done once, both panels can be seen as the best choice for the Daylight criterion. Thus, the number of sensors has to be chosen carefully in each specific case. Figure 112, shows the maximum absolute difference between the average and either the maximum or minimum as a percentage of the average for every model. This number is presented on the vertical axis, and the horizontal axis shows the number of sensors. The figure shows that, in general, as the number of hexagons in the geometry increases, the relative error size enlarges. Although it is not indubitable. Next, on average, the error size is larger at the winter solstice compared to the summer solstice. Finally, the exact numbers per model and grid size can be seen in Appendix G.

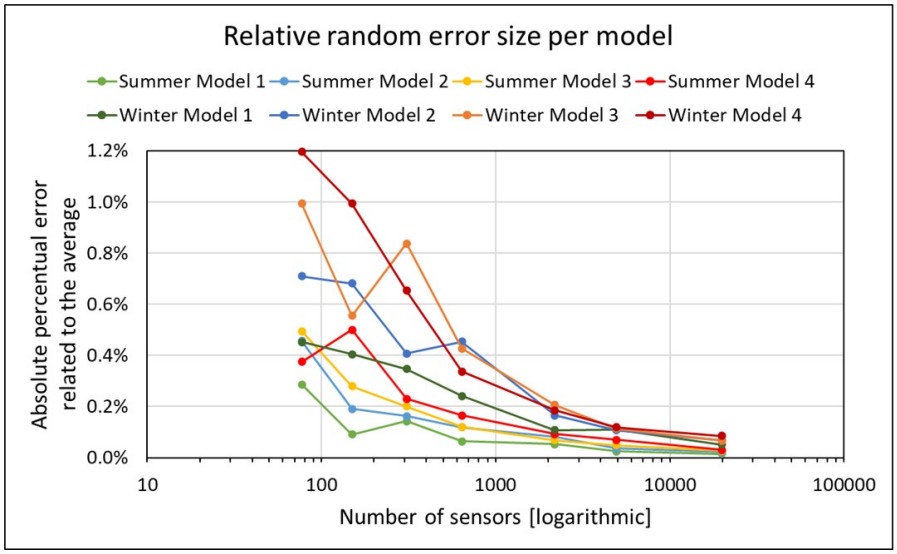


Figure 112: Percentage of random error to the average per model.

14.4 Sensitivity analysis of daylight calculation with reflection

As seen in the previous sub-chapter, the results of the daylight calculation with reflection have a small inconsistency. Hence, the number of sensors has to be partly based on the difference in the results obtained by similar geometries. This comparison uses all models as a reference, and the small changes are based on alterations to all variables except the N_{hh} and N_{vh}. Additionally, grid F is used to limit the error size and correctly compare the geometry alterations. Moreover, the different models are used to understand the effect of N_{hh} and N_{vh} on the magnitude of the error.

The error size of Model 1 with Grid G is 0.015 % and 0.051 % for the summer and winter, respectively. Table 32 shows the absolute difference between Model 1 and the similar geometries. The upper number in the second column is the solar radiation at the summer solstice, and the lowest number is the solar radiation at the

winter solstice. The smallest absolute differences are 0.10 % and 0.24 % in the summer and winter, respectively. These numbers are much larger than the error, so it can be concluded that the difference is mostly caused by the change in geometry. In model 2, the error size in Grid G equals 0.022 % and 0.069 % in the summer and winter, respectively, while the table shows the smallest changes are 0.10 and 0.55 %, thus the changes in the results are mainly caused by the differences in variables. Furthermore, the error size of Model 3 with Grid G is 0.028 and 0.067 % at the summer and winter solstices, respectively. The table shows that 0.27 and 0.82 % are the smallest changes caused by alterations in the variables. This change is also much larger than the random error with Grid G. Finally, in model 4, an error size of 0.031 and 0.085 % is presented, and the minimum change shown in the table is 0.26 and 0.83 % at the summer and winter solstices, respectively. In general, the value of the minimum change slightly increases as more hexagons are presented in the geometry.

	Model 1	Model 2	Model 3	Model 4
Domain 2 instead of 1	1.28 % 0.58 %	1.97 % 0.89 %	2.60 % 1.42 %	2.88 % 1.96 %
Domain 3 instead of 1	0.44 % 0.68 %	0.74 % 1.29 %	0.63 % 2.03 %	0.26 % 2.78 %
Reference point 0 instead of 0.5	3.50 % 2.20 %	6.69 % 4.40 %	11.11 % 7.52 %	16.63 % 11.52 %
Reference point 1 instead of 0.5	5.70 % 3.10 %	10.70 % 6.14 %	16.87 % 10.20 %	23.54 % 15.17 %
Angle 30 instead of 40	2.50 % 2.40 %	4.70 % 4.74 %	7.20 % 7.83 %	9.55 % 11.65 %
Ind. height change 0 instead of 0.5	0.10 % 0.24 %	0.10 % 0.55 %	0.27 % 0.82 %	1.32 % 0.83 %
Ind. height change -0.5 instead of 0.5	0.43 % 0.62 %	0.76 % 1.43 %	0.70 % 2.38 %	0.35 % 3.27 %

Table 32: Magnitude of result changes with small alterations core geometry in all Models.

14.5 Conclusion

In the case of the daylight calculation without reflection, it can be seen that convergence takes place when the number of sensors exceeds 50,000. Subsequently, the maximum difference between the results at 71,600 and 205,000 sensors is 6 %, however, the computation time increases by 200 %. Therefore, it is chosen to use 71,600 sensors because an increase in sensors has little effect on the outcome but heavily influences the computation time. Also, the behaviour of the convergence is relatively similar in all patterns; thus, the overestimation has little influence on the comparison of different geometries within the Daylight criterion.

Second, the number of sensors for the daylight calculation with reflection depends mainly on the relative error because the difference between Grid C and Grid G is not greater than 0.22 %, while Grid G uses 64 times more sensors and the computation time at least quadruples. The acceptable error size depends on the smallest possible change between two different geometries, which are considered in the optimisation model. Thus, this should be checked for each specific optimisation case to be able to draw the correct conclusions. In this case, the minimum change between very similar geometries equals 0.10 and 0.24 % for the summer and winter, respectively, in Model 1. This would mean that at least Grid D must be chosen, as the error size equals 0.064 and 0.241 % for the summer and winter, respectively. Although, preferably, a few more sensors need to be used, as in, for example, the most unfavourable situation, one geometry would have a result that is 0.064 % smaller than the average and the other 0.064 % larger. This concludes at 0.128 %, which is larger than 0.10 %. The same process shows that the error in case the of the winter solstice would also be too large. Therefore, in this case, using a little more sensors than Grid E would suffice and limit the computation time to a minimum. Subsequently, in Model 2, using the same reasoning, a grid size between Grids E and F needs to be used.

In the case of Model 3, the minimum change due to small alterations in the variables is equal to 0.27 and 0.82 % for the summer and winter solstices, respectively. This minimum change is much larger than the measured difference in Model 1. Subsequently, Grid D would almost already provide sufficient accuracy. Additionally, in Model 4, the smallest change is equal to 0.26 and 0.83 %, which means that a grid somewhere between Grid D and E is required. Thus, it appears that as the number of hexagons (vertically and horizontally) increases, the number of required sensors reduces slightly. Moreover, with regard to efficient use of computational power, this finding is beneficial as the overall daylight computation time increases as the number of hexagons increases.

15 Case study

A case study is performed to examine the behaviour of the optimisation model in a real-life implementation. The goal is to obtain a profound understanding of the characteristics of the model, such as the computing time, the convergence, and the sensitivity related to the orientation and magnitude of the load. These distinct aspects are discussed in the results and conclusion section. First, the characteristics of the facade element are specified, which are mainly the required input parameters, such as geometrical specifications. Subsequently, the characteristics of the optimisation process are given, which comprise the used weights, the variables and their domains, and the method for the sensitivity analysis. Then, the results of the optimisation models are explained, which are divided into the obtained optimal solution and the sensitivity analysis. Finally, conclusions are drawn based on the analysed results.

15.1 Framework

The case study is intended to be a real-life application; hence, the study is designed around the newly erected building, known as Echo, at the campus of the Delft University of Technology. Echo, as shown in Figure 113, was opened in the spring of 2022 and intended to be energy-positive. Thus, it is a well-suited option for the optimisation model, as the model strongly emphasises sustainability with the daylight and insulation criterion. The largest glass panels are 2.6 by 5 m and have a thickness up to 5.1 cm [4]. These panels weigh up to 1,200 kg and have an average U-value of 1.1 W/m^2K. The glass panels are supported at the bottom and top. The glass panels are designed to allow substantial sunlight, which limits the use of artificial light, and limit the solar radiation from heating the building, which is accomplished by an additional coating to optimise transparency and insulation.

Figure 113: The Echo building at the TU Delft campus [21]

15.1.1 Characteristics facade element

The general dimensions of the largest glass panel in Echo are used, which are equal to a width of 2.6 m, a height of 5 m, and a thickness of 5.1 cm. The overall thickness is divided into two glass sheets with a thickness of 2 mm, which is the upper limit of thin glass, and a core of 46.8 mm. The maximum thin glass thickness is selected because of the large size of the facade panel and the associated stress as a result of uniaxial bending. Moreover, roughly the same overall thickness is chosen to closely resemble the original situation. The same glass and core material are used in the case study as in the experiments. Additionally, the same line width and layer height are used. These values are shown in Table 33. The insulation properties are used from Brugman [25], where it is assumed that the outside and inside temperatures are equal to 21 degrees Celsius. Additionally, the same supporting condition is used; thus, a hinge support is located at the bottom and top of the panels. The orientation of the panel is assumed to be southwest, which is the longer side of the building, which is not visible

in Figure 113. This side has the main entrance and equals 67 degrees, where 90 degrees means west-facing and 0 degrees means south-facing.

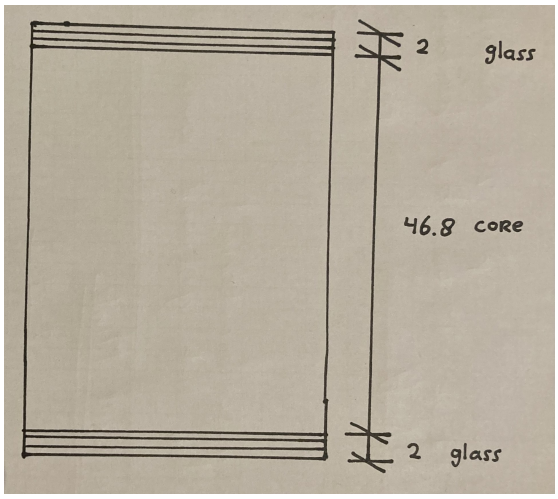
The wind load can be estimated based on Equation (1) in Chapter Structural Requirements. The $c_s c_d$ factor is equal to 1, as the panel width is smaller than 3 m. Additionally, the predominant wind direction in the city of Delft is west, which means that the panel is located in zone D. The zone determines the c_{pe} value, which is equal to +1.0, where the plus sign refers to a pressure force. Moreover, the q_p in Area II and at a height of no more than 7 m is equal to 0.58 kN/m^2. Thus, the wind pressure is equal to +0.58 kN/m^2. According to the NEN the SLS limits the maximum deflection to 50 mm as explained in Chapter 2.

Finally, the material used for the core is FGF PIPG Glassfiber (30%) [3]. This is the same material used in Pfarr and Louter [61]. It is chosen because of the higher strength and stiffness compared to the FGF PET-G used in the experiments, which is advantageous in a panel of this size.

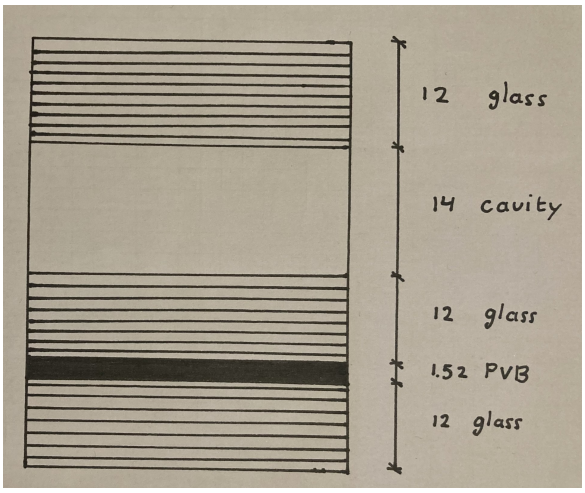
Symbol	**Short description**	**Definition**
W	Width panel	2600 $[mm]$
H	Height panel	5000 $[mm]$
T_c	Thickness core	46.8 $[mm]$
T_g	Thickness glass	2 $[mm]$
L_w	Core layer width	6 $[mm]$
L_h	core layer height	1.8 $[mm]$
λ_{air}	Thermal conductivity air	0.025 $[W/mK]$
λ_{core}	Thermal conductivity core	0.29 $[W/mK]$
λ_{glass}	Thermal conductivity glass	1.19 $[W/mK]$
α_{in}	Heat transmission coefficient inside	7.8 $[W/m^2K]$
α_{out}	Heat transmission coefficient outside	7.8 $[W/m^2K]$
$Load$	Uniform surface pressure	0.58 $[kN/m^2]$
$TB_{support}$	Support conditions Top and Bottom	The presence of top and bottom supports [-]
$LR_{support}$	Support conditions Left and Right	The absence of left and right supports [-]
$Young_{glass}$	Young's modulus glass	7000 $[kN/cm^2]$
$Shear_{glass}$	Shear modulus glass	3000 $[kN/cm^2]$
$Weight_{glass}$	Density glass	24.8 $[kN/m^3]$
Ts_{glass}	Tensile strength glass	80 $[kN/cm^2]$
Cs_{glass}	Compressive strength glass	80 $[kN/cm^2]$
$Young_{core}$	Young's modulus core material	202 $[kN/cm^2]$
$Shear_{core}$	Shear modulus core material	100 $[kN/cm^2]$
$Weight_{core}$	Density core material	12.7 $[kN/m^3]$
Ts_{core}	Tensile strength core material	5 $[kN/cm^2]$
Cs_{core}	Compressive strength core material	5 $[kN/cm^2]$
Or_{facade}	Orientation facade	67 [-]

Table 33: Overview of all the input parameters according to the classification made in Chapter 10.

The cross sections of the original panel as well as the sandwich panel are shown in Figure 114. This provides a good overview of the thickness of thin glass compared to a regular window. Furthermore, based on the cross section of the original panel and the density of 2500 kg/m^3 obtained from the NEN-EN2608, the weight is equal to 1170 kg [57]. This is similar to the weight reported by Octatube [4].



(a) Cross section of the sandwich panels with a total thickness of 50.8 mm.



(b) Cross section of the original panel with a total thickness of 51.52 mm. A single fully tempered glass layer of 12 mm is positioned at the outside, and a laminated layer of heat strengthened glass and 4 PVB intermediate sheets of 0.38 mm are positioned at the inside.

Figure 114: Cross sections.

15.2 Characteristics optimisation

The characteristics of the optimisation are described below. This covers the used variables and the Pareto front. Furthermore, tests are done for both orientations of the core geometry while using the exact same variables and Pareto front discritisation in order to compare them with one another.

15.2.1 Variables

As mentioned before, there are six variables that affect the final geometry of the core. The variables and their possible values are listed below. The variables are chiefly determined by providing the option to have small enough hexagons, which could be used in a real-life application, while roughly limiting the computation time to 12 hours. This limit is chosen in order to allow running up to two optimisations per day. The total number of possibilities is equal to 729, which is a small number for optimisation algorithms; however, the computation time needed for a single geometry already reaches 10 to 15 minutes.

$$\begin{aligned} N_{hh} &= 12, 16, 20 \\ N_{hw} &= 12, 16, 20 \\ Indiv_H &= -0.5, 0, 0.5 \\ P_H &= 1, 2, 3 \\ A_c &= 0, 22.5, 45 \\ RP_{incl} &= 0, 0.5, 1 \end{aligned} \tag{47}$$

15.2.2 Pareto front

The Pareto front consists of optimal solutions that originated from different weights. This provides an understanding of the effect of weight changes on the design and to what extent those changes extend. This can be helpful in the decision-making process. In this case, the weights vary between 0 and 1, while the sum of all weights is equal to 1. This domain is discretised into the 16 combinations shown in the table below. Moreover, the first three tests determine the near-optimal solution in cases where a single criterion is used, while test 14 searches for the near-optimal solution in cases where all criteria have equal weight. The other tests are a set of all possible combinations with a weight of 0.25, 0.50, or 0.75 per criterion. These tests provide insights into the sensitivity or insensitivity of the near-optimal solutions to weight alterations.

	W_T	W_I	W_D
Test 1	1.00	0	0
Test 2	0	1.00	0
Test 3	0	0	1.00
Test 4	0.75	0.25	0
Test 5	0.75	0	0.25
Test 6	0.25	0.75	0
Test 7	0	0.75	0.25
Test 8	0.25	0	0.75
Test 9	0	0.25	0.75
Test 10	0.50	0.50	0
Test 11	0.50	0	0.50
Test 12	0	0.50	0.50
Test 13	0.25	0.50	0.25
Test 14	0.333	0.333	0.333
Test 15	50	25	25
Test 16	25	25	50

Table 34: Overview of tests for the Pareto front

15.3 Optimisation results

The results of the optimisation models are discussed in this chapter. The results are divided based on the orientation of the hexagon pattern in the core geometry.

15.3.1 Results with normal orientation

Table 35 states the bandwidth in which all possible constraints lie with respect to the three criteria, and Table 36, and the first three tests in Table 37, show the designs of the most and least desired solutions per criterion. The bandwidth needs to be known upfront in order to determine the objective values for all other 13 tests. Subsequently, of all designs, the maximum deflection is 32.8 mm, which is smaller than the maximum allowed deflection of 50 mm, and the Von-Mises Unity-Check has a maximum of 0.103. When the safety factor of 2 is applied, the value of 0.206 is well below 1.00. This safety factor is applied because the Karamba3D results underestimate the stress as concluded in Chapter 12. The designs with the largest maximum deflection are, in general, the patterns with the least number of hexagons in both directions. Thus, all designs fulfill the structural constraints.

	Most desired value	Least desired value
Transparency Criterion; Area core [mm^2]	531,865	2,673,139
Insulation Criterion; U-value	0.63803	0.72467
Daylight Criterion	17.127	14.651

Table 35: Overview of extreme values per criterion

	N_{hh}	N_{hw}	$Indiv_H$	P_H	A_c	RP_{incl}
Least desired Transparency	20	20	-0.5	1	45	0
Least desired Insulation	20	20	0	-	0	-
Least desired Daylight	16	12	+0.5	1	45	1

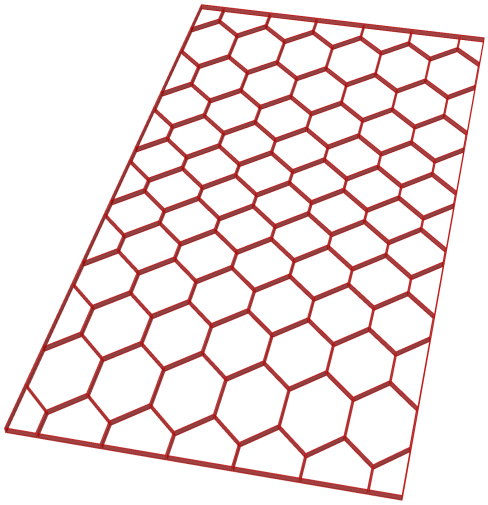
Table 36: Overview variables of the final results of the least desired solutions per criterion

Test	F_{final}	O_T	O_I	O_D	Weight [kg]	N_{hh}	N_{hw}	$Indiv_H$	P_H	A_c	RP_{incl}
Test 1	1.00	1.00	0.990	0.640	164	12	12	-0.5	1	0	-
Test 2	1.00	0.151	1.00	0.267	164	12	12	0	-	45	0.5
Test 3	1.00	0.802	0.615	1.00	170	20	12	-0.5	1	0	-
Test 4	0.998	1.00	0.990	0.640	164	12	12	-0.5	1	0	-
Test 5	0.913	1.00	0.990	0.651	164	12	12	0	-	0	-
Test 6	0.993	1.00	0.990	0.640	164	12	12	-0.5	1	0	-
Test 7	0.908	1.00	0.990	0.661	164	12	12	-0.5	3	0	-
Test 8	0.920	0.802	0.615	1.00	170	20	12	-0.5	1	0	-
Test 9	0.904	0.802	0.615	1.00	170	20	12	-0.5	1	0	-
Test 10	0.995	1.00	0.990	0.640	164	12	12	-0.5	1	0	-
Test 11	0.901	0.802	0.615	1.00	170	20	12	-0.5	1	0	-
Test 12	0.826	1.00	0.990	0.661	164	12	12	-0.5	3	0	-
Test 13	0.910	1.00	0.990	0.661	164	12	12	-0.5	3	0	-
Test 14	0.883	1.00	0.990	0.661	164	12	12	-0.5	3	0	-
Test 15	0.913	1.00	0.990	0.661	164	12	12	-0.5	3	0	-
Test 16	0.856	0.805	0.824	0.897	167	16	12	0	-	0	-

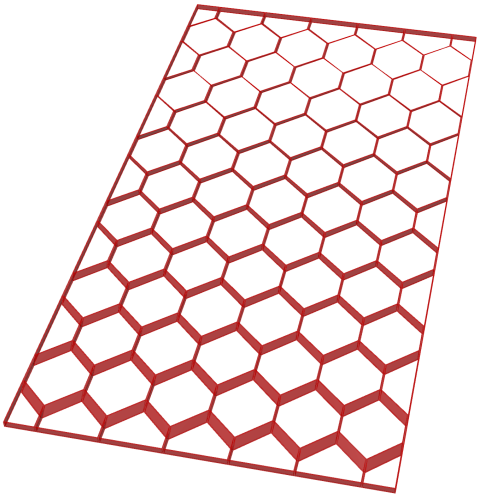
Table 37: Overview tests Pareto front

The first test, which only considers the transparency criterion and is shown in Figure 115a, already shows that the near-optimal transparency design also almost optimises for the insulation criterion, which has a score of 0.99. However, test 2, shown in Figure 115b, reveals that the design with an insulation score of 1.00 results in a transparency score of 0.151. Thus, although the criteria roughly strive for the same kind of design, the truly near-optimal design of the insulation criterion performs poorly with respect to transparency. This can be seen when looking into the design variables, which have the same number of hexagons but show a sharp contrast in the inclination. Next, both the transparency and insulation criteria prefer a solution with the minimum number of hexagons. When considering the daylight criterion, test 1 shows a fairly okay daylight score, while test 2 has a really worse daylight score and is also the lowest daylight score of all sixteen tests. This reveals the detrimental effect of inclination in general, as test 2 as well as the least desired daylight design both have the maximum inclination. Additionally, test 3, displayed in Figure 116a shows that a maximum number of hexagons horizontally while minimising the hexagons vertically is the most optimal design for the daylight criterion. The transparency score of test 3 is relatively high, and the insulation score is slightly above average. Therefore, these three tests show that especially the insulation criteria favours a solution with low transparency and daylight score, while the other two criteria steer to less detrimental designs for the other, not optimised for, criteria. Also, test 1 has the highest average score when all three criteria are considered, which already gives an idea of the design choice in test 14, where all criteria are equally considered. The design of test 14 is shown in Figure 116b. The design only differentiates from test 1 in the P_H variable, which is related to the factorial change of the height of hexagons per row. This change results in a slightly higher daylight score. Additionally, it appears that the other two criteria are not affected; however, it has to be noted that the numbers are rounded to three significant numbers.

Moreover, as the number of hexagons horizontally is maximised by the daylight criterion while the other two criteria minimise it, this number gives a good indication of the influence of the daylight criterion on the final design. The only designs with N_{hh} greater than 12 are tests 8, 9, 11, and 16. These tests have a daylight weight of at least 0.50 and an insulation weight equal to or smaller than 0.25. This means that the transparency score is less affected by the increase in N_{hh} than the insulation score, which also became apparent when looking at test 3. Subsequently, when looking at the tests that consider at least two criteria, the N_{vh}, A_c, and RP_{incl} always have the same values; the N_{hh} is equal to 12 in 8 out of the 13 tests; the $Indiv_H$ is -0.5 in 11 out of 13 tests; and the P_H changes mostly between 1 and 3. Thus, only three variables are affected by the weight changes, and N_{hw} and $Indiv_H$ are equal to 12 and -0.5, respectively, in more than half of the tests. Another interesting observation is that, beside the first three tests, the criterion score is at least 0.615 and the fitness score is at least 0.826, which means that in all cases, all objectives have an above-average score. The weight of the panels is between 164 and 170 kg, which is around 14 % of the glass panel's weight used in the Echo building. The glass sheets weigh 131 kg together, and the core weighs 33 to 39 kg.

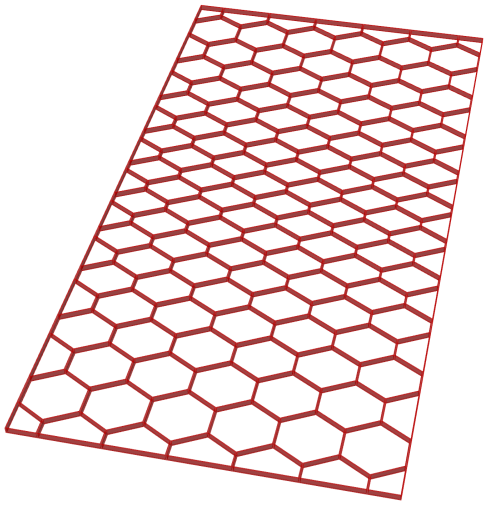


(a) Test 1, most optimal for transparency criterion with an objective score of 1.00 for transparency, 0.99 for insulation and 0.64 for daylight.

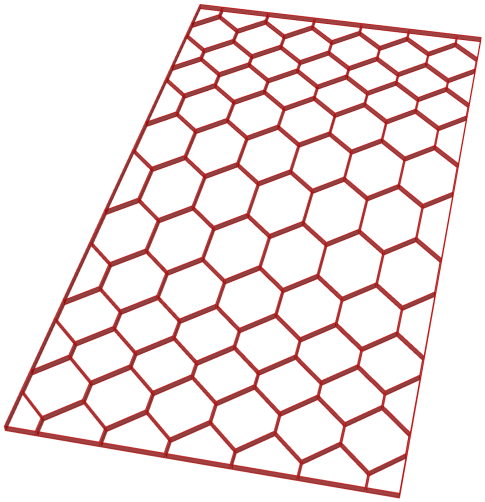


(b) Test 2, most optimal for insulation criterion with an objective score of 0.151 for transparency, 1.00 for insulation and 0.267 for daylight.

Figure 115: Overview of designs normal orientation.



(a) Test 3, most optimal for daylight criterion with an objective score of 0.802 for transparency, 0.615 for insulation and 1.00 for daylight.



(b) Test 14, most optimal with equal weight for all criteria with a fitness score of 0.883 and an objective score of 1.00 for transparency, 0.99 for insulation and 0.661 for daylight.

Figure 116: Overview of designs normal orientation.

15.3.2 Results with rotated orientation

Table 38 states the bandwidth in which all possible constraints lie with respect to the three criteria, and Table 39 and the first three tests in Table 40 show the designs of the most and least desired solutions per criterion. The bandwidth needs to be known upfront in order to determine the objective values for all other 13 tests. Subsequently, of all designs, the maximum deflection is 34.4 mm, which is smaller than the maximum allowed deflection of 50 mm, and the Von-Mises Unity-Check has a maximum of 0.056. When the safety factor of 2 is applied, the value of 0.112 is well below 1.00. The designs with the largest maximum deflection are, in general, the patterns with the least number of hexagons in both directions. Thus, all designs fulfill the structural constraints.

	Most desired value	Least desired value
Transparency Criterion; Area core [mm^2]	556,813	2,121,753
Insulation Criterion; U-value	0.6448	0.7360
Daylight Criterion	15.694	13.403

Table 38: Overview of extreme values per criterion

	N_{hh}	N_{hw}	$Indiv_H$	P_H	A_c	RP_{incl}
Least desired Transparency	20	20	-0.5	3	45	0
Least desired Insulation	20	20	+0.5	1	0	-
Least desired Daylight	12	20	-0.5	1	45	0

Table 39: Overview variables of the final results of the least desired solutions per criterion

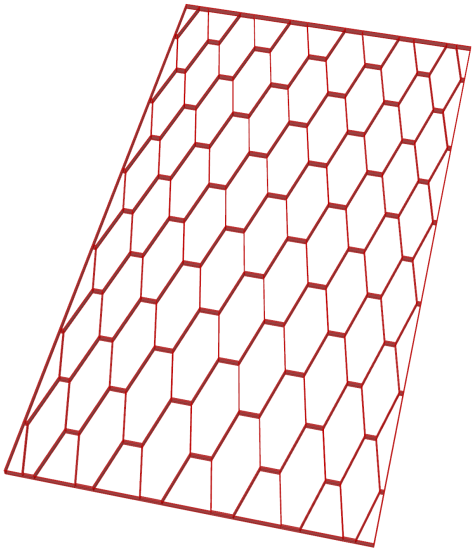
Test	F_{final}	O_T	O_I	O_D	Weight [kg]	N_{hh}	N_{hw}	$Indiv_H$	P_H	A_c	RP_{incl}
Test 1	1.00	1.00	0.981	0.840	165	12	12	+0.5	1	0	-
Test 2	1.00	0.254	1.00	0.648	165	12	12	-0.5	1	45	0.5
Test 3	1.00	0.825	0.685	1.00	170	20	12	+0.5	1	0	-
Test 4	0.995	1.00	0.981	0.697	165	12	12	+0.5	2	0	-
Test 5	0.963	1.00	0.981	0.853	165	12	12	0	-	0	-
Test 6	0.986	1.00	0.981	0.697	165	12	12	+0.5	2	0	-
Test 7	0.949	1.00	0.981	0.853	165	12	12	0	-	0	-
Test 8	0.945	0.825	0.685	0.985	170	20	12	-0.5	2	0	-
Test 9	0.910	0.825	0.685	0.985	170	20	12	-0.5	3	0	-
Test 10	0.991	1.00	0.981	0.697	165	12	12	+0.5	2	0	-
Test 11	0.927	1.00	0.981	0.853	165	12	12	0	-	0	-
Test 12	0.917	1.00	0.981	0.853	165	12	12	0	-	0	-
Test 13	0.954	1.00	0.981	0.853	165	12	12	0	-	0	-
Test 14	0.944	1.00	0.981	0.853	165	12	12	0	-	0	-
Test 15	0.959	1.00	0.981	0.853	165	12	12	0	-	0	-
Test 16	0.922	1.00	0.981	0.853	165	12	12	0	-	0	-

Table 40: Overview tests Pareto front

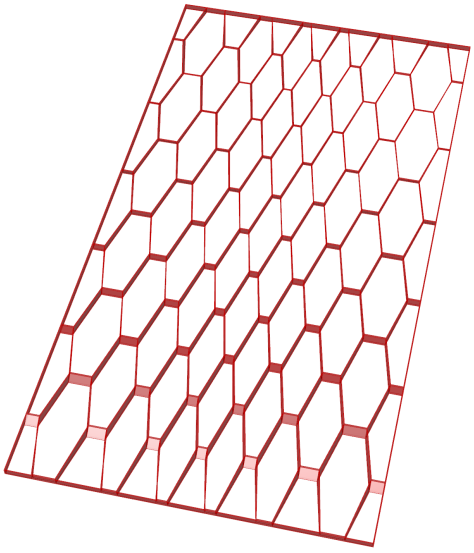
First, the near-optimal solution, if only transparency is considered, requires the minimum number of hexagons and no inclination. This is shown in Table 40 and in Figure 117a, and the other two criteria have a score of at least 0.84. This means that the near-optimal solution for the transparency criteria is also a good solution for the other two criteria. This cannot be said about the insulation criterion, which is optimised for in test 2, as shown in Figure 117b. This optimal solution differs mainly in the inclination compared to test 1, and as a result, the transparency score is only 0.254 in test 2. Thus, inclination has an enormous effect on the transparency score. The daylight score is 0.648, which is still a relatively good score, although it is significantly smaller than in test 1. Moreover, test 3 optimises the design with regard to solely the daylight criterion, and the design is shown in Figure 118a. Thereby, test 3 has the maximum number of hexagons horizontally, while in tests 1 and 2, the minimum number was used. This shows a clear distinction in the optimal design for the daylight criterion in contrast to the other two criteria. Furthermore, the insulation criterion is the only criterion that favours inclination (A_c), and although the number of hexagons is increased by eight horizontally in test 3, the transparency score is still 0.825. This indicates that the transparency score is only slightly affected by the change in N_{hh} and is primarily influenced by the inclination. Then, if all criteria are considered equally (test 14), a near-optimal design is found with the minimum number of hexagons, no factorial change in height of the hexagons, and no inclination. The design is shown in Figure 118b. The minimum number of hexagons indicates the effect of the insulation and transparency criteria, while the absence of inclination is caused by the transparency and daylight criteria. This shows that in both cases, the near-optimal transparency design is almost not compromised. As a result, the transparency score is equal to 1.00, and with a fitness score of 0.944 and a minimum objective score of 0.853, the final design performs very well on all three criteria.

The other tests, except for tests 1, 2, 3, and 14, reveal that only N_{hh}, $Indiv_H$, and P_H are changing in the final solution. Additionally, N_{hh} is equal to 12 in 10 out of the 12 tests, and $Indiv_H$ and P_H are equal to

0 and not relevant in 7 out of 12 tests. This shows the limited difference between the final designs found in all tests, and all criteria are not really compromised. For example, the N_{hh} is only equal to 20 if the daylight criterion has a weight of 0.75. Subsequently, in all tests except the first three, the transparency score is 0.825, the minimum insulation score is 0.685, and the daylight score is not smaller than 0.697; thus, the transparency criterion has on average the highest score, and the insulation and daylight criterion have an equal minimum, but on average the insulation score is much higher. The weight of the panels is between 165 and 170 kg, which is around 14 % of the glass panel's weight used in the Echo building. The glass sheets weigh 131 kg together, and the core weighs 34 to 39 kg.

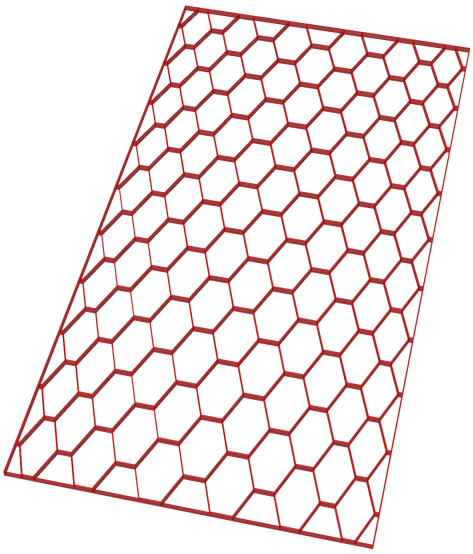


(a) Test 1, most optimal for transparency criterion with an objective score of 1.00 for transparency, 0.981 for insulation and 0.84 for daylight.

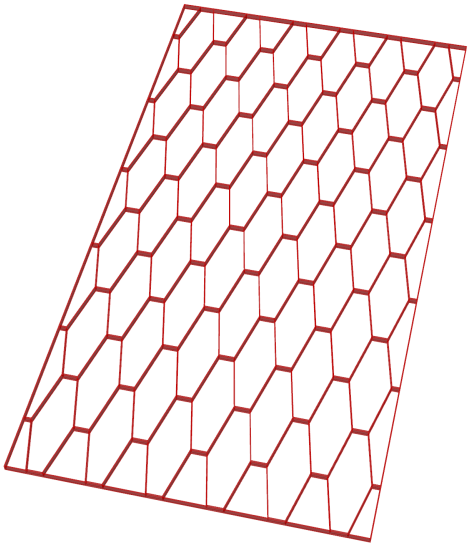


(b) Test 2, most optimal for insulation criterion with an objective score of 0.254 for transparency, 1.00 for insulation and 0.648 for daylight.

Figure 117: Overview of designs rotated orientation.



(a) Test 3, most optimal for daylight criterion with an objective score of 0.825 for transparency, 0.685 for insulation and 1.00 for daylight.



(b) Test 14, most optimal with equal weight for all criteria with a fitness score of 0.944 and an objective score of 1.00 for transparency, 0.981 for insulation and 0.853 for daylight.

Figure 118: Overview of designs rotated orientation.

15.4 Sensitivity analysis

The sensitivity analysis is used to better understand the robustness of the model and the designs found. This is tested through two input parameters: the orientation of the facade panel with respect to the daylight criterion and the applied load. The effect of different orientations on the final design and the robustness of these designs in other orientations are examined. The applied load affects the structural constraint; and currently, all designs meet this constraint, however, this load is increased in steps to understand the critical load and sensitivity of the panels toward changes in the applied load.

15.4.1 Optimal pattern with changing facade orientation

In total, five different orientations are used, which are: east, southeast, south, southwest, and west. These orientations are chosen because they face the sun during different stages of the day, which affects the angle of the incoming sunlight. The other orientations are not used because of the limited direct sunlight reaching through the facade panel.

The near-optimal designs of the patterns with normal orientation are shown in Table 41. This reveals that in almost all cases, all variables have the same value except for N_{vh}, which has a higher value in the south and southwest orientations and the P_H changes in the east. Thus, if the sun is at its highest stage during the day, an increased number of hexagons over the width is preferred. Next, Table 42 shows the least-favoured designs with regard to the daylight criterion. The main difference between the most and least preferred solutions is the inclination, and in all orientations, the N_{vh} is 20. Furthermore, all near-optimal solutions are used in all orientations, and their daylight score is determined, which is shown in Figure 119. This gives an indication of the robustness of using a single design in all orientations and shows that the most optimal solution for the south orientation is the least robust design, as it does not perform well in the other orientations. The near-optimal solutions found in the east, southeast, and West are all really robust and have a daylight score of at least 0.85 in all orientations, but the near-optimal solution of the southeast and west slightly outperforms the design found for the east direction. This pattern is shown in Figure 121a.

Orientation	east	southeast	south	southwest	west
Facade Design	NR2012A	NR2012B	NR2020	NR2016	NR2012B
Weight [kg]	170	170	181	175	170
N_{hh}	20	20	20	20	20
N_{vh}	12	12	20	16	12
$Indiv_H$	-0.5	-0.5	-0.5	-0.5	-0.5
P_H	3	1	1	1	1
A_c	0	0	0	0	0
RP_{incl}	-	-	-	-	-

Table 41: Overview of the results of the pattern (normal orientation) with different orientations, most optimal.

Orientation	east	southeast	south	southwest	west
Weight [kg]	181	181	181	181	181
N_{hh}	20	20	20	20	20
N_{vh}	20	20	20	20	20
$Indiv_H$	-0.5	-0.5	-0.5	-0.5	-0.5
P_H	2	2	2	2	2
A_c	45	45	45	45	45
RP_{incl}	1	1	1	1	1

Table 42: Overview of the results of the pattern (normal orientation) with different orientations, least optimal.

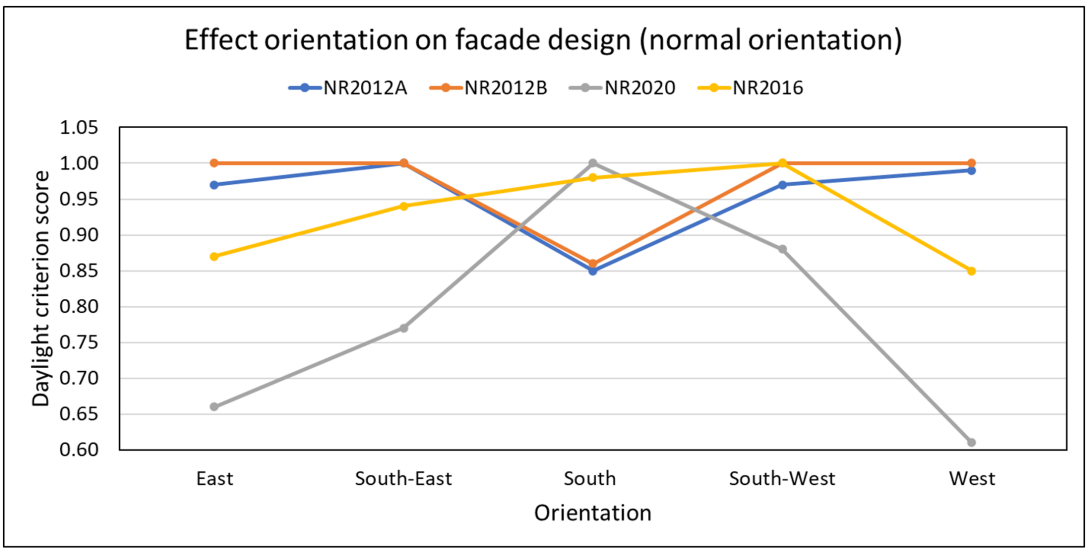


Figure 119: Comparing robustness designs in all orientations.

The same procedure is performed with the rotated hexagon pattern, and the near-optimal solutions of all orientations are shown in Table 43. This also shows that in the south direction, a larger value for N_{vh} is preferred. Subsequently, the other two variables that change in the final designs are $Indiv_H$ and P_H, while N_{hh}, A_c, and RP_{incl} do not change at all. In total, three different designs were found. Next, the least optimal designs of each orientation are determined and mainly differ from the near-optimal solution in their value of N_{vh} and the inclination, A_c. The three near-optimal designs are used in all orientations, and their daylight score is calculated and shown in Figure 120. This depicts that the near-optimal solution for the south orientation does not perform well in the other orientations, while the other two designs perform very well in all orientations. The near-optimal solution for the east, southeast, and west performs the best overall. This design is shown in Figure 121.

Orientation	east	southeast	south	southwest	west
Facade Design	R2012C	R2012C	R2020	R2012D	R2012C
Weight [kg]	170	170	184	170	170
N_{hh}	20	20	20	20	20
N_{vh}	12	12	20	12	12
$Indiv_H$	+0.5	+0.5	-0.5	-0.5	+0.5
P_H	1	1	2	3	1
A_c	0	0	0	0	0
RP_{incl}	-	-	-	-	-

Table 43: Overview of the results of the pattern (rotated orientation) with different orientations, most optimal.

Orientation	east	southeast	south	southwest	west
Weight [kg]	184	179	170	181	181
N_{hh}	20	12	20	16	16
N_{vh}	20	20	12	20	20
$Indiv_H$	-0.5	-0.5	+0.5	-0.5	-0.5
P_H	1	3	3	1	1
A_c	45	45	45	45	45
RP_{incl}	0	0	0	0	0

Table 44: Overview of the results of the pattern (rotated orientation) with different orientations, least optimal.

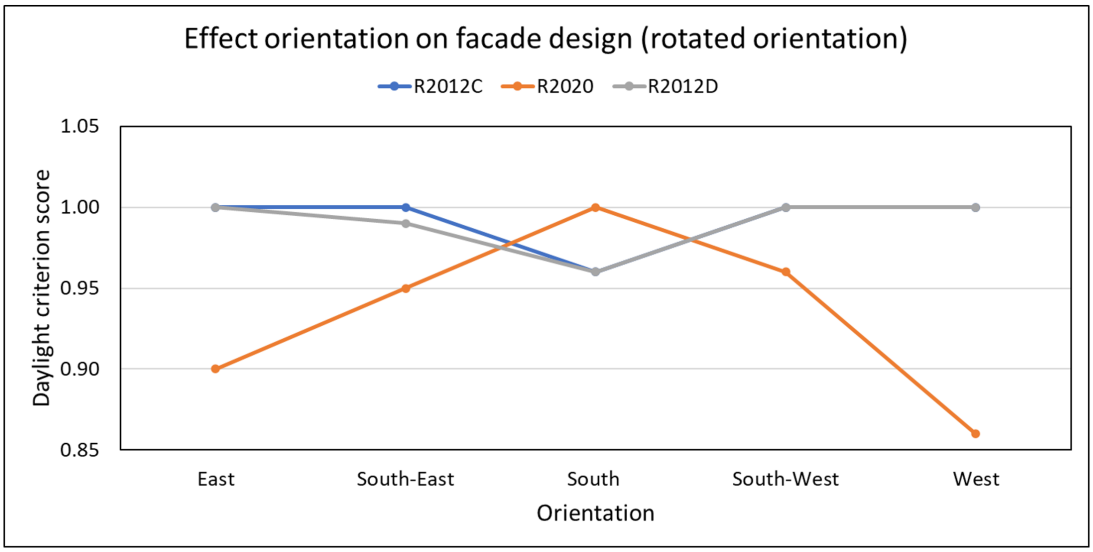
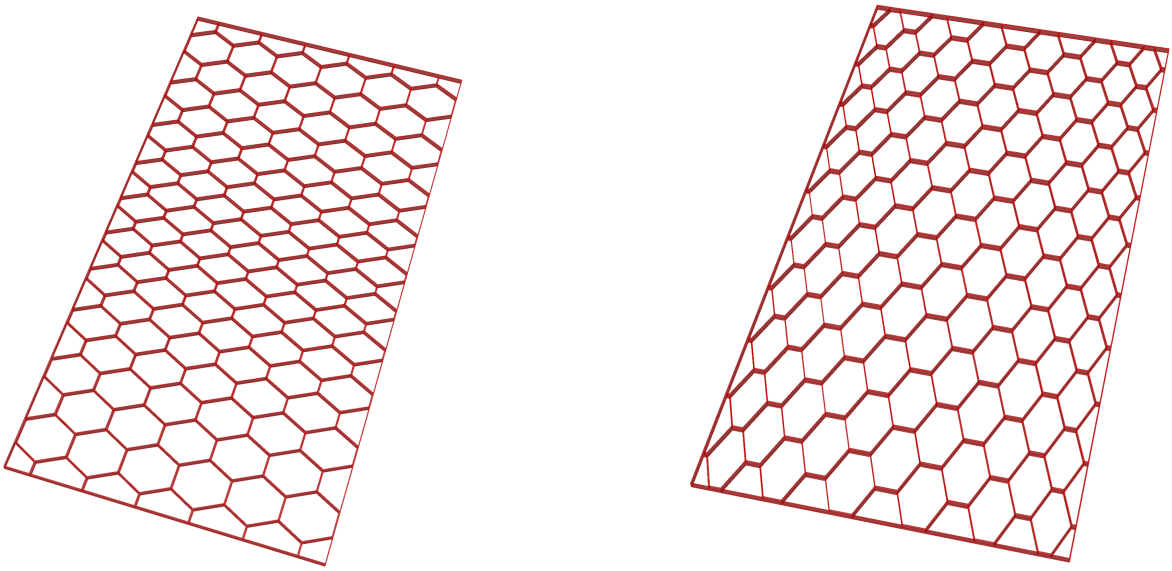


Figure 120: Comparing robustness designs in all orientations.



(a) Most robust design normal orientation, NR2012B

(b) Most robust design rotated orientation, R2012C

Figure 121: Overview of designs.

15.4.2 Change in the magnitude of the load

The external load is an important aspect of the determination process of the final design, as it can exclude certain designs from being used because of their lack of structural safety. Therefore, the magnitude of the load is increased in steps in order to investigate its influence on the possible designs. Additionally, currently, the magnitude of the load is small enough and does not prevent any of the possible designs from being classified as structurally unsafe. Thus, it is also examined at which load the weakest design no longer fulfills the structural requirement. Also, N_{vh} and N_{hh} are important variables within the optimisation process, and they are important variables regarding the deflection and Unity-Check. So, the effects of different values for N_{vh} and N_{hh} are considered.

First of all, the rotated patterns in Table 45 shows multiple facade designs and their respective maximum deflection and Unity-Check. The table displays the weakest design first and the strongest design last, where the designs in between go from weak to strong with respect to the deflection. The weakest design has the minimum number of hexagons, a $Indiv_H$ of -0.5, a P_H of 1, and no inclination. This pattern has a maximum deflection of 33.85 mm, which is below the allowed maximum deflection of 50 mm. Then, a higher number of N_{vh} results

in stiffer panels than an equally large increase in N_{hh}. Furthermore, the strongest design has a N_{hh} of 12 and a N_{vh} of 20, although the pattern with the maximum number of hexagons has a very similar deflection. Also, in the strongest design, the $Indiv_H$ is 0.5, and it has an inclination of 22.5 degrees. Thus, a pattern with an inclination is stiffer than a pattern without an inclination. The Unity-Check has, in general, a larger value with higher deflection; however, this is not always the case, as can be seen in the table. The Unity-Check values are all smaller than 0.12, which means that the maximum deflection is the limiting factor.

Additionally, the load and deflection are linearly correlated, which means that if the load increases by 25 %, the maximum deflection also increases by 25 %. Therefore, the maximum allowable load per design can be easily calculated, which is shown in the last row of the table. This shows that as the load is increased by slightly more than 46 % (0.85 kN/m^2), the first pattern is structurally unsafe. In the Pareto front, multiple designs have the minimum number of hexagons both vertically and horizontally, which means that in all these cases, a different near-optimal design may be found. Then, as the load increases by 56 % (0.905 kN/m^2), all possible designs within the variable domains are structurally unsafe. Thus, with the current variables, the designs all have roughly the same stiffness.

N_{hh}	12	16	20	12	16	12	20	12
N_{vh}	12	12	12	16	16	20	20	20
$Indiv_H$	-0.5	-0.5	-0.5	-0.5	-0.5	-0.5	-0.5	0.5
P_H	1	1	1	1	1	1	1	1
A_c	0	0	0	0	0	0	0	22.5
RP_{incl}	-	-	-	-	-	-	-	0
Maximum deflection [mm]	34.36	34.08	33.85	33.02	32.96	32.39	32.38	31.96
Unity-Check	0.112	0.113	0.090	0.116	0.091	0.076	0.078	0.054
Maximum allowable load [%]	146	147	148	151	152	154	154	156

Table 45: Overview of numerous designs and their maximum deflection and Unity-Check of the rotated pattern

Second, the non-rotated pattern, multiple designs are shown in Table 46. The table is ordered from the weakest to strongest design with respect to the maximum deflection. This shows that a maximum number of hexagons horizontally and a minimum number of hexagons vertically is the weakest design, with a maximum deflection of 33.01 mm. The other variables have the same value as the weakest design in the rotated pattern. Then, in general, the number of hexagons horizontally has a more negative effect on the maximum deflection than the number of hexagons vertically, which was also the case at the rotated pattern. The strongest design has a N_{hh} of 12, a N_{vh} of 20, and different values for the last three variables compared to the weakest design. This also shows the positive effect of an inclination on the maximum deflection. Furthermore, the Unity-Check value roughly decreases as the maximum deflection reduces, and it is not larger than 0.21. Thus, the maximum deflection is the critical criterion for structural safety.

Finally, the relation between the load and deflection remains linear, and as the load increases by 51 % to 0.88 kN/m^2, the weakest design no longer fulfills the structural constraint. At a load of 157 % of the original load, or 0.91 kN/m^2, all possible designs do not fulfill the structural requirement. When comparing the rotated and non-rotated patterns, the non-rotated pattern is, in general, stiffer; however, it has a higher value for the Unity-Check.

N_{hh}	20	16	12	16	12	20	12	12
N_{vh}	12	12	12	16	16	20	20	20
$Indiv_H$	-0.5	-0.5	-0.5	-0.5	-0.5	-0.5	-0.5	-0.5
P_H	1	1	1	1	1	1	1	3
A_c	0	0	0	0	0	0	0	22.5
RP_{incl}	-	-	-	-	-	-	-	0
Maximum deflection [mm]	33.01	32.90	32.77	32.36	32.20	32.10	31.97	31.86
Unity-Check	0.198	0.190	0.206	0.148	0.160	0.130	0.132	0.143
Maximum allowable load [%]	151	152	153	155	155	156	156	157

Table 46: Overview of numerous designs and their maximum deflection and Unity-Check of the non-rotated pattern

15.5 Conclusion

The case study based on the Echo building shows that the multi-criteria optimisation model can be applied in a real-life situation. It also reveals that, in this case, the transparency and insulation criteria strive for a minimisation of the number of hexagons, while the daylight criterion strives for maximising the hexagons horizontally and minimising them vertically. Furthermore, the score of the criteria is always greater than 0.60 when multiple criteria are considered, which means that all obtained near-optimal designs are performing fairly well on all objectives. Additionally, the panels weigh around 170 kg, while the original glass panel used in the building weighs 1,200 kg. This indicates the good strength-to-weight of a sandwich panel. Moreover, the U-value of the near-optimal panels are around 0.7 W/m^2K, while the facade panel used in the Echo building has a U-value of 1.1. Thus, there is a significant reduction in the U-value compared to the used glass panels. Figure 122, shows the building with implemented sandwich panels. It has to noted that a large misalignment is presented in predicting the stiffness of the panel, which limits the trustworthiness of the structural model. Additionally, the original panel uses PVB on the inside to enhance its strength, prevent the glass from shattering, and reduce the injury risk in case of breakage, which is not considered in the new design [5].

The sensitivity study shows that near-optimal panels for the daylight criterion facing west or east have a maximum number of hexagons horizontally and a minimum number of hexagons vertically, while panels facing south have a maximum number of hexagons both vertically and horizontally. Moreover, the near-optimal panels for the west and east orientations have the most robust design when they are being used in all considered orientations. Next, investigating the effect of the magnitude of the applied load shows that the maximum deflection increases linearly as the magnitude of the load increases. All panels comply with the structural requirement if the load is increased up to 46 % and 51 % for the rotated and non-rotated patterns, respectively. However, as the load increase is larger than 56 % and 57 % for the rotated and non-rotated patterns, respectively, all patterns are no longer structurally safe. A load increase between these values means that some designs are no longer fulfilling the structural requirement.

Figure 122: The Echo building with the near-optimal solutions of the normal-orientated hexagon structure in case all three criteria have equal weights.

16 Discussion

The discussion provides a short summary and covers the thesis process and the limitations and assumptions. The thesis process explains the steps and procedures followed in the thesis. This further clarifies the approach and the relationship between the different sections. This also gives some reflection of the process and guidelines for the creation of similar optimisation tools. Then, the limitations and assumptions related to the multi-criteria optimisation model are discussed. Subsequently, the limitations of the structural aspect are examined, which is primarily related to the comparison of the conducted experiments and the Diana models.

16.1 Summary

The thesis can be divided into two parts, which are shown in Figures 123 and 124. The boxes on the same row show aspects that were done simultaneously, and the arrow shows the relationship and chronological order between the different parts. The first part is shown in Figure 123, and it represents the first phase of the thesis, which explored the usability of topology optimisation. The different steps are discussed in the following chapters, and a short description of the gathered insights is provided.

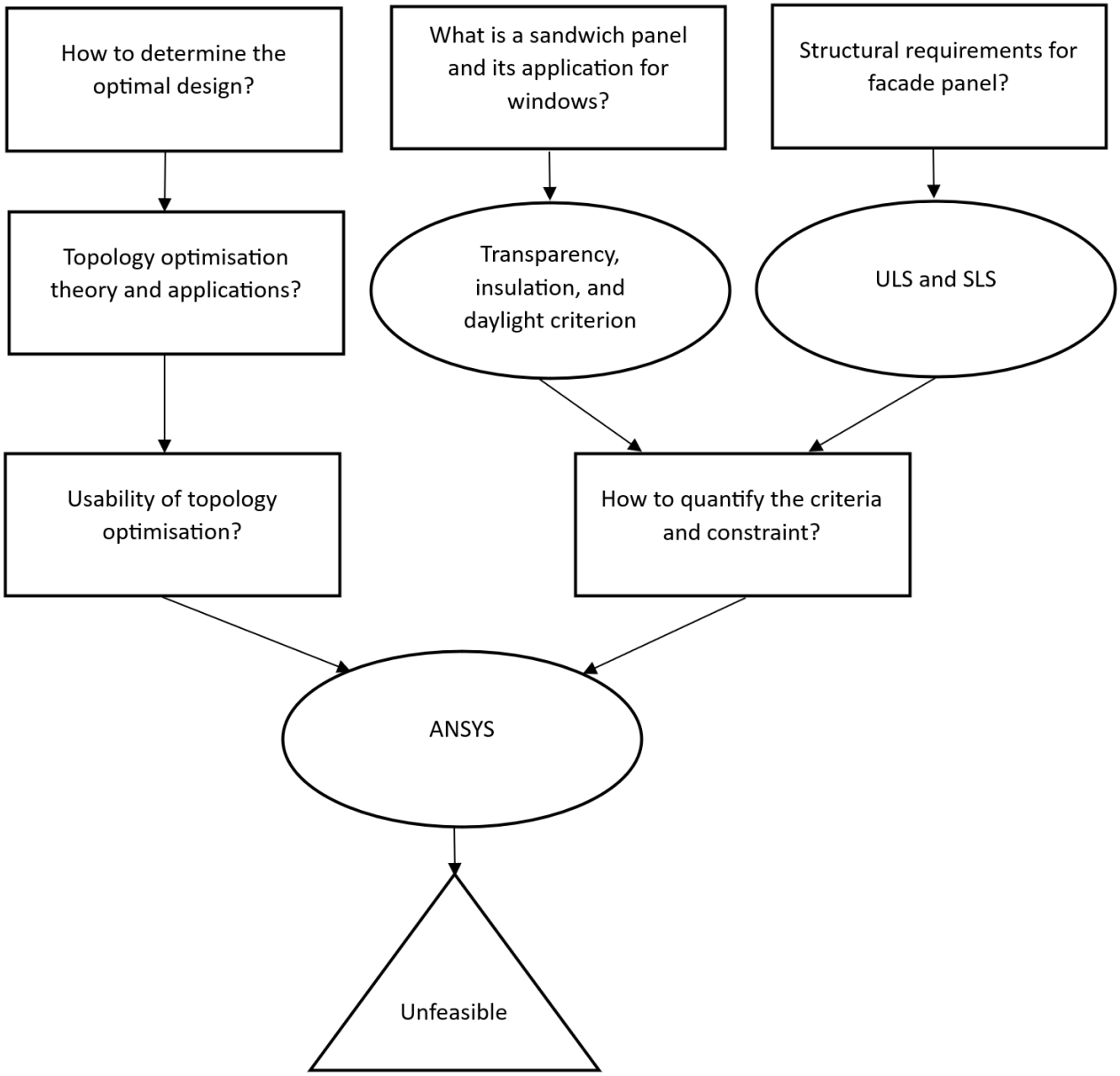


Figure 123: The flowchart describing the first part of the thesis.

16.1.1 What is a sandwich panel and its application for windows?

The first sub-question, in the middle at the top of Figure 123, is answered in Chapter 2 and is twofold. The first part is to understand the important characteristics of a sandwich panel. A sandwich panel is comprised of three layers, where the layer in the middle chiefly increases the distance between the two outer layers to enhance its overall bending stiffness. For example, a sandwich panel with a center layer with twice the thickness of the outer layers combined raises the bending stiffness by a factor of fifteen. These characteristics generate an efficient structure that is well-suited as a substitute for regular windows because the sandwich panel needs thinner glass sheets and has a lower weight while still having good insulation characteristics.

These characteristics have resulted in numerous studies into the usability of the sandwich panel as a facade panel. The materials used in a sandwich panel as a facade panel are thin glass sheets, typically made of aluminosilicate glass. The thin glass sheets have a thickness between 0.1 and 2 mm and are manufactured through, for example, a procedure called the micro-float process. The core material used in recent literature comprises different types of plastic manufactured with additive manufacturing to allow for a multitude of designs. Examples of used geometries are a space frame consisting of triangular shapes, which is also known as a truss-structure; a Voronoi pattern that produces an irregular shape; and a honeycomb pattern.

A first study into the topic of a facade panel with two thin glass sheets and an additive-manufactured core published in 2017 looked into the potential advantages of the sandwich panel over a regular window [74]. The subsequent paper investigated the structural efficiency of a sandwich with an additive manufactured core and multiple designs based on a Voronoi pattern [54]. This paper showed promising results. The next two papers focused more on the characteristics of the facade panel in relation to the durability of the core and the insulation characteristics of the panel [25, 65]. The durability aspects researched include UV radiation, elevated temperatures, and fire safety. Finally, a paper published in 2023 investigated design considerations such as aesthetics, structural safety, functional requirements, and manufacturing processes [61]. To summarise, the previous papers established the potential advantages of the facade panel, tested the structural behaviour and structural requirements, reviewed the potential disadvantages related to the core's durability, and provided a first overview of the different objectives and constraints determining the design. Therefore, the next step in researching the facade panel is creating a tool to determine the optimal design of the facade panel with the required constraints and desired objectives, as well as proving its usability.

This literature review showed the importance of transparency, sunlight, and insulation of the facade panel. These criteria are further researched in Chapters 4 and 5. This step is presented in the oval-shaped box in Figure 123, and the findings are briefly mentioned below.

The insulation capacity is its resistance to heat transfer, which occurs through radiation, convection, and conduction. Calculating the insulation capacity can be a complicated process, and assessing it through experiments is crucial. Fortunately, in one of the earlier papers on this type of facade panel, the insulation characteristics were investigated. Moreover, a simple calculation method has been proposed by Brugman [25] and uses the material properties related to the heat resistance of the materials, the overall thickness of the panel, and the contact areas between the core and the glass sheets. Thus, this calculation method is used in the tool to determine the insulation characteristics of the facade panels.

The subsequent objective is related to natural light. The natural light in buildings provides numerous health benefits as well as an increase in the productivity, which has been a topic of increased attention in recent decades [62]. The direct incoming sunlight can be perceived as undesirable by the inhabitants and may damage the light-sensitive objects in the building as well as reduce the readability of digital screens. Therefore, an optimal facade element provides sufficient indirect sunlight while preventing direct sunlight. However, in the Netherlands, the winter months have relatively short days; hence, it is desired to maximise the total incoming sunlight during this season.

16.1.2 What are the structural requirements of a facade element?

This question, described in Chapter 3, is answered through the use of the NEN codes, NEN-EN 1991-1-1-4+A1+C2:2011/NB:2019 + C1:202, which are the regulations followed by the Dutch construction industry. The most obvious load on a facade element is the wind load. This load is predominantly affected by the geographical location, orientation, and height of the facade element. The largest wind loads are observed at rural locations near the coast, and a wind load can be a compressive or tensile force depending on the wind direction. The other relevant load is the cavity pressure located between the glass layers. This is caused by a difference in outdoor and indoor temperatures, a change in the height of the production and installation locations, and an external load. These loads, except for the cavity pressure caused by a change in height, are variable loads. The NEN codes use the method of load combination when multiple loads are present to

determine the most unfavourable situation.

After discovering the most detrimental loading situation, the facade panel must remain structurally sound, known as the ultimate limit state (ULS), which focuses on limiting the maximum stresses in the materials. Also, the inhabitants must feel safe and be comfortable; this is dealt by the serviceability limit state (SLS), which concentrates on limiting deflection. Thus, the optimisation tool calculates the maximum stress and deflection and controls if these remain below the allowable levels. The tool only uses the wind load as the external load and is based on Dutch regulations. Additionally, a Python script is created to find the wind load and cavity pressures based on some input parameters. These aspects are shown in the two boxes at the top right of Figure 123.

16.1.3 How to determine an optimal design of the core geometry with topology optimisation?

This question explores topology optimisation as a possible method to determine the optimal design. First, an understanding of the theory behind topology optimisation is required. Then, the different applications using topology optimisation are investigated and compared with each other, and the best-suited application is selected to discover its usability.

Topology optimisation is a structural optimisation method developed at the end of the twentieth century, where the geometrical volume is divided into small segments with the use of finite elements. Each segment can either contain material or be void. This allows the algorithm to find the optimal position of the material, and a volumetric constraint is normally applied while the algorithm tries to find the stiffest geometry, known as compliance optimisation. The main principle is derived from the equilibrium constraint written as $K \cdot U = F$, where K is the stiffness matrix, U the displacement vector, and F the force vector. Compliance is equal to the inverse of the stiffness of the system. The two major drawbacks are the dependency of the results on the mesh size and the overestimation of the stiffness of checkerboards. The most recent topology optimisation algorithms take care of these drawbacks through additional steps or changes in the optimisation process. This is covered in Chapter 6.

These applications can be divided into three groups: the freestanding scripts, the topology optimisation integrated in software, and a combination of the two above-mentioned methods. The freestanding scripts, made in Matlab or Python, allow for endless flexibility in specifying the objectives and constraints; however, a profound understanding of coding is required. The second group requires the least amount of knowledge and has the smallest computational demand, but it has the poorest flexibility as the objectives and constraints are defined by the companies owning the software. Thus, depending on the relevant objectives and constraints in a specific situation, the software may or may not be usable. The considered programs are Rhinoceros and ANSYS because of the available university licenses. The last group has the advantages of both; for example, ABAQUS has a scripting interface. However, this last group is relatively new; therefore, little research has been conducted, and its functionalities are not optimised yet. For example, ANSYS introduced PyAnsys, but this Python library is not yet able to utilize the topology optimisation in ANSYS. Finally, it proved to be impractical to use either the first or last group in the given time span for this thesis. Subsequently, the most usable program discussed in group two is ANSYS because of its highest number of possible constraints, objectives, and reliable stress and strain information. These objectives include compliance, mass, volume, and stress, while the constraints are related to, for example, volume, compliance, displacement, and stress. This is discussed in Chapter 7.

After determining the best-suited topology optimisation application, optimisation models are created in ANSYS to better understand its usefulness in this context, which is done in Chapter 8. A small panel is used to reduce the computation time. The panel has a width of 360 mm, a length of 770 mm, and a thickness of 20 mm. The core is 16 mm, and each outer layer is 2 mm thick. A linear cubical mesh element with a size of 2x2x2 mm is used. The structural behaviour is determined by a linear structural finite element analysis. The model is built of a single material, for simplicity, and is a chemically strengthened glass with its material properties from NEN-EN2608. The applied constraint limits the maximum deflection and stress with respect to the NEN codes, and the objective is to minimise the total volume. Next, a window can be either two-sided supported, or four-sided; therefore, two models are used. Subsequently, a uniform pressure of 5.4 kN/m^2 is applied, which is in the order of magnitude of a very high wind load. This is equal to a facade element placed at 300 m in a coastal area. In total, six optimisations are performed that have a core volume constraint of 40 %, 30 %, or 20 % of the total core volume (770 x 360 x 16 mm) and are either two- or four-sided supported.

The results show that the ability to resist shear is an important aspect of the core layer, and a panel that is supported along all four sides requires less material in the core layer. The computation time ranges from 30 to 120 minutes, and the found core designs have a highly complicated pattern. The patterns are symmetrical along one or two axes, as either one or two axes of symmetry are present in the two- and four-sided supported

panels. Moreover, in both models, lines in the core connect both outside layers near the supports, while in the middle of the panel or on the yield lines in the four-sided supported panel, cylinders connect the layers.

Although the model provides logical results, the core cannot be easily 3D-printed. Additionally, no objectives related to daylight and insulation can be added. Therefore, the topology optimisation method is not usable in this case. These previously mentioned steps are shown on the left in Figure 123. Additionally, this revelation of the inability to use topology optimisation is the end of the first part of the thesis process.

Next, a new approach to determining the optimal design is searched for. This resulted in the use of a honeycomb pattern and the software Rhino. This is decided because the honeycomb pattern has a relatively good strength-to-weight ratio, shear resistance, and dipl. ing. Daniel Pfarr already has a Rhino Grasshopper script that creates a honeycomb pattern based on some parameters. The different steps needed to create a multi-criteria model are shown in Figure 124.

Figure 124: The flowchart describing the second part of the thesis.

First, the optimisation algorithm has to be chosen, which is done in Chapter 9. Well-known optimisation algorithms are the linear and nonlinear models. The linear model uses solely linear objectives and constraints, while the nonlinear models can have either nonlinear objectives or constraints or both. These models are especially used during the initial phases of an investigation to explore the basic behaviour of the end result with respect to the variables, objectives, and constraints. When multiple objectives are used, the relative importance of them becomes important, and two well-known approaches can be used. The first approach is the a priori approach. This requires the preferences to be known upfront through the use of, for example, weights

for each objective. The other approach, the posteriori approach, uses a vast array of weight combinations to understand the trade-off between those objectives. This trade-off can be represented through a Pareto front that gives an overview of the relationship between the different objectives and the optimal solutions. Subsequently, (meta)heuristics models can be used to solve the most complex problems with, for example, a large number of variables, non-trivial objective functions, and a limited computation time.

The optimisation algorithm for this tool should be able to cope with discrete variables and a highly non-linear problem while computational efficiency is required, which means that a (meta)heuristic model is the only suitable algorithm. Next, the genetic algorithm is chosen as (meta)heuristics model because of its integration in Rhino Grasshopper. The genetic algorithm is based on the theory of evolution, and it iteratively searches for the near-optimal solution. The algorithm usually includes a termination criterion dependent on the desired accuracy and computation time. Thus, the optimisation tool uses the genetic algorithm to find the near-optimal design of the facade panel with the before-specified input parameters.

16.1.4 How to create a multi-criteria optimisation tool?

In order to create a multi-criteria optimisation tool, different steps have to be followed. The first step is to define the variables, constraints, and objectives used in the optimisation process, as these give a general outline of the tool. The next step is to create an overview of all the required input parameters for the constraints, objectives, and optimisation algorithm. After establishing the framework, the structural constraints need to be validated, which is done by comparing them with DIANA and performing two experiments to assess the accuracy of the computer models. Subsequently, the daylight criterion has mesh-sensitive results, so a deeper understanding of its behaviour is essential. Finally, a case study is done to investigate the usability of the tool.

First, the variables, input parameters, constraints, objectives, and fitness function are defined. The optimisation tool uses six variables related to the core geometry, which are: the number of hexagons vertically and horizontally; the factorial change in the height of each row of hexagons; the pattern determining the factorial change per row; the inclination of the core per line segment determined by the distance to the reference point; and the location of the reference point. Subsequently, the model uses a structural constraint prescribing the maximum allowable deflection and stress based on the NEN codes. Moreover, the three objectives are: the transparency criterion, which maximises the transparency of the panel; the insulation criterion, which maximises the resistance to heat transfer; and the daylight criterion, which is a ratio between the total incoming sunlight at the winter solstice divided by the proportion of direct sunlight and the total sunlight at the summer solstice. Therefore, the final fitness function is the sum of all criteria multiplied by their respective weights, where each criterion has a value between zero and one and the sum of the weights equals one. The constraint is added to the fitness function and prevents structurally unsafe designs from becoming a near-optimal solution.

Subsequently, the optimisation tool is created in Rhino Grasshopper. The panel is created in four steps with the use of variables and input parameters. First, the 2D honeycomb pattern is created, then the line thickness is added, followed by altering the hexagon sizes, and finally, a 3D honeycomb pattern with inclination is generated. The next step is to calculate the value for each criteria and control the structural requirements. The transparency and insulation criterion are simply computed; however, the daylight criterion is more complicated and uses the Ladybug library in Grasshopper to determine the incoming sunlight. The structural behaviour is modelled with Karamba3D, and the genetic algorithm optimises the design based on the fitness score.

After creating the optimisation tool, the structural model in Karamba3D has to be validated. Karamba3D uses shell elements to represent the panel, and it is validated with a Diana model with shell elements and one with solid elements. The validation focuses on the maximum deflection and Unity-Check based on Von-Mises stresses. The Karamba3D models use multiple meshes comprised of shell elements to find an optimum between the required accuracy and the computation time. A structural linear analysis is performed in all models.

The results reveal that all Karamba3D models have a deflection smaller than the Diana model with solid elements, presumed to be the most accurate model, and greater than the Diana model with shell elements. The overestimation of the Karamba3D compared to the Diana model with solid elements ranges from 16 % to 24 %. The Unity-Check comparison shows that stress singularities appear when the element size in the Karamba3D models reduces to less than 10 mm. Thus, a larger element size, such as 10 mm, is used; however, the Unity-Check value in Karamba3D is always smaller than the Diana model with solid elements. Therefore, a safety factor of 2 is applied in Karamba3D to obtain conservative values.

Beside the validation of Karamba3D, two strength experiments are conducted to investigate the magnitude of the difference in maximum deflection in reality and the Diana model with solid elements used to validate Karamba3D. The panels are 710 by 450 mm, and the core consists of 10 layers of 1.8 mm with glass sheets

made from Falcon alumino-silicate glass 1 mm thick. The core has a line width of 6 mm, and it is made from FGF PET-G. The layers are glued together with LOCTITE AA3345, and only one side of the core is milled due to time constraints. The panels are hinged supported at two sides, and loaded with 40 to 48 kg at the center of the panel. The panel with a rotated honeycomb pattern shows a roughly linear increase in the deflection as the load increases, with an average stiffness of 80 N/mm. The non-rotated pattern shows some signs of local failure in the glue layer, especially at a load of 40 kg. This panel has an average stiffness of 58 N/mm, which is significantly lower than the panel with the rotated pattern. However, during the initial loading phase of the panels, the panel with the non-rotated pattern behaved the stiffest.

Subsequently, finite element models are created where the supports only allow compressive reaction forces to closely match the experiments. This is modelled with boundary interface elements, and a nonlinear analysis is conducted where the glass and core are modelled as linear elastic. Also, a perfect connection between the glass and core is assumed for simplicity. The stiffness found is roughly 7.5 times as high as the largest stiffness measured during the experiments. This part is further explained in the limitations section of the Discussion. Ideally, this conclusion probably requires alterations in the structural constraint, therefore, a dotted arrow is added in the figure. This step is not done in this thesis, thus, a dotted line is used.

Thereafter, the daylight criterion uses the Ladybug components within Grasshopper, and the direct and indirect daylight is measured through a mesh of sensors. The results appear to be mesh-sensitive, therefore, numerous meshes are compared to discover the best mesh with respect to the desired accuracy and computational demand. The calculation of direct sunlight and total sunlight uses different Ladybug components; thus, they are investigated separately. First, the direct sunlight, it is found that the radiation converges as the number of sensors exceeds 71,600, while no clear convergence is visible when fewer sensors are used. Also, when the number of sensors is increased above 71,600, the results hardly change while the computation time increases rapidly. Therefore, 71,600 sensors are required for sufficient accuracy and minimal computation time.

Second, in order to calculate the direct and indirect sunlight, another Ladybug component is used, which includes sunlight reflecting off the facade panel. The found solar radiation is mesh-sensitive, and additionally, a random error, whose magnitude depends on the number of sensors, is present. Therefore, multiple meshes with a different number of sensors are compared, and the results of each mesh are repeated thirty times to find the random error size. Next, a sensitivity analysis with respect to daylight is performed to uncover the required accuracy and maximum desired size of the random error. It is found that a mesh with at least 2,000 sensors is required to have a small enough random error to correctly differentiate between the possible designs.

Finally, a case study is done to investigate the usability of the optimisation tool in a real-life situation. The case study is based on the newly constructed Echo building at the TU Delft campus. It reveals that, in this case, the transparency and insulation criteria strive for a minimisation of the number of hexagons, while the daylight criterion strives for maximising the hexagons horizontally and minimising them vertically. Furthermore, all the near-optimal designs perform fairly well on all objectives. Additionally, the panels weigh around 170 kg, while the original glass panel used in the building weighs 1,200 kg. This shows that using a sandwich panel can greatly limit the weight of the panel. The insulation capacity is increased as well; the original panel has a U-value of 1.1 W/m^2K and the sandwich panels have a U-value of around 0.7 W/m^2K. Although the case study provided insightful findings, a huge disadvantage is the computational demand. The optimisation only considers three possible values per variable and already takes, on average, 12 hours to run a single optimisation.

The sensitivity study shows that the near-optimal panels of the daylight criterion for the west and east orientations have the most robust design when they are being used in the different orientations. Next, investigating the effect of the magnitude of the applied load shows that the maximum deflection increases linearly as the magnitude of the load increases, and if the load is increased by less than 46 % and 51 % for the rotated and non-rotated patterns, respectively, all patterns still fulfill the structural requirement. However, as the load increases by more than 56 % and 57 % for the rotated and non-rotated patterns, respectively, all patterns are no longer structurally safe. This means that there is a small window in which the weakest patterns no longer fulfill the requirement while the strongest patterns remain structurally safe.

16.2 Limitations and assumptions

The next part of the discussion is related to the limitations and assumptions of the thesis. This is divided into the optimisation model and all related aspects, and the structural behaviour is focused specifically on the comparison of the finite element models and the experiments.

16.2.1 Optimisation model

There are two prominent and general limitations of the optimisation model, which are discussed first. Then, some smaller limitations and assumptions are discussed. The first general limitation is that the model is designed around the hexagon pattern. Thus, the current model only finds the optimal hexagon shape in accordance with the chosen variables. It is not possible to optimise other shapes or find the optimal shape within a number of different shapes. Although the design is limited to a honeycomb pattern, recent literature on this topic concluded that the honeycomb pattern has good strength and stiffness [65]. The hexagon pattern is also the chosen pattern in the paper by Pfarr and Louter [61].

The second prominent limitation is the computational demand of the optimisation model. The case study uses a total of 729 design possibilities and only three criteria and one constraint. Nevertheless, a single optimisation took, on average, 12 hours to run. The most computationally demanding parts are the structural constraint and the daylight criterion. Thus, if a larger variety of designs is preferred or more criteria desired, the number of optimisations has to decrease, or a supercomputer should be used. All models in this thesis were performed on a laptop because the supercomputer at Tu Delft could not be used. The supercomputer at the Tu Delft could not be used because Rhino 7 is not an available software on the supercomputer. Although, a supercomputer would significantly reduce the computation time, it is not a prerequisite for using the optimisation tool. The optimisation tool can be used on desktop or laptop. It is advised to use a relatively new computer, at least 16 GB of RAM, and a dedicated graphics card.

Next, there are multiple smaller assumptions and limitations of the Rhino model, which are discussed below. First, the line width of the core should be significantly smaller than half of the line length of the smallest hexagon side. This is due to the implementation of adding the double line width of certain lines with a selection method based on the line length. This is shown in Figure 210 in Appendix H.

Another limitation is that the maximum factorial extension and shortening of the hexagons have to be carefully chosen. The factorial change is applied to change the hexagon length along a row while following a certain pattern related to a sine function. The sum of the new hexagon heights has to be equal to the original sum of the hexagon heights, as all the hexagons have to fit between the outer edges of the core. Moreover, as the lines are adjusted in two steps, the exact elongation or shortening is difficult to determine. Thus, it is advised to limit the domain of the variable $Indiv_H$ to ± 0.6. This topic is explained more extensively in Chapter 10.

Furthermore, the model uses multiple equality selection procedures to correctly shape the design of the hexagon, for example, selecting lines for the elongation and shortening processes. However, the script is susceptible to round-off errors. These are errors caused by numbers with an infinite number of decimals, but the computer program has to round the number off at a certain number of decimals. Especially when multiple rounded-off numbers are, for example, multiplied, the magnitude of the error increases. This has caused some problems in the model, and you should be aware of its presence. This could be dealt with by using an "Equality within tolerance" component, however, the magnitude of the tolerance can be difficult to asses. The tolerance must be defined to cope with a variety of designs and variable combinations. This would require extensive testing and predicting all possible design alterations that influence the required magnitude of the tolerance. In some cases, the magnitude of the tolerance can be easily defined and in such cases an "Equality within tolerance" component is used to prevent the occurrence of round-off errors. However, as a result of these complexities in defining the tolerance, the "Equality" component is still used at certain stages of the tool, although it is limited to a minimum.

Also, the mesh used in the daylight criterion should be carefully chosen because in the calculation component from the Ladybug library, which determines the direct and indirect sunlight, a random error is present. In other words, when the same calculation is performed twice, the answers differ slightly. This requires good understanding of the magnitude of the random error in the applied situation and the desired accuracy of the calculations. For example, in the examined situation of Chapter 14, some minor changes in the design only differ less than 1 % in their total incoming sunlight radiation. Thus, comparing these designs truthfully, the random error should be significantly smaller than 1 %. Additionally, as the required accuracy depends on the considered designs, a sensitivity study related to the daylight criterion and the determination of the magnitude of the random error with different mesh sizes need to be performed. This process may be avoided when a small enough mesh size is used; however, the unnecessary accuracy strongly increases the computational demand.

Another limitation related to the daylight criterion is the sunlight data. The data is used from a website that displays the available locations [48]. The availability of these data may be limited in certain areas, preventing the use of accurate local data. For example, in the Netherlands, there are fifty different locations. Thus, if specific sunlight data is required, this tool may not be usable.

Finally, the optimisation model uses Karamba3D to determine the structural behaviour. This requires shell elements instead of volumetric elements, which means that the center lines have to be used. Additionally, along

the boundary of the core, the line width changes, which limits the exactness of the center lines representing the lines due to the next reason. In the model, it has been decided to use a rectangular shape to represent the boundary line; thus, some small deviations between the true center lines and the used center lines exist. However, the validation with the Diana models shown in Chapter 12 reveals that the impact of this is probably limited.

16.2.2 Structural behaviour

The other important limitation is the discrepancy between the conducted experiments and the finite element models made in Diana. This is discussed in Chapter 13. A total of two experiments are performed, which are simply supported at their shortest edges and have a load applied in the center of the panel. The maximum stiffness during the experiments was 7.5 times as small as the calculated stiffness with the Diana models. Some possible causes are: a too stiff core; boundary conditions that add unintended stiffness; and the assumption that the glue layer provides a rigid connection.

The core of the panel is additive manufactured, and the Diana model uses a linear and homogeneous representation. Additionally, the E-modulus used is gathered from the manufacturer, which means that it is probably an overestimation of the stiffness. Furthermore, the 3D-printed structure consists of ten layers. This creates a highly heterogeneous material where the bonding between the printed layers affects the overall stiffness. This is a likely reason for the discrepancy between the stiffnesses. Also, a Diana model is created that only includes the core of the panel. The boundary conditions and loading conditions have remained the same. This resulted in a stiffness of around 5 and 6 N/mm for the non-rotated and rotated patterns, respectively. This stiffness is well below the stiffness of the experiments, which have a maximum of roughly 90 N/mm. The core structure itself reaches the stress limit of the core material at a deflection of 43 and 32 mm for the non-rotated and rotated patterns, respectively. This is in line with the experiments that also showed that the rotated pattern appears stronger.

The other possible attribute to the discrepancy can be the formulation of the boundary conditions. This uses a solid glass panel measuring 670 x 450 x 20 mm in order to compare it with analytical calculations. A Diana model with boundary interface elements to have a support with zero resistance to tension, and a Diana model with line supports are created. These models have the same deflection, which means that the boundary interface elements behave as expected. Next, an analytical calculation is performed that uses the beam theory. Two extreme cases are explored, which have constant pressure and a point load in the middle. The Diana results lie between these two cases, which would be expected. Thus, the boundary conditions are probably well represented and not a cause of the discrepancy.

Finally, the Diana models use the assumption that the glue layer provides a rigid connection. This is obviously not true; however, the extent of its importance with regard to the discrepancy is hard to determine, although in previous papers the same assumption has been applied. In the paper of Pfarr and Louter [61], a similar hexagon pattern was tested, although this panel's size is much larger. This panel is 876 by 1938 mm. The glue layer was applied by a robot, which results in an even distribution of the adhesive. Furthermore, both sides of the core structure were milled. The milling process greatly enhances the strength, especially the shear strength [61]. In the experiments performed, the glue was manually applied and only one side was milled. These are important differences between both experiments. The research performed an experiment where the deflections were measured with DIC, Digital Correlation System, and a finite element model in ANSYS with solid elements. The glue layer is also not included in the ANSYS model and assumes a rigid connection between the glass and core. The results of both are very similar until an applied pressure of 1000 Pa over the entire area. Afterwards, the difference grows as the non-linearity becomes more prominent, according to the paper. This shows that the manual application of the glue and having only one milled surface are likely causes of reducing the rigidity of the glue connection, especially the not-milled surface.

Thus, the cause of the discrepancy is probably an overestimation of the E-modulus of the core and the (local) failure of the glue connection, although the extent of both has not been determined. This uncertainty affects the credibility of the case study and, mainly, the comparison of the original panel and the sandwich panel. Therefore, it is not certain if the near-optimal panels are sufficiently stiff, as the structural model calculated. Moreover, only two experiments were performed, and the effect of the causes of the discrepancy may be more prominent in these panels than in the sandwich panel used in the case study. Thus, it is unknown if this difference would exist in the sandwich panels used in the case study and what the magnitude of the difference would be.

16.3 Thesis process

The main goal of the thesis is to develop a multi-criteria optimisation tool for the sandwich panel used as a facade element. Additionally, the main research question does not indicate a priori which criteria and constraints are used and which design possibilities it considers. Therefore, the thesis can be divided into the different steps or stages required to create an optimisation tool related to the design of a structural component, which is discussed in the Chapter 16.1. This also provides the opportunity to develop certain guidelines or principles that can be used in future research related to developing similar optimisation tools. These aspects are discussed in the next paragraphs. Subsequently, Figure 125 shows the general development of a design optimisation tool.

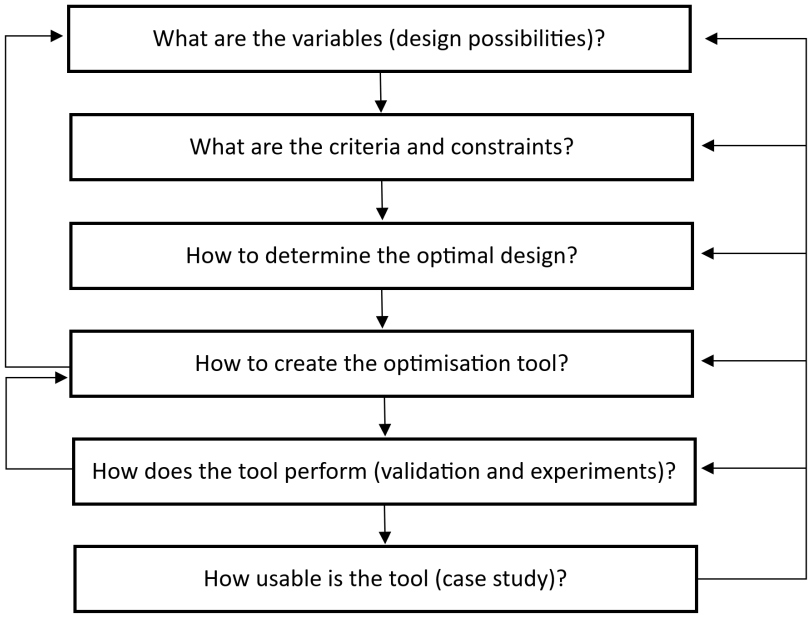


Figure 125: The general stages in the creation of an optimisation tool.

The starting point in the development process is to define the variables or possible designs, which greatly affect the overall complexity of the tool, the relevant objectives and constraints, and the range of applicability of the tool. Next, the criteria and constraints have to be selected. The criteria and constraints are chosen based on the preferred functionalities or performances of the design as well as the variables. The variables should affect the fitness score of the different criteria and constraints in order to be able to optimise its design. Furthermore, the criteria should not all strive for the same design, as a multi-criteria optimisation tool helps to discover the best compromised design. Then, the optimisation algorithm has to be defined to determine the optimal design, which is related to the variables, criteria, and constraints. This is extensively explained in Chapter 9. Finally, the optimisation tool has to be created in a certain environment, for example, a Python script or integrated into software. This choice adds restrictions to the allowable options within the variables, criteria, and constraints. The creation of the tool in an existing program probably requires less in-depth knowledge compared to using a Python script; however, the available optimisation algorithms may be very limited. As all these first four steps are closely related to each other, it is advised to quickly go through them initially while considering the available building time, prior experience with programs, and the preferred range of applicability. Those considerations reduce the likelihood of the need to rebuild the complete tool when time constraints are emerging or the applicability does not align.

During this thesis, the initial composition of the optimisation tool had to be partly discarded due to underestimating the building process. Originally, topology optimisation was considered because of its extended flexibility in the design possibilities and, therefore, being able to find the optimal design. However, as a result of limited experience with topology optimisation, substantial time has been spent to understand the theory, the applications, and the applicability of the applications related to this research. It took two to three months before the use of topology optimisation was abandoned as a result of the nonexistence of usable scripts or programs and therefore the complex building process, and printability. Although it was a necessary step during the initial phase of the thesis, it comprises one-fourth of the total duration of the research. This clearly indicates the necessity of quickly considering the first four steps prior to diving deeply into each of these steps.

After creating the optimisation tool, some of the criteria and constraints may demand validation; for example, in this thesis, the structural and sunlight models. This is an important step to create the credibility needed when using the tool and comparing its design to another in a case study. Additionally, the results of the validations and experiments may cause changes in the tool. Thus, it is desired to finish this step completely before starting the case study.

In this thesis, the validation was finished before the actual case study, and some alterations were made. However, the stiffness experiments were conducted during the case study, and the experiments showed some large misalignment between the structural models and experiments. This misalignment is extensively discussed in Chapter 16.3.2. Ideally, these results desire changes in the tool in order to have a trustworthy comparison in the case study. Unfortunately, this was not a possibility due to the extent of the needed time to properly improve the optimisation tool. Therefore, the optimisation tool ought to be completely validated to have a highly valued case study.

In the end, the case study shows the performance of the new design compared to the original design. This provides useful information about the potency of the new element. Moreover, the differences in the designs have to be closely examined, as this may reveal some aspects that were not considered in the optimisation tool but are an important part of the design. This analysis could result in necessary changes to the optimisation tool in each of the steps shown in Figure 125. Thus, during the creation, validation, and comparison, new alterations to the optimisation tool might appear. These possible changes have to be taken into account during the planning process, although it is difficult to know upfront if changes are needed and their extent during these stages.

One of these emerging aspects is the existence of PVB in the original panel in the case study. This creates a laminated glass layer on the inside to enhance its strength, prevent the glass from shattering, and in case of breakage diminish the risk of injury. This aspect is not part of the design considerations of the optimisation tool, while it is an important aspect of the final design. This is not incorporated into the optimisation tool as the tool is designed for the preliminary design phase, but it shows a possible desired alteration of the optimisation tool as a result of the case study results.

17 Conclusion

The main research question is: "*How to develop a multi-criteria optimisation tool for a glass facade element with a 3D-printed core?*". The first things to consider when developing a multi-criteria optimisation model are the variables, objectives, and constraints. These factors will determine the required optimisation algorithm. The structural requirement is the only constraint, and the objectives are related to transparency, insulation, and sunlight. The first optimisation algorithm that is considered is topology optimisation because it allows almost complete freedom of the core design and is closely related to FEA, which is important for the incorporation of the structural requirement. A variety of applications are considered, such as free-standing codes in Python and Matlab and topology optimisation integrated into software, for example, ANSYS. However, it is concluded that the topology optimisation cannot be used in the tool. The applications with it integrated into the software cannot be used because of the desired objectives, which are not related to structural performance, while most of these programs only consider objectives such as stress, displacement, and weight. Thereby, it is impossible to add objectives to these programs; thus, these programs can only be considered when the desired objectives are available. The second option, the free-standing codes, can, in theory, be used because of the flexibility related to adding variables, constraints, and objectives. Nevertheless, the codes require a high level of understanding of Python or Matlab and could result in a relatively long development time. Furthermore, an important issue is the printability of the final designs found by topology optimisation. Chapter 7 shows that the final designs obtained through the topology optimisation in ANSYS are not printable, and the complexity of the printing process can negatively affect the structural performance. Thus, due to the above-mentioned reasons, topology optimisation cannot be used in this case.

Then, a hexagon structure is used as a core design because: a hexagon structure has a relatively good shear resistance and strength-to-weight ratio; a hexagon-based design can be adjusted by multiple different variables; each line of the pattern can be continuously printed, which improves the structural strength; dipl. ing. Daniel Pfarr already has a small script that creates a honeycomb pattern in Rhino Grasshopper. Next, the genetic algorithm is used because the algorithm needs to cope with discrete variables and highly non-linear objectives and constraint while computational efficiency is required. Also, the algorithm should be widely available to be integrated into existing software.

The created tool was tested with a case study, which concluded that the multi-criteria optimisation tool is able to determine near-optimal designs based on multiple objectives and preferences. Additionally, the tool first creates the facade panel and then adds all the constraints and objectives. This process allows for the ability to simply add other constraints and objectives if desired. However, the tool has a high computational demand, which is closely related to the objectives and constraint used. To conclude, the multi-criteria optimisation tool is a good method of determining the optimal design in the preliminary design phase.

18 Recommendations

Within this thesis, a multi-criteria optimisation tool has been built that uses a honeycomb pattern as its base core geometry. However, there are still some important limitations that need to be discussed and opportunities that can be explored in further research. The general recommendations are divided into the multi-criteria optimisation tool itself and the assessment of the structural behaviour, which has been extensively addressed in the Discussion.

First, the optimisation tool itself. The structural requirements in the existing model only include the wind load as the external load. This is the easiest load to incorporate, and using a single load prevents the application of the method of load combinations described in the NEN. However, the cavity pressure is another important load and might, in some cases, be the most influential external load. Therefore, it is advised to include the cavity pressures in the optimisation tool. Another recommendation is related to the usability of the tool with regard to loading situations. A first step has been made by creating a Python script that can determine the cavity pressure and wind load. This script and the method of load combination can be added to the model, which makes the model more accessible. Moreover, the use of laminated glass on the inside might be incorporated as a design consideration which in the case of breakage diminishes the risk of injury.

Subsequently, insulation is probably, beside the structural requirements, the most important aspect of a substitute for windows. Currently, the whole model is based on a previously written paper and a simple calculation method. It is advisable to perform additional experiments to test the accuracy of the calculation method extensively. For example, the current calculation method is not really affected by the inclination of the core, while this could limit heat transfer through radiation.

Next is topology optimisation. In theory, topology optimisation is able to accommodate the optimisation procedure, while the core designs are not limited to a predefined design. Also, based on the size of the elements, the number of variables, which can either contain material or be void, is huge. Although the large number of variables can create a tool with enormous computational demand, further exploring the use of topology optimisation, especially as separate scripts, can be a viable option. Additionally, the integrated software and supplementary script are promising methods, but they have not been properly developed yet to be usable.

Furthermore, the current tool has proven that it is possible and manageable to use an optimisation tool to uncover the near-optimal design of the facade panel. Thus, the next step is to add on to the existing tool. Additional objectives can be used, as multiple aspects were discussed in the paper by Pfarr and Louter [61]. Moreover, according to the regulations, the windows are required to have a certain U-value, which can be added to the tool as a constraint. Next, the current model only works with a honeycomb pattern, which can be rotated and is affected by six variables. The tool can be expanded by adding another pattern, for example, a Voronoi pattern. This further enhances the architectural versatility of such a facade panel. Also, the optimisation tool only uses two glass sheets and a single additive manufactured layer, while using three glass layers and two 3D printed layers might be more efficient with regard to, for example, the daylight and insulation criterion. Finally, in the case study, only facade elements that are vertically placed are examined, while it is also interesting to investigate panels that are placed at an angle. This addition can also be added to the sensitivity study related to the daylight criterion, and it would be interesting to see which designs are near-optimal in different orientations and angles.

The other important recommendations are related to the reliability of predicting structural behaviour through computer models, especially FEM. The current research indicates that a large discrepancy exists between the conducted experiments and the finite element models of Diana. Therefore, the calculation of the deflection and potentially the stresses is not reliable yet. Moreover, in order to properly conduct a case study and compare the sandwich panel with a window, the correctness of the structural calculations is a necessity. Therefore, future research should focus on this prior to any of the suggested recommendations related to the optimisation tool itself.

References

[1] (n.d.). https://www.worldometers.info/world-population/world-population-projections/

[2] (n.d.). https://ceadgroup.com/wp-content/uploads/2022/05/MCPP-Recycled-PETG.pdf

[3] (n.d.). https://ceadgroup.com/wp-content/uploads/2021/05/Materials-MCPP-PIPG30GF.pdf

[4] (2022). https://www.octatube.nl/nl_NL/project-item.html/projectitem/221-echo-tu-delft

[5] (2023). https://glassforum.org/pvb-laminated-glass/

[6] (2023). https://karamba3d.com/

[7] (2023). https://dianafea.com/

[8] (2023). https://dianafea.com/

[9] Abdulghany, A. (2017). Generalization of parallel axis theorem for rotational inertia. *American Journal of Physics*, *85*(10), 791–795.

[10] AGC. (2023). https://www.agc-yourglass.com/sites/default/files/agc_docs/Falcon_0420_EN.pdf

[11] Akilo, M. (2018). Design and analysis of a composite panel with ultra-thin glass faces and a 3d-printed polymeric core. *University of Bologna, Masterarbeit*.

[12] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., & Sigmund, O. (2011). Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, *43*(1), 1–16.

[13] Ansola Loyola, R., Querin, O. M., Garaigordobil Jiménez, A., & Alonso Gordoa, C. (2018). A sequential element rejection and admission (sera) topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, *58*(3), 1297–1310.

[14] Argyris, J., Papadrakakis, M., Apostolopoulou, C., & Koutsourelakis, S. (2000). The tric shell element: Theoretical and numerical investigation. *Computer Methods in Applied Mechanics and Engineering*, *182*(1-2), 217–245.

[15] Argyris, J., Tenek, L., & Olofsson, L. (1997). Tric: A simple but sophisticated 3-node triangular element based on 6 rigid-body and 12 straining modes for fast computational simulations of arbitrary isotropic and laminated composite shells. *Computer Methods in Applied Mechanics and Engineering*, *145*(1-2), 11–85.

[16] Ashby, M., Melia, H., Figuerola, M., Philips, L., & Gorse, S. (2018). The ces edupack materials science and engineering package. *no. February*, 1–27.

[17] Baumers, M., Tuck, C., Wildman, R., Ashcroft, I., & Hague, R. (2011). Energy inputs to additive manufacturing: Does capacity utilization matter? *2011 International Solid Freeform Fabrication Symposium*.

[18] Beheshti, Z., & Shamsuddin, S. M. H. (2013). A review of population-based meta-heuristic algorithms. *Int. j. adv. soft comput. appl*, *5*(1), 1–35.

[19] Bendsoe, M. P., & Sigmund, O. (2003). *Topology optimization: Theory, methods, and applications*. Springer Science & Business Media.

[20] Bialkowski, S. (2022). Topos. https://www.food4rhino.com/en/app/topos

[21] Bloem, E. (2022). $Evabloem_unstudio_echo_exterior - 0091280x628$. Octatube. https://www.octatube.nl/nl_NL/project-item/projectitem/221-echo-tu-delft.html

[22] Boender, C. G. E., Rinnooy Kan, A., Timmer, G., & Stougie, L. (1982). A stochastic method for global optimization. *Mathematical programming*, *22*, 125–140.

[23] Bourell, D., Kruth, J. P., Leu, M., Levy, G., Rosen, D., Beese, A. M., & Clare, A. (2017). Materials for additive manufacturing. *CIRP annals*, *66*(2), 659–681.

[24] Boyce, P. (1997). Light, sight and photobiology. *Lighting Futures*, *2*(3).

[25] Brugman, S. (2019). Thin glass composite panel with 3d printed core: Thermal and structural properties.

[26] Castillo, E., Conejo, A. J., Pedregal, P., Garcia, R., & Alguacil, N. (2011). *Building and solving mathematical programming models in engineering and science*. John Wiley & Sons.

[27] Challis, V. J. (2010). A discrete level-set topology optimization code written in matlab. *Structural and multidisciplinary optimization*, *41*(3), 453–464.

[28] Chang, K.-H. (2014). *Design theory and methods using cad/cae: The computer aided engineering design series*. Academic Press.

[29] Collins, B. L. (1975). Windows and people: A literature survey: Psychological reaction to environments with and without windows.

[30] Cuttle, C. (1983). People and windows in workplaces. *Proceedings of the people and physical environment research conference, Wellington, New Zealand*, 203–212.

[31] EduPack, C. (2017). Granta design ltd., cambridge, uk.

[32] Edwards, L., & Torcellini, P. (2002). Literature review of the effects of natural light on building occupants.

[33] England, A. (1971). On stress singularities in linear elasticity. *International Journal of Engineering Science*, *9*(6), 571–585.

[34] Evans, A. (2001). Lightweight materials and structures. *MRS bulletin*, *26*(10), 790–797.

[35] Ferrari, F., & Sigmund, O. (2020). A new generation 99 line matlab code for compliance topology optimization and its extension to 3d. *Structural and Multidisciplinary Optimization*, *62*(4), 2211–2228.
[36] Finnegan, M. C., & Solomon, L. Z. (1981). Work attitudes in windowed vs. windowless environments. *The Journal of Social Psychology*, *115*(2), 291–292.
[37] Hexcel. (1999). Hexweb: Honeycomb attributes and properties: A comprehensive guide to standard hexcel honeycomb materials, configurations, and mechanical properties.
[38] Huang, X., & Xie, Y.-M. (2010). A further review of eso type methods for topology optimization. *Structural and Multidisciplinary Optimization*, *41*(5), 671–683.
[39] Huang, X., & Xie, Y. (2010). Evolutionary topology optimization of continuum structures with an additional displacement constraint. *Structural and multidisciplinary optimization*, *40*(1), 409–416.
[40] Incropera, F. P., DeWitt, D. P., Bergman, T. L., Lavine, A. S., et al. (1996). *Fundamentals of heat and mass transfer* (Vol. 6). Wiley New York.
[41] Kaszynski, A. (2020). *pyansys: Python Interface to MAPDL and Associated Binary and ASCII Files* (Version 0.43.2). Zenodo. https://doi.org/10.5281/zenodo.4009467
[42] Katopodes, N. D. (2018). *Free-surface flow: Computational methods*. Butterworth-Heinemann.
[43] Kennedy, G., & Goodchild, C. (2004). Practical yield line design. *Concrete Centre, Surrey, UK*.
[44] Leiva, J. P. (2011). Structural optimization methods and techniques to design efficient car bodies. *Proceedings of the International Automotive Body Congress, Troy, MI, USA*, 9–10.
[45] Leslie, R. (2003). Capturing the daylight dividend in buildings: Why and how? *Building and environment*, *38*(2), 381–385.
[46] Lin, H., Xu, A., Misra, A., & Zhao, R. (2020). An ansys apdl code for topology optimization of structures with multi-constraints using the beso method with dynamic evolution rate (der-beso). *Structural and Multidisciplinary Optimization*, *62*(4), 2229–2254.
[47] Liu, K., & Tovar, A. (2014). An efficient 3d topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, *50*(6), 1175–1196.
[48] LLC, L. T. (n.d.). https://www.ladybug.tools/epwmap/
[49] Louter, C., Akilo, M., Miri, B., Neeskens, T., Ribeiro Silveira, R., Topcu, Ö., van der Weijde, I., Zha, C., Bilow, M., Turrin, M., et al. (2018). Adaptive and composite thin glass concepts for architectural applications. *Heron*, *63*(1/2), 199–218.
[50] Louter, C., Veer, F., & Belis, J. (2008). Redundancy of reinforced glass beams: Temperature, moisture and time dependent behaviour of the adhesive bond. *Conference on Architectural and Structural Applications of Glass*, 479–490.
[51] Marais, H., Christen, H., Cho, S., De Villiers, W., & Van Zijl, G. (2021). Computational assessment of thermal performance of 3d printed concrete wall structures with cavities. *Journal of Building Engineering*, *41*, 102431.
[52] Mwema, F. M., Akinlabi, E. T., Mwema, F. M., & Akinlabi, E. T. (2020). Basics of fused deposition modelling (fdm). *Fused deposition modeling: strategies for quality enhancement*, 1–15.
[53] *Nederlandse technische afspraak 8800; energieprestatie van gebouwen - bepalingsmethode*. (2018). The Royal Netherlands Standardization Institute.
[54] Neeskens, T. (2018). Thin glass composites: Based on a structural efficiency increasing design strategy.
[55] *NEN-EN 1990+A1+A1/C2*. (2019). The Royal Netherlands Standardization Institute.
[56] *NEN-EN 1991-1-4+A1+C2*. (2011). The Royal Netherlands Standardization Institute.
[57] *NEN2608-2014*. (2014). The Royal Netherlands Standardization Institute.
[58] Nogal, M. (2022). Session n5. (meta)heuristics.
[59] Pariafsai, F. (2016). A review of design considerations in glass buildings. *Frontiers of Architectural Research*, *5*(2), 171–193.
[60] Pastia, C. (2020). 3d printing of buildings. limits, design, advantages and disadvantages. could this technique contribute to sustainability of future buildings? *Critical Thinking in the Sustainable Rehabilitation and Risk Management of the Built Environment: CRIT-RE-BUILT. Proceedings of the International Conference, Iași, Romania, November 7-9, 2019*, 298.
[61] Pfarr, D., & Louter, C. (2023). Prototyping of digitally manufactured thin glass composite façade panels. *Architecture, Structures and Construction*, 1–11.
[62] Phillips, D. (2004). *Daylighting: Natural light in architecture*. Routledge.
[63] Ribeiro Silveira, R. (2016). Flexible transparency: A study on thin glass adaptive façade panels.
[64] Russell, T. M. (1971). Building physics: Lighting: Wr stevens pergamon press, oxford, 45s., paperback 35s.
[65] Saleh, C. (2019). Ultra thin composite panel: A research on the durability and stiffness of a composite panel of (thin) glass and recycled pet.
[66] Santarsiero, M. (2015). Laminated connections for structural glass applications.

[67] Schittich, C. (2001). Glass architecture in the second half of the 20th century. *Glass Construction Manual*.
[68] Sigmund, O. (2001). A 99 line topology optimization code written in matlab. *Structural and multidisciplinary optimization*, *21*(2), 120–127.
[69] Sigmund, O. (2007). Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, *33*(4), 401–424.
[70] Simoen, C. (2016). Thin glass.
[71] Suresh, K. (2010). A 199-line matlab code for pareto-optimal tracing in topology optimization. *Structural and Multidisciplinary Optimization*, *42*(5), 665–679.
[72] Terman, M. (1989). Daylight deprivation and replenishment: A psychobiological problem with a naturalistic solution. *Proceedings of the International Daylight Conference, Long Beach, CA*, 438–445.
[73] Tyflopoulos, E., & Steinert, M. (2022). A comparative study of the application of different commercial software for topology optimization. *Applied Sciences*, *12*(2), 611.
[74] van der Weijde, I. (2017). Ultra lightweight thin glass facade: With structural, sun shading, insulating and transparent properties.
[75] Van Dijk, N. P., Maute, K., Langelaar, M., & Van Keulen, F. (2013). Level-set methods for structural topology optimization: A review. *Structural and Multidisciplinary Optimization*, *48*(3), 437–472.
[76] Veer, F., Janssen&, M., & Nägele, T. (2005). The possibilities of glass bond adhesives. *transition*, *19*(23), 40.
[77] Wang, M. Y., Wang, X., & Guo, D. (2003). A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, *192*(1), 227–246. https://doi.org/https://doi.org/10.1016/S0045-7825(02)00559-5
[78] Wells, G. N. (2011). The finite element method: An introduction. *Lecture notes for CT5142*.
[79] Wotton, E., & Barkow, B. (1983). An investigation of the effects of windows and lighting in offices. *Proceedings of the International Daylighting Conference*, 405–411.
[80] Xie, Y. M., & Steven, G. P. (1993). A simple evolutionary procedure for structural optimization. *Computers & structures*, *49*(5), 885–896.
[81] Zuo, Z. H., & Xie, Y. M. (2015). A simple and compact python code for complex 3d topology optimization. *Advances in Engineering Software*, *85*, 1–11.

Appendix A: Wind load magnitudes

Height	Area I			Area II			Area III	
[m]	Coast	Rural	Urban	Coast	Rural	Urban	Rural	Urban
1	0.93	0.71	0.69	0.78	0.60	0.58	0.49	0.48
2	1.11	0.71	0.69	0.93	0.60	0.58	0.49	0.48
3	1.22	0.71	0.69	1.02	0.60	0.58	0.49	0.48
4	1.30	0.71	0.69	1.09	0.60	0.58	0.49	0.48
5	1.37	0.78	0.69	1.14	0.66	0.58	0.54	0.48
6	1.42	0.84	0.69	1.19	0.71	0.58	0.58	0.48
7	1.47	0.89	0.69	1.23	0.75	0.58	0.62	0.48
8	1.51	0.94	0.73	1.26	0.79	0.62	0.65	0.51
9	1.55	0.98	0.77	1.29	0.82	0.65	0.68	0.53
10	1.58	1.02	0.81	1.32	0.85	0.68	0.70	0.56
15	1.71	1.16	0.96	1.43	0.98	0.80	0.80	0.66
20	1.80	1.27	1.07	1.51	1.07	0.90	0.88	0.74
25	1.88	1.36	1.16	1.57	1.14	0.97	0.94	0.80
30	1.94	1.43	1.23	1.63	1.20	1.03	0.99	0.85
35	2.00	1.50	1.30	1.67	1.25	1.09	1.03	0.89
40	2.04	1.55	1.35	1.71	1.30	1.13	1.07	0.93
45	2.09	1.60	1.40	1.75	1.34	1.17	1.11	0.97
50	2.12	1.65	1.45	1.78	1.38	1.21	1.14	1.00
55	2.16	1.69	1.49	1.81	1.42	1.25	1.17	1.03
60	2.19	1.73	1.53	1.83	1.45	1.28	1.19	1.05
65	2.22	1.76	1.57	1.86	1.48	1.31	1.22	1.08
70	2.25	1.80	1.60	1.88	1.50	1.34	1.24	1.10
75	2.27	1.83	1.63	1.90	1.53	1.37	1.26	1.10
80	2.30	1.86	1.66	1.92	1.55	1.39	1.28	1.15
85	2.32	1.88	1.69	1.94	1.58	1.42	1.30	1.17
90	2.34	1.91	1.72	1.96	1.60	1.44	1.32	1.18
95	2.36	1.96	1.74	1.98	1.62	1.46	1.33	1.20
100	2.38	1.96	1.77	1.99	1.64	1.48	1.35	1.22
110	2.42	2.00	1.81	2.03	1.68	1.52	1.38	1.25
120	2.45	2.04	1.85	2.05	1.71	1.55	1.41	1.28
130	2.48	2.08	1.89	2.08	1.74	1.59	1.44	1.31
140	2.51	2.12	1.93	2.10	1.77	1.62	1.46	1.33
150	2.54	2.15	1.96	2.13	1.80	1.65	1.48	1.35
160	2.56	2.18	2.00	2.15	1.83	1.67	1.50	1.38
170	2.59	2.21	2.03	2.17	1.85	1.70	1.52	1.40
180	2.61	2.24	2.06	2.19	1.88	1.72	1.54	1.42
190	2.63	2.27	2.08	2.20	1.90	1.75	1.56	1.44
200	2.65	2.29	2.11	2.22	1.92	1.77	1.58	1.46
225	2.70	2.35	2.17	2.26	1.97	1.82	1.62	1.50
250	2.74	2.40	2.23	2.30	2.01	1.86	1.66	1.54
275	2.78	2.45	2.28	2.33	2.05	1.91	1.69	1.57
300	2.82	2.50	2.32	2.36	2.09	1.95	1.72	1.60

Appendix B: Python script wind load

```
import pandas as pd
import numpy as np
import math

class WindLoad:
    def __init__(self,Area,Terrain,Height,Width,Tglass,Length,TLayer,VolumeFraction):
        self.at = Area
        self.tt = Terrain
        self.H = Height
        self.wf = Width
        self.tG = Tglass                        #Thickness glass panel
        self.lengthF = Length
        self.AreaFacade = Length * Width
        self.t3D = TLayer                       #Thickness 3D printed layer
        self.VolumeFraction = VolumeFraction
    def Load(self):
#Calculate Peak Velocity Pressure
        PvP = pd.read_excel("PeakVelocityPressure.xlsx")
        a = -1
        b = 0
        if self.at == 'AreaI':
            a = 0
        if self.at == 'AreaII':
            a = 3
        if self.at == 'AreaIII':
            a = 5
        if self.tt == 'coast':
            b = 1
        if self.tt == 'rural':
            b = 2
        if self.tt == 'urban':
            b = 3
        if a==-1 or b==0 or a==6 and b==1:
            return "False␣input␣values␣given,␣check␣Area␣type␣and␣Terrain␣type"
        if self.H > 300:
            return "This␣program␣only␣works␣with␣heights␣between␣0␣and␣300␣meter"
        for i in range(len(PvP)):
            if self.H == PvP.Height[i]:
                Pressure = PvP.iloc[i][a+b]
            if self.H > PvP.Height[i] and self.H < PvP.Height[i+1] and i <=len(PvP)-1:
                deltaP = (self.H - PvP.Height[i])*(PvP.iloc[i+1][a+b] - PvP.iloc[i][a+b])
                Pressure = PvP.iloc[i][a+b] + P2
#Caculate Structural Factor
        Eglass = 70000000000 #N/m2
        Part1 = (1/12)*self.tG*self.tG*self.tG*self.wf
        Part2 = (self.wf*self.tG)*(0.5*self.tG + 0.5*self.t3D)**2
        Ib = 2*(Part1 + Part2)
        K = np.pi #assumption glass supported by two hinge supports
        L = np.maximum(self.wf, self.lengthF)
        rhoGlass = 2600 #kg/m3
        m = 2*rhoGlass*self.tG*self.wf
        n1b = (K*K/(2*np.pi*L*L))*np.sqrt(Eglass*Ib/m) #eigenfrequency
        if n1b > 5:
            Check = False #assumption correct to use cscd =1
            CsCd = 1
        if n1b <= 5:
            Check = True #assumption incorrect, need to recalculate after 3D topology
            CsCd = 1
        self.CsCd = CsCd
#Calculate Force Coefficient
        CV = pd.read_excel("Cf0Value.xlsx")
```

```
        Ratio = self.wf / self.lengthF
        for i in range(len(CV)):
            if Ratio == CV.iloc[i][0]:
                Cf0 = CV.iloc[i][1]
            if Ratio > CV.iloc[i][0] and Ratio < CV.iloc[i+1][0] and i+1 < len(CV) -1:
                Cf0 = CV.iloc[i][1]+(Ratio-CV.iloc[i][0])*(CV.iloc[i+1][1]-CV.iloc[i][1])
        if self.lengthF >= 50:
            Labda = np.minimum(70, 1.4 * self.lengthF / self.wf)
        if self.lengthF < 15:
            Labda = np.minimum(70, 2 * self.lengthF / self.wf)
        if self.lengthF >= 15 and self.lengthF < 50:
            testLabda1 = 2 - (0.6 / 35) * (self.lengthFacade - 15)
            Labda = np.minimum(70, testLabda1 * self.lengthF / self.wf)
                #Table 7.16 Eurocode
        Acc = self.wf * (self.tG*2 + self.t3D)
        Aaa = self.wf*self.tG*2 + self.wf*self.t3D*self.VolumeFraction
        Phi = Aaa / Acc
        if Labda > 1 and Labda <=10 and Phi <=0.9 and Phi >=0.5:
            Func1 = 0.825 + 0.05*math.log10(Labda)
            Func2 = 0.885 + 0.025*math.log10(Labda)
            GammaLabda = Func2 - (Func2 - Func1)*(1/4)*(10*(Phi - 0.5))
                #determined from Figure 7.36 Eurocode
        if Labda > 1 and Labda <=10 and Phi <0.5 and Phi >=0.1:
            Func2 = 0.885 + 0.025*math.log10(Labda)
            Func3 = 0.99 + 0.005*math.log10(Labda)
            GammaLabda = Func3 - (Func3 - Func2)*(1/4)*(10*(Phi - 0.1))
                #determined from Figure 7.36 Eurocode
        if Labda > 10 and Labda <= 70 and Phi <=0.9 and Phi >=0.5:
            Func11 = 0.7744 + 0.1006*math.log10(Labda)
            Func21 = 0.8390 + 0.0710*math.log10(Labda)
            GammaLabda = Func21 - (Func21 - Func11)*(1/4)*(10*(Phi - 0.5))
                #determined from Figure 7.36 Eurocode
        if Labda > 10 and Labda <= 70 and Phi <0.5 and Phi >=0.1:
            Func21 = 0.8390 + 0.0710*math.log10(Labda)
            Func31 = 0.9891 + 0.0059*math.log10(Labda)
            GammaLabda = Func31 - (Func31 - Func21)*(1/4)*(10*(Phi - 0.1))
                #determined from Figure 7.36 Eurocode
        Cf = GammaLabda * Cf0
        self.Cf = Cf
#Calculate Wind Load
        Fwind = CsCd * Cf * (self.wf * self.lengthF) * Pressure # Newton
        Fpressure = Fwind / (self.wf *self.lengthF)             # N/m2
        return Fwind, Fpressure, Check
```

Appendix C: Insulation equations

The insulation factor, ϕ, can be calculated with Equation (48), (49), (50), and (51) in case of a rectangular facade element.

$$\phi = \frac{1}{1 + (\frac{a}{a'})^4} \tag{48}$$

$$a' = 28.9 \cdot \left(\frac{s \cdot t_{blad,1,ser}^3 \cdot t_{blad,2,ser}^3}{(t_{blad,1,ser}^3 + t_{blad,2,ser}^3) \cdot \chi} \right)^{0.25} \tag{49}$$

$$\chi = \left(\frac{z_1}{16}\right) \cdot \left(0.4198 + 0.22 \cdot e^{\left(-6.8 \cdot \left(\frac{a}{b}\right)^{1.33}\right)}\right) \cdot \left(\frac{b}{a}\right)^2 \tag{50}$$

$$z_1 = 181.8 \cdot \left(\frac{a}{b}\right)^2 \cdot \left(0.00406 + 0.00896 \cdot \left(1 - e^{\left(-1.123 \cdot \left(\frac{b}{a} - 1\right)^{1.097}\right)}\right)\right) \tag{51}$$

Where:

- a is the length of the shortest edge of the rectangular glass panel. $[mm]$
- a' is the cavity characteristic length of the rectangular glass panel. $[mm]$
- s is the cavity thickness. $[mm]$
- b is the length of the longest edge of the rectangular glass panel. $[mm]$
- χ is the form factor of the characteristic length of the glass panel. [-]

Under the assumption that only one glass layer is used in a glass panel, Equation (52) can be used. Relevant values of t_pl, and the respective t_{nom} are given in Table 47.

$$t_{blad,i,ser} = t_{pl} = t_{nom} - \Delta_t \tag{52}$$

Where:

- t_pl is the glass thickness. $[mm]$
- t_{nom} is the nominal thickness of the glass panel. $[mm]$
- Δ_t is the maximum permissible deviation of the thickness. $[mm]$

t_{nom} **in mm**	t_{pl} **in mm**
3	2.8
4	3.8
5	4.8
6	5.8

Table 47: Nominal glass thickness and minimum glass thickness [57].

Appendix D: Visual representation stresses of DIANA with shell elements

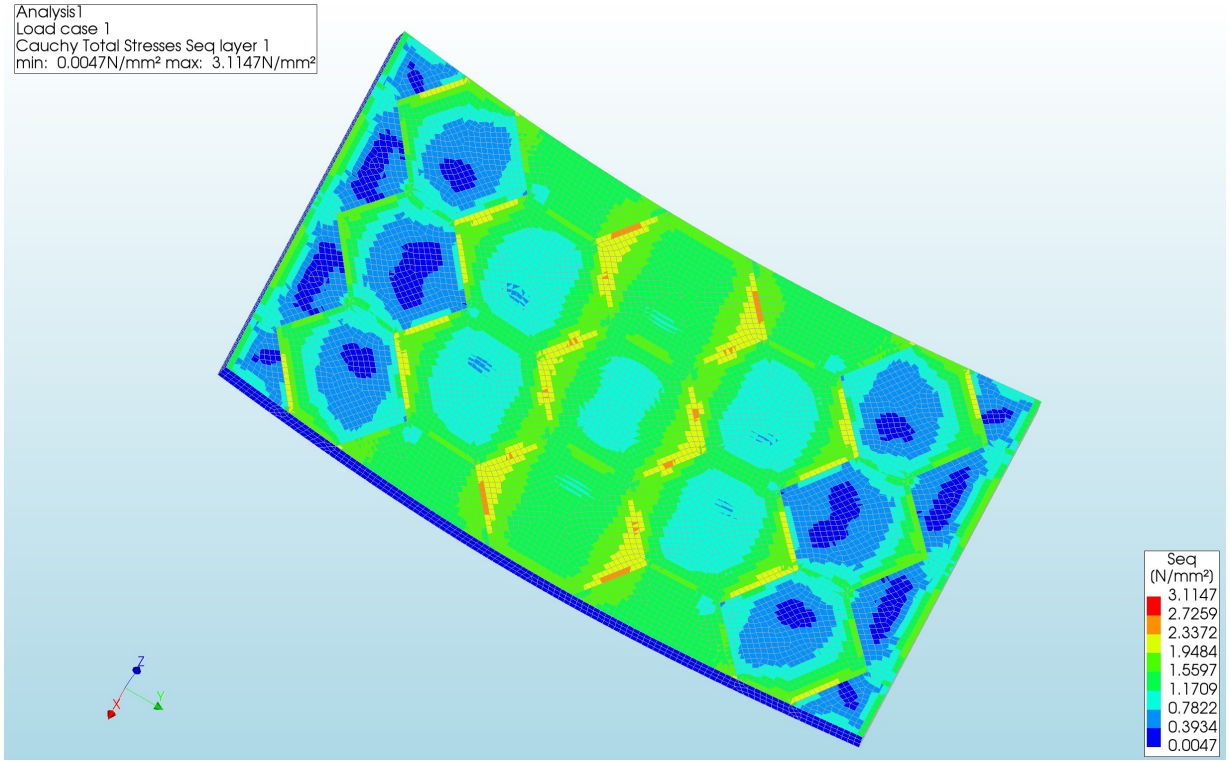


Figure 126: The Von Mises stresses of the top side of Pattern 1 with an element size of 6 millimetre

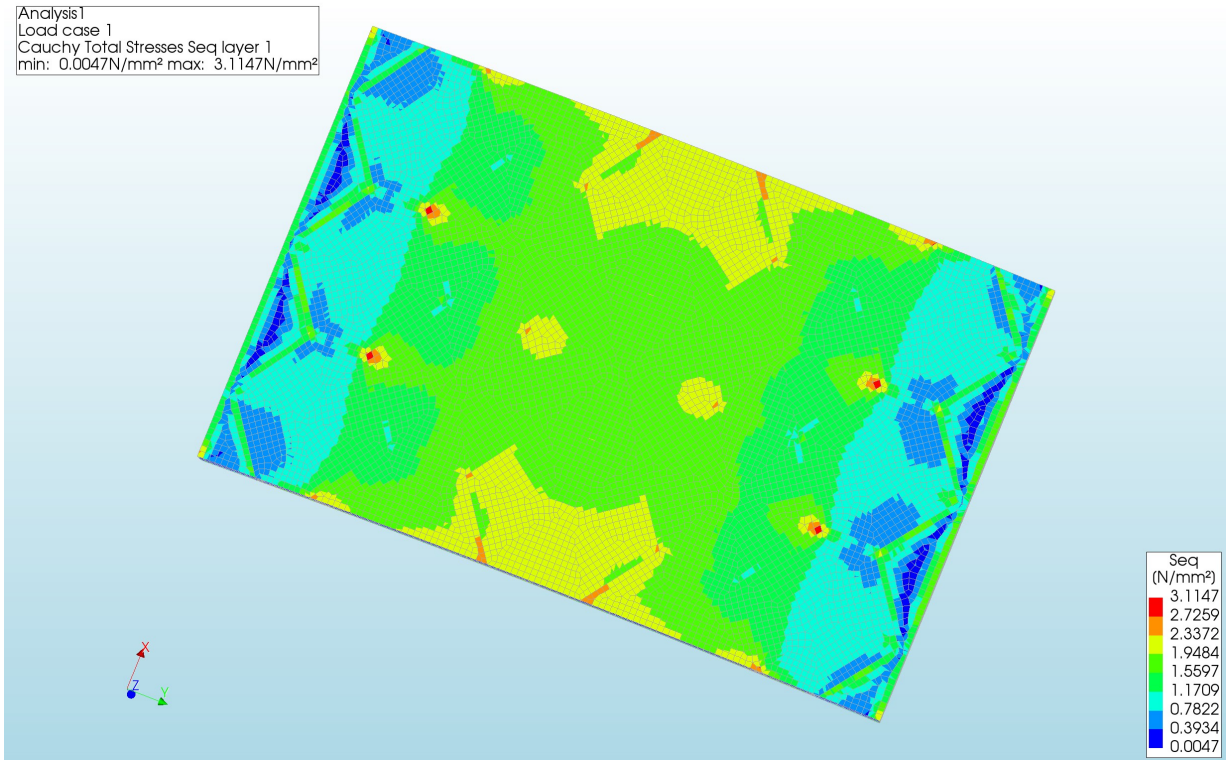


Figure 127: The Von Mises stresses of the bottom side of Pattern 1 with an element size of 6 millimetre

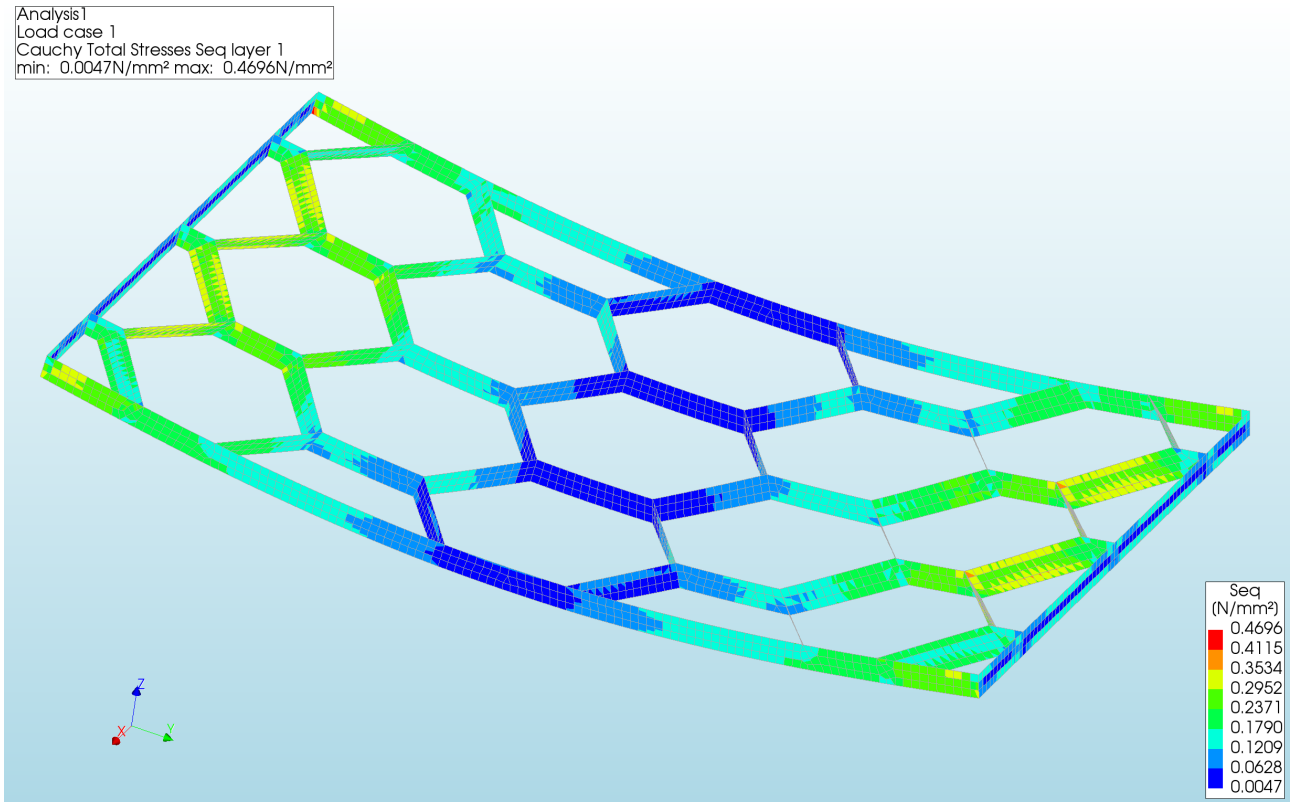


Figure 128: The Von Mises stresses of the Core of Pattern 1 with an element size of 6 millimetre

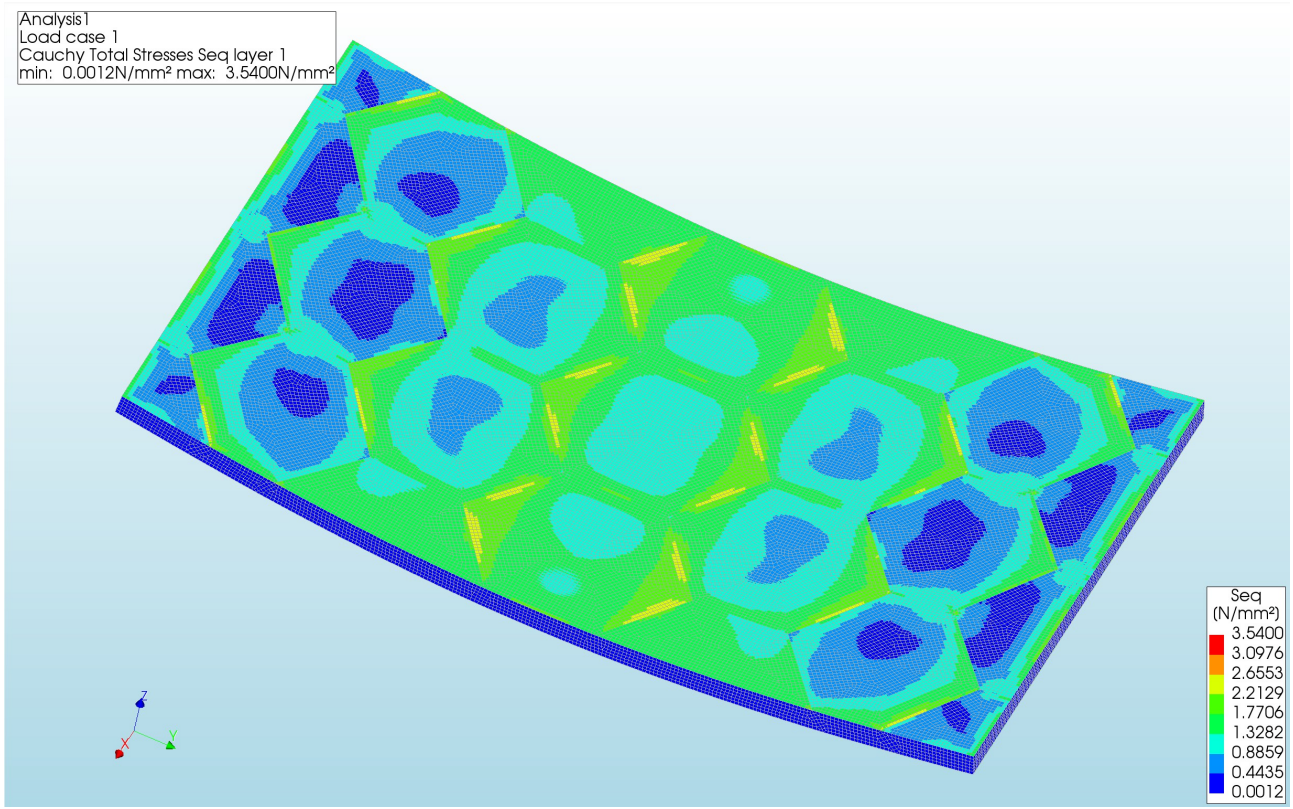


Figure 129: The Von Mises stresses of the top side of Pattern 1 with an element size of 3 millimetre

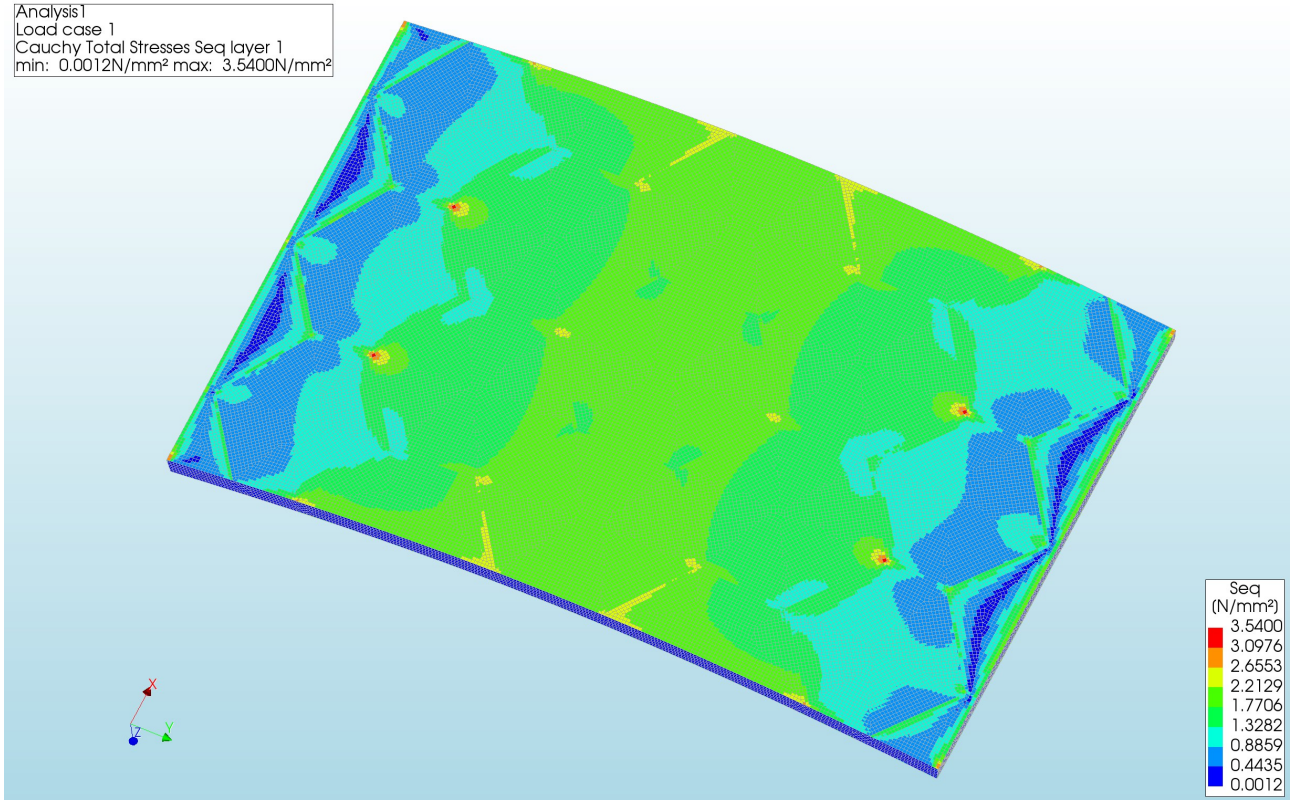


Figure 130: The Von Mises stresses of the bottom side of Pattern 1 with an element size of 3 millimetre

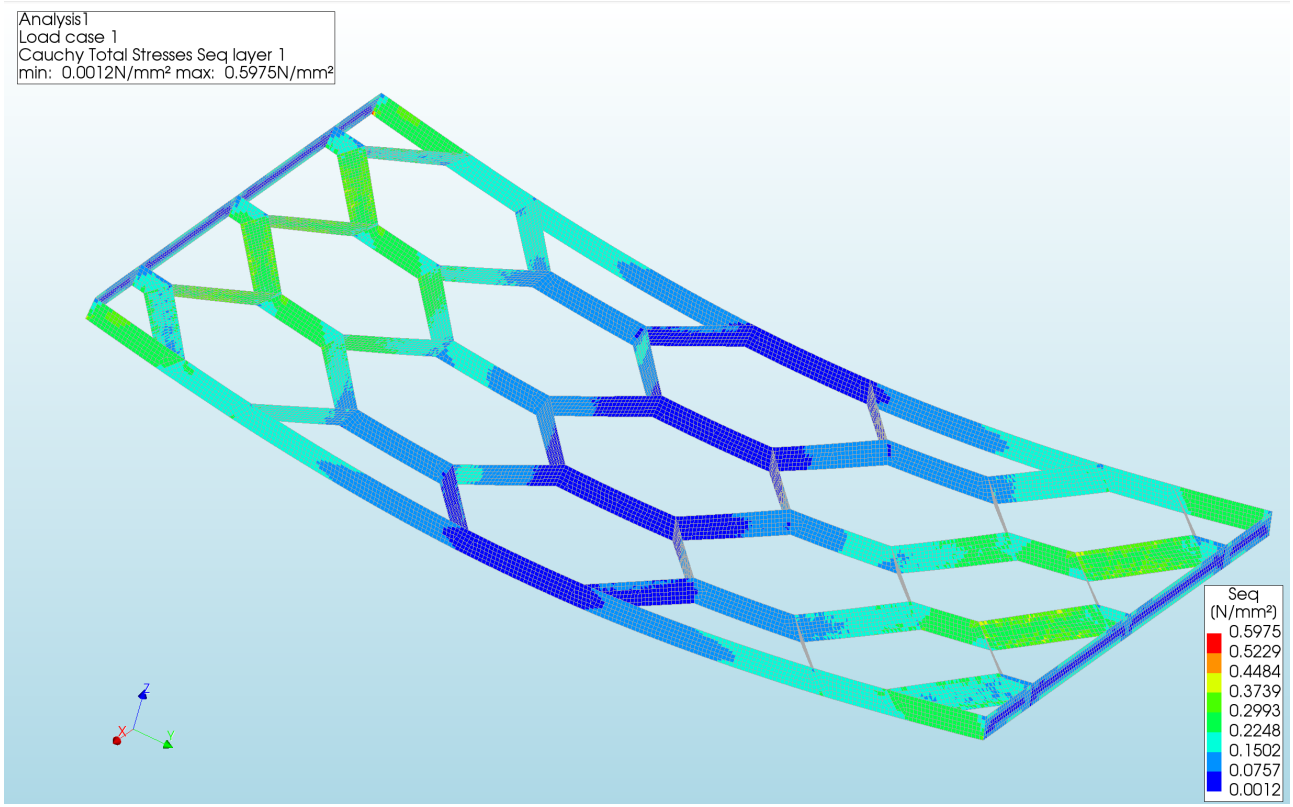


Figure 131: The Von Mises stresses of the Core of Pattern 1 with an element size of 3 millimetre

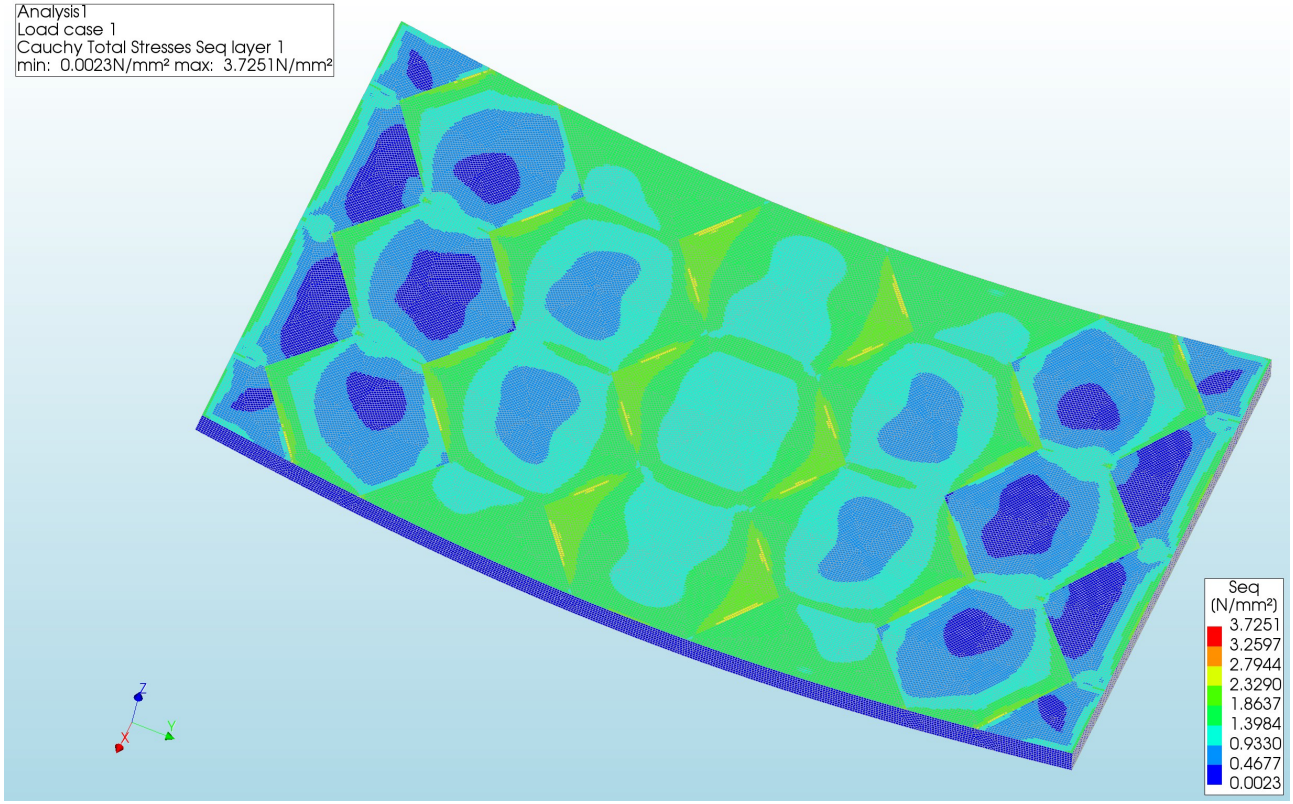


Figure 132: The Von Mises stresses of the top side of Pattern 1 with an element size of 2 millimetre

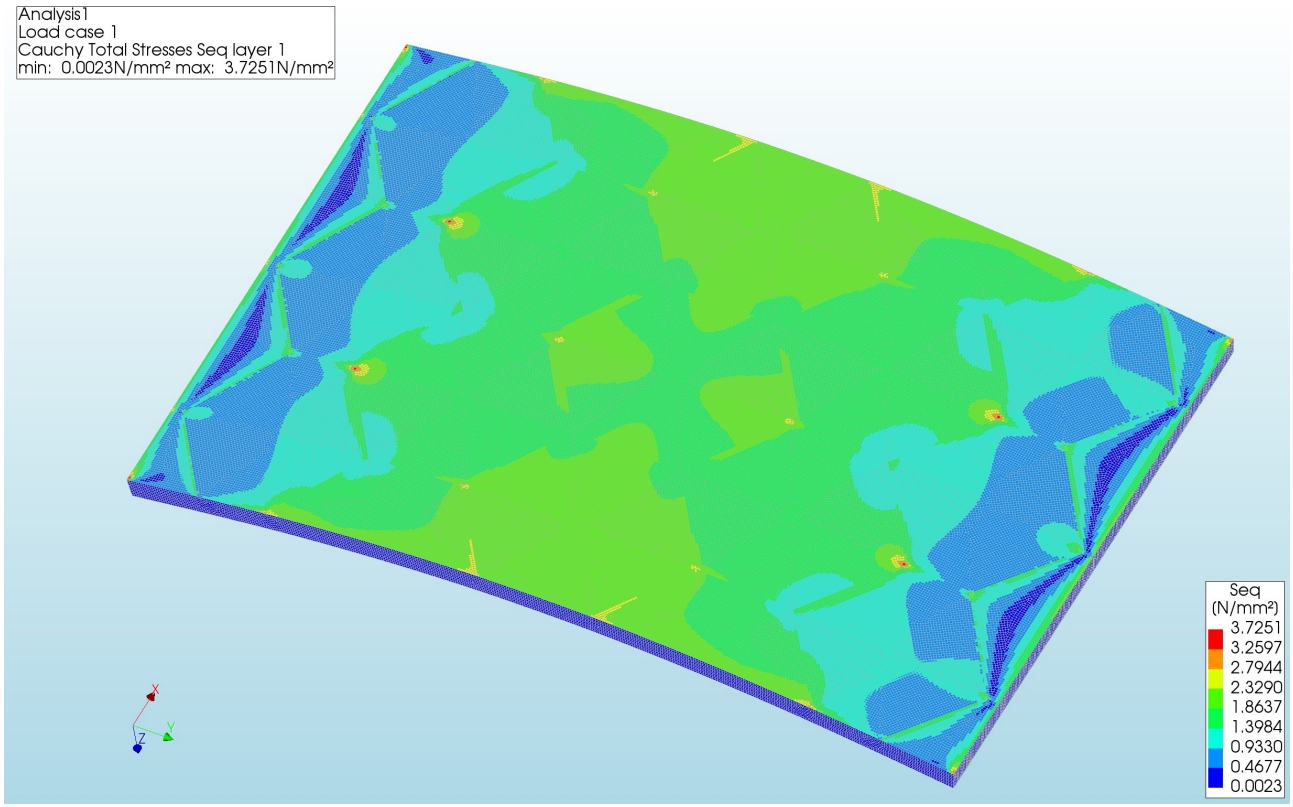


Figure 133: The Von Mises stresses of the bottom side of Pattern 1 with an element size of 2 millimetre

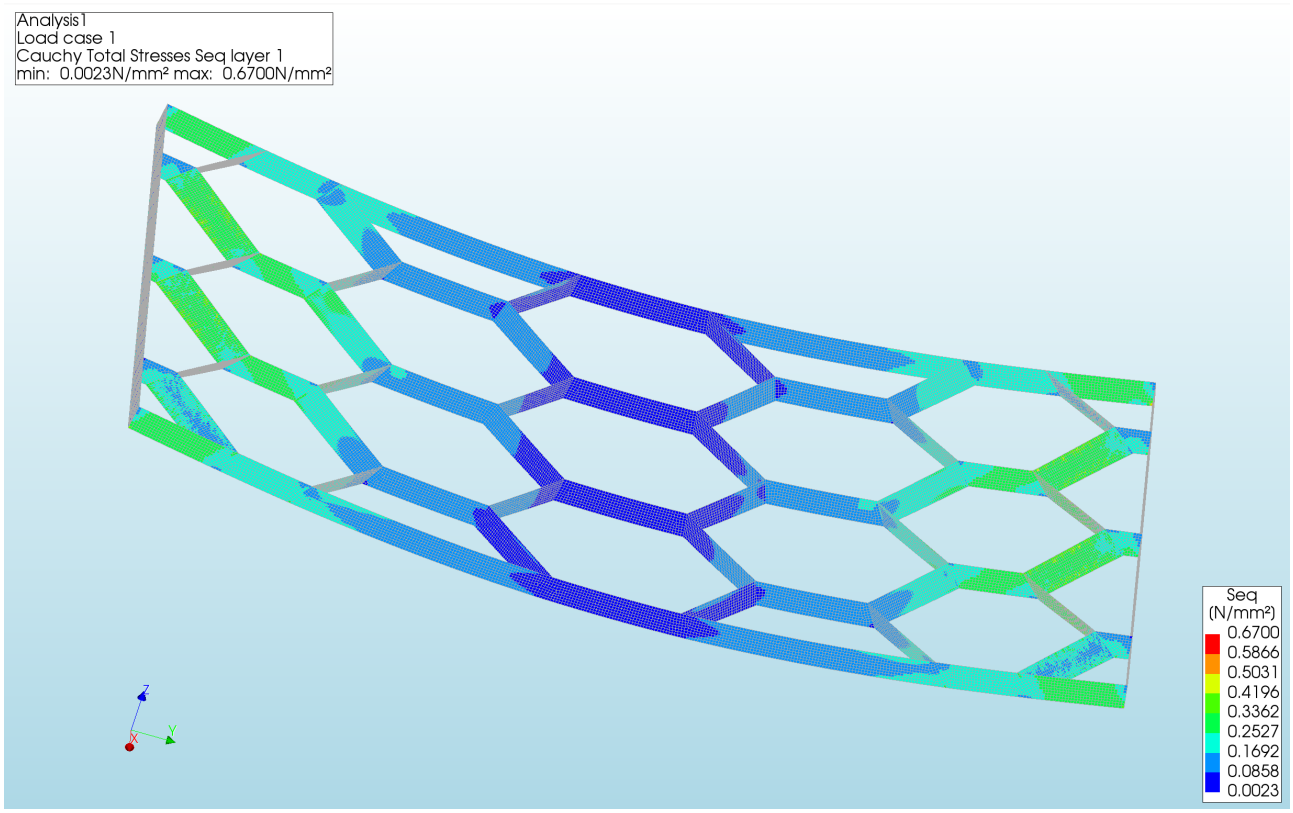


Figure 134: The Von Mises stresses of the Core of Pattern 1 with an element size of 2 millimetre

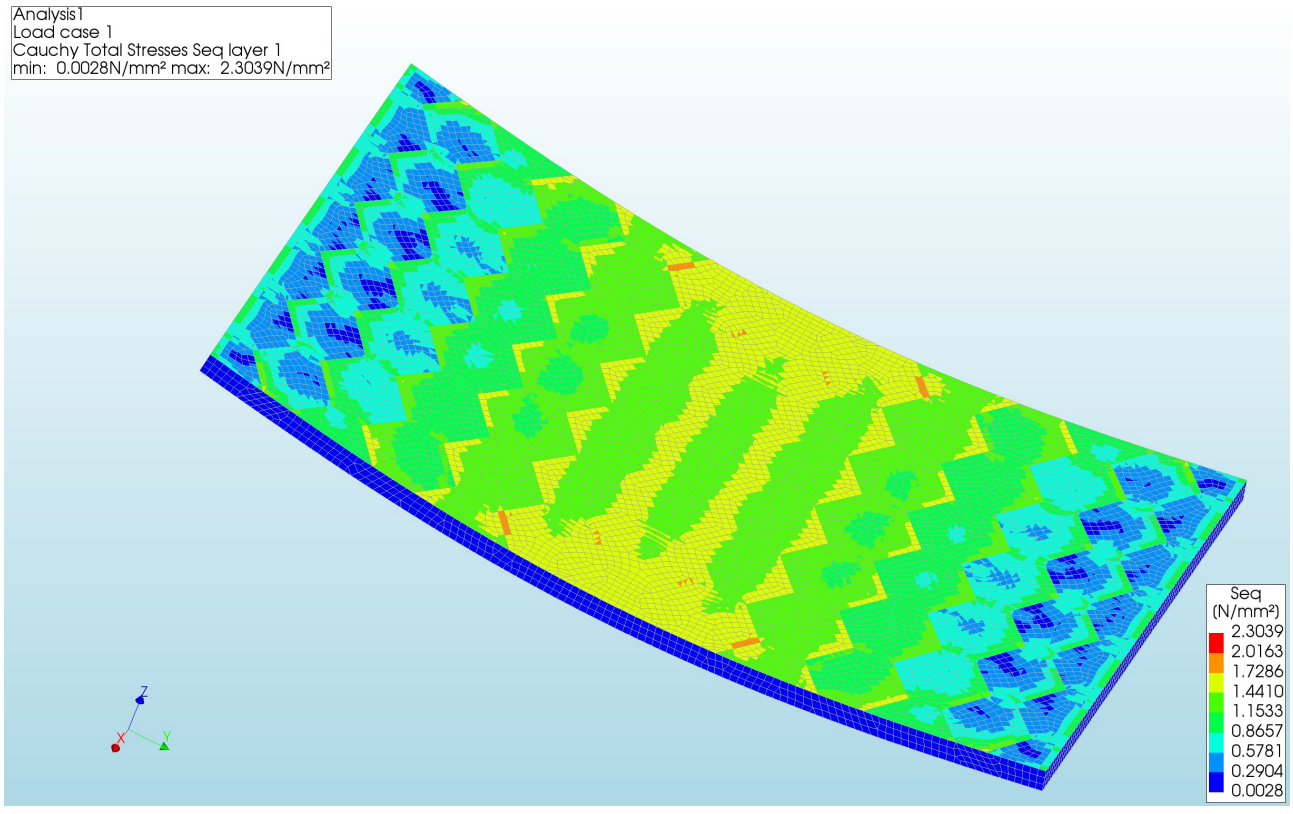


Figure 135: The Von Mises stresses of the top side of Pattern 2 with an element size of 6 millimetre

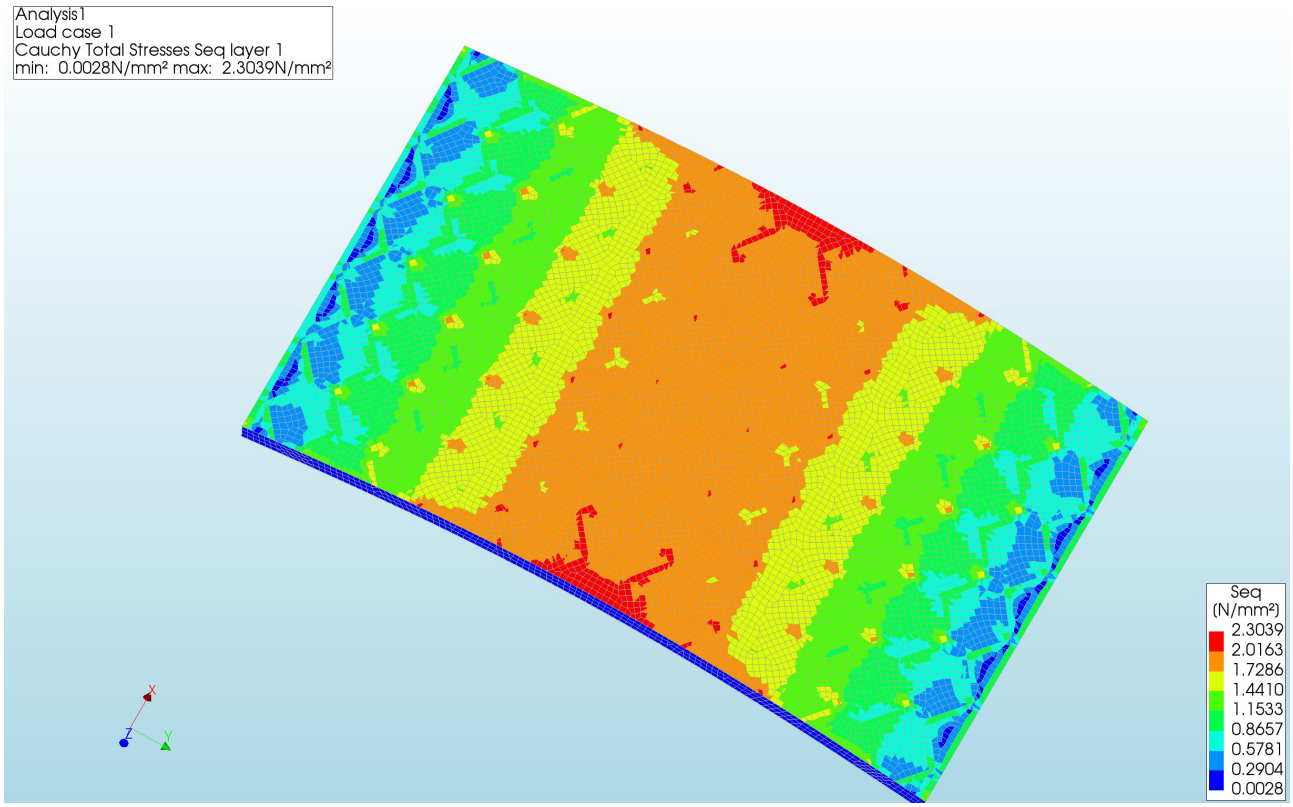


Figure 136: The Von Mises stresses of the bottom side of Pattern 2 with an element size of 6 millimetre

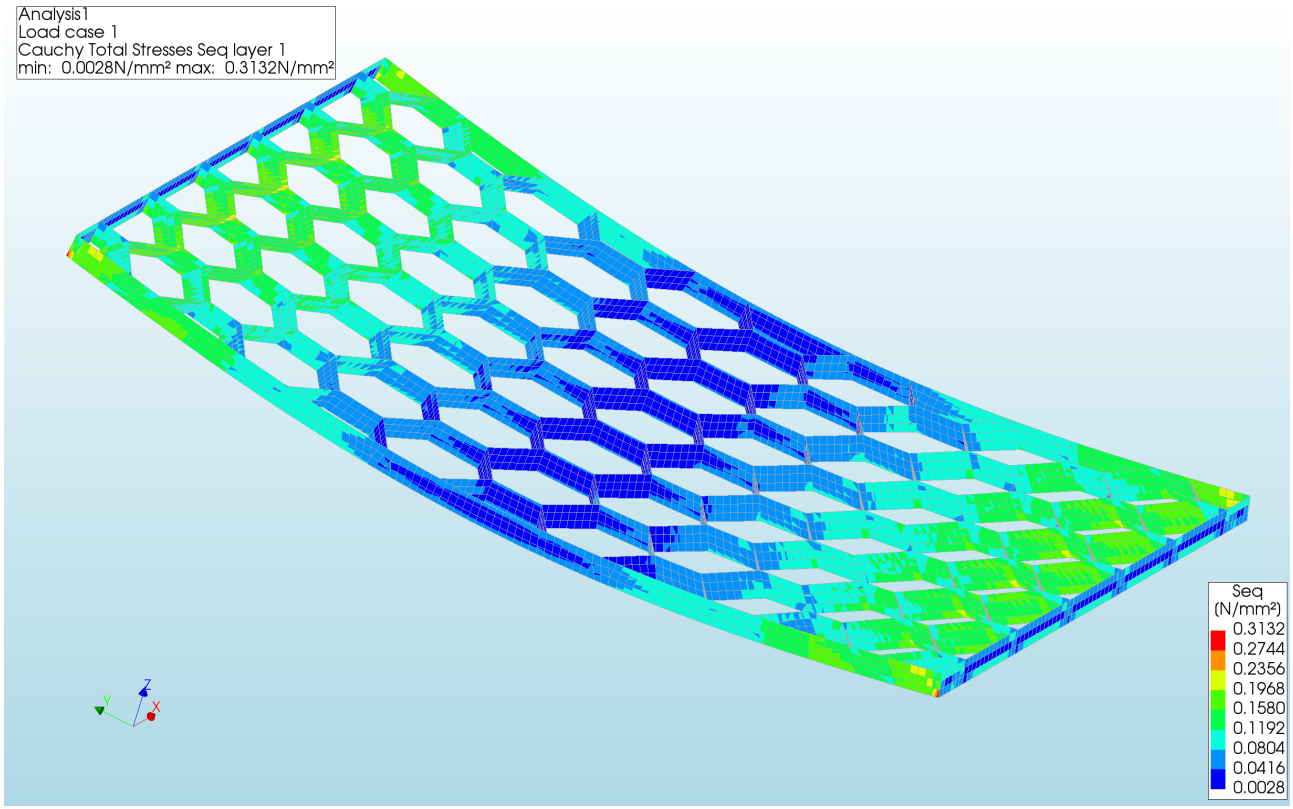


Figure 137: The Von Mises stresses of the Core of Pattern 2 with an element size of 6 millimetre

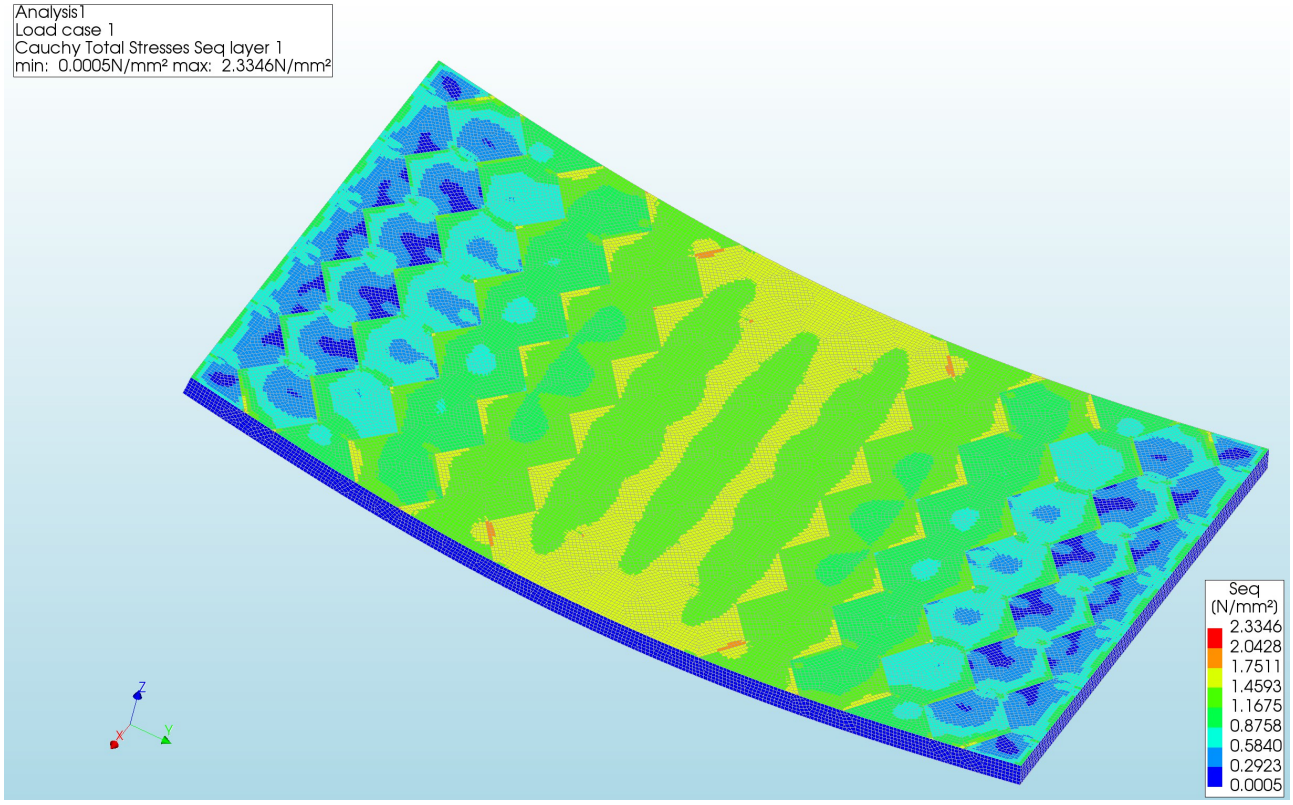


Figure 138: The Von Mises stresses of the top side of Pattern 2 with an element size of 3 millimetre

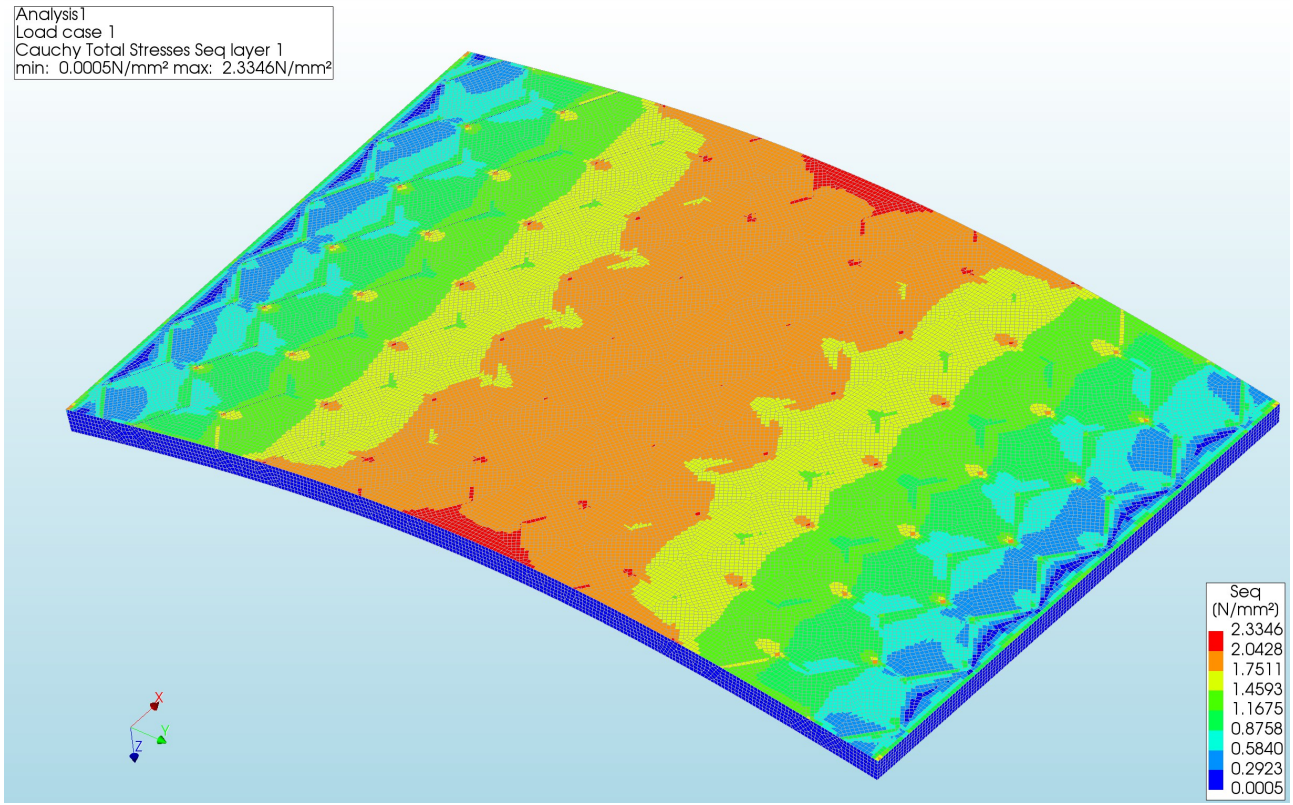


Figure 139: The Von Mises stresses of the bottom side of Pattern 2 with an element size of 3 millimetre

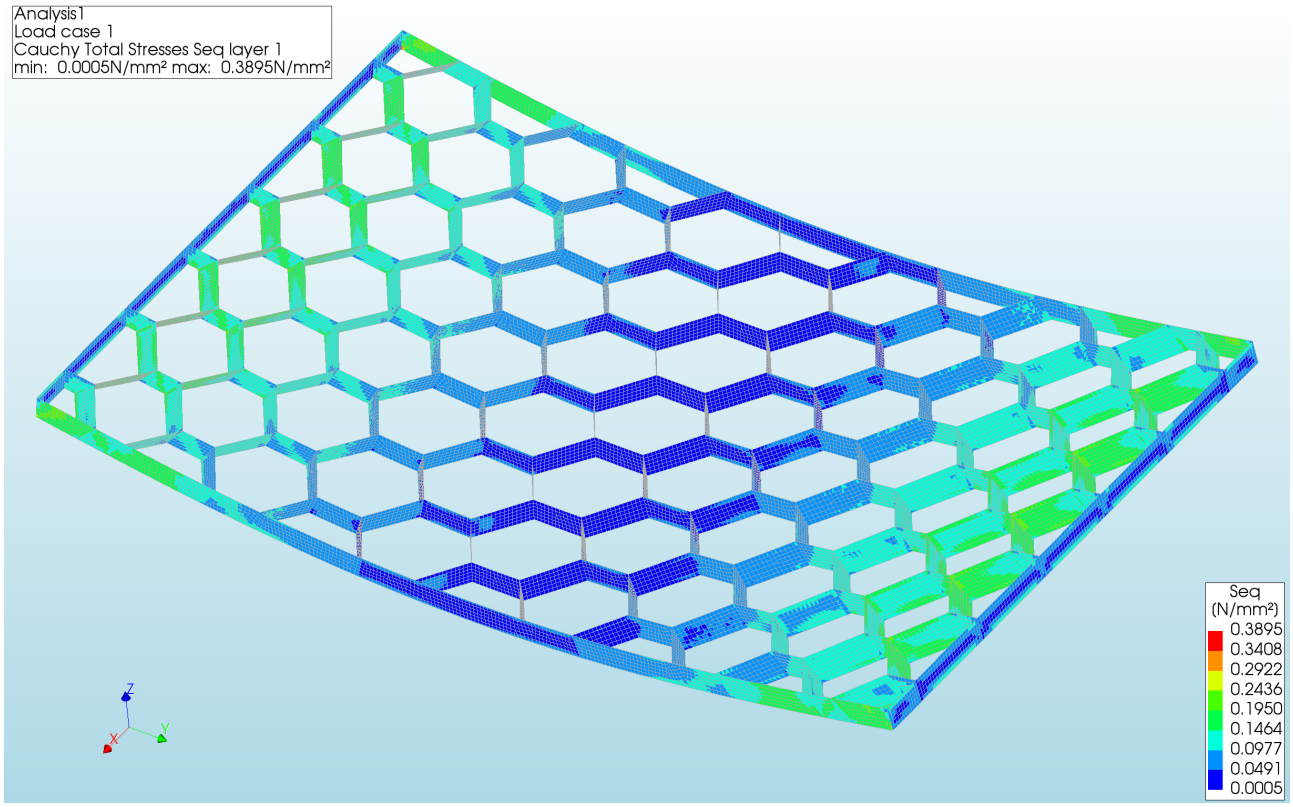


Figure 140: The Von Mises stresses of the Core of Pattern 2 with an element size of 3 millimetre

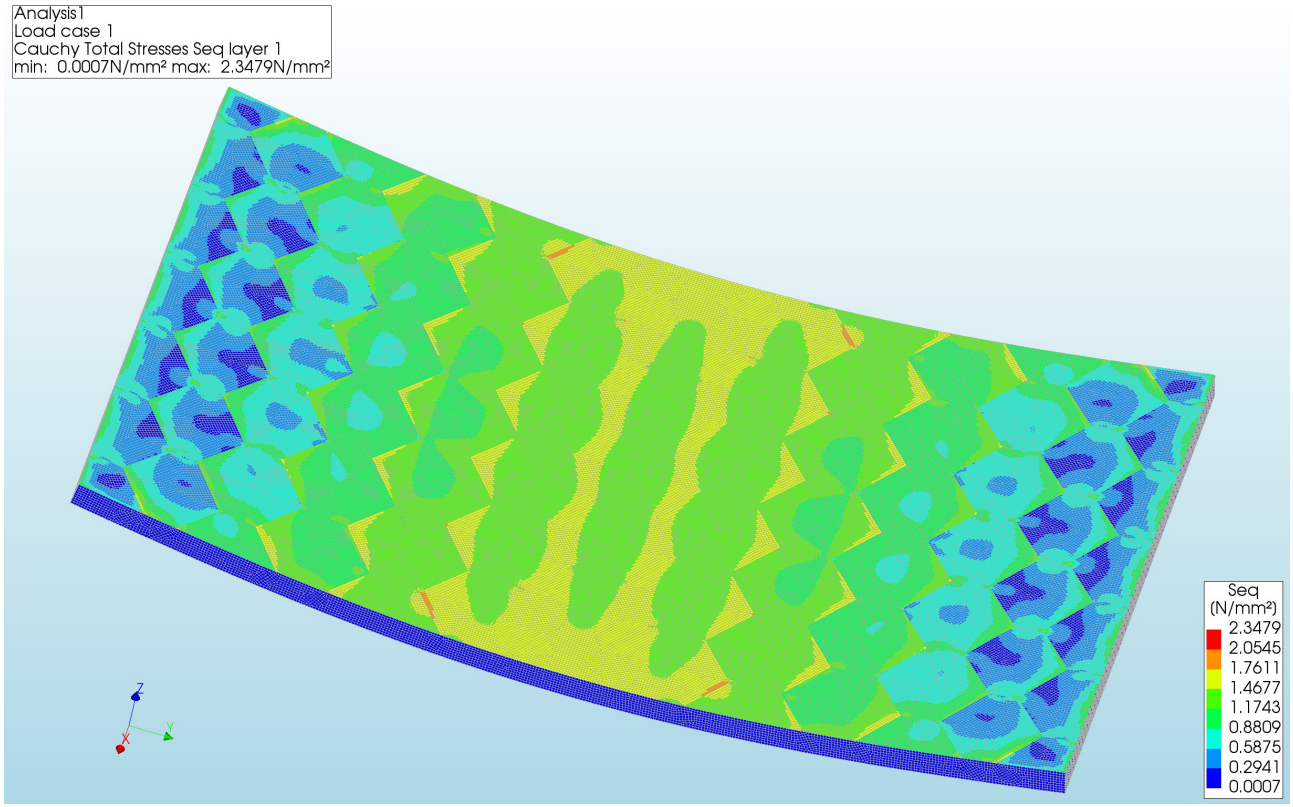


Figure 141: The Von Mises stresses of the top side of Pattern 2 with an element size of 2 millimetre

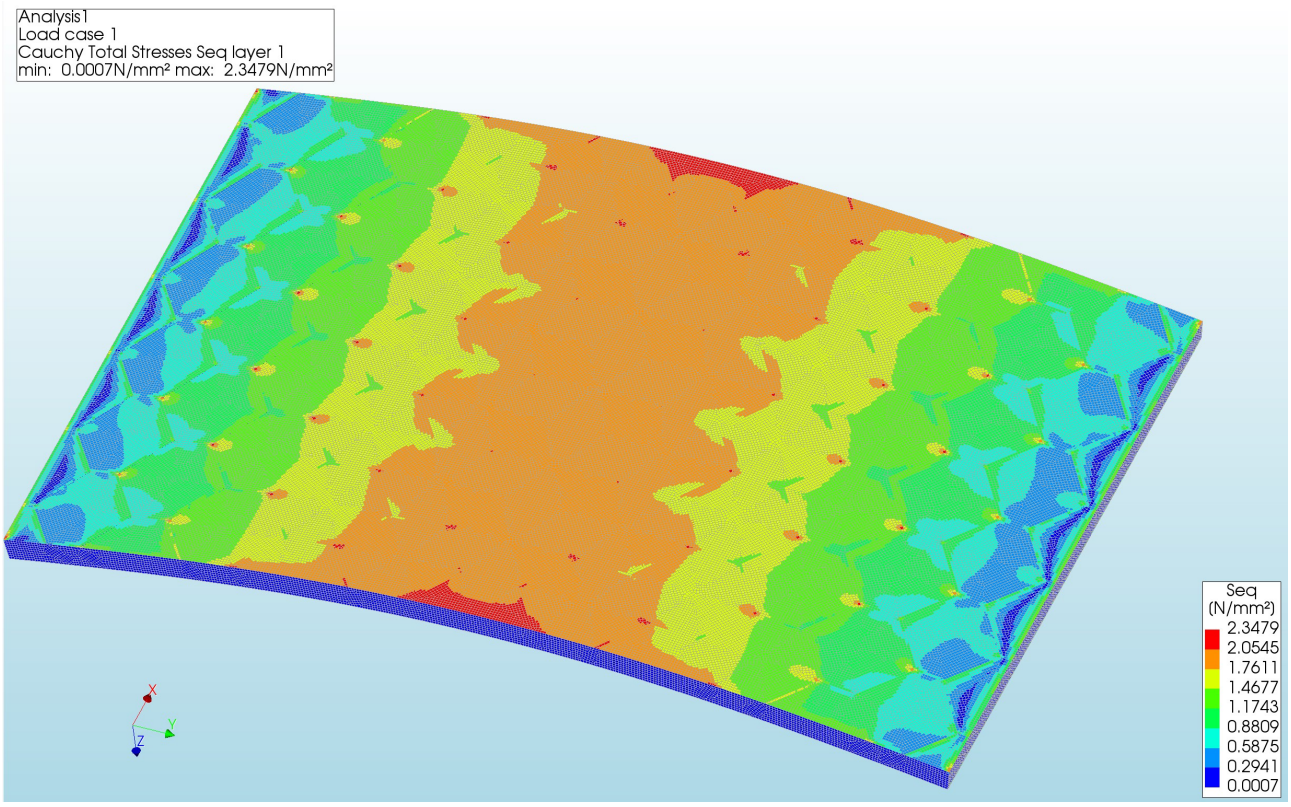


Figure 142: The Von Mises stresses of the bottom side of Pattern 2 with an element size of 2 millimetre

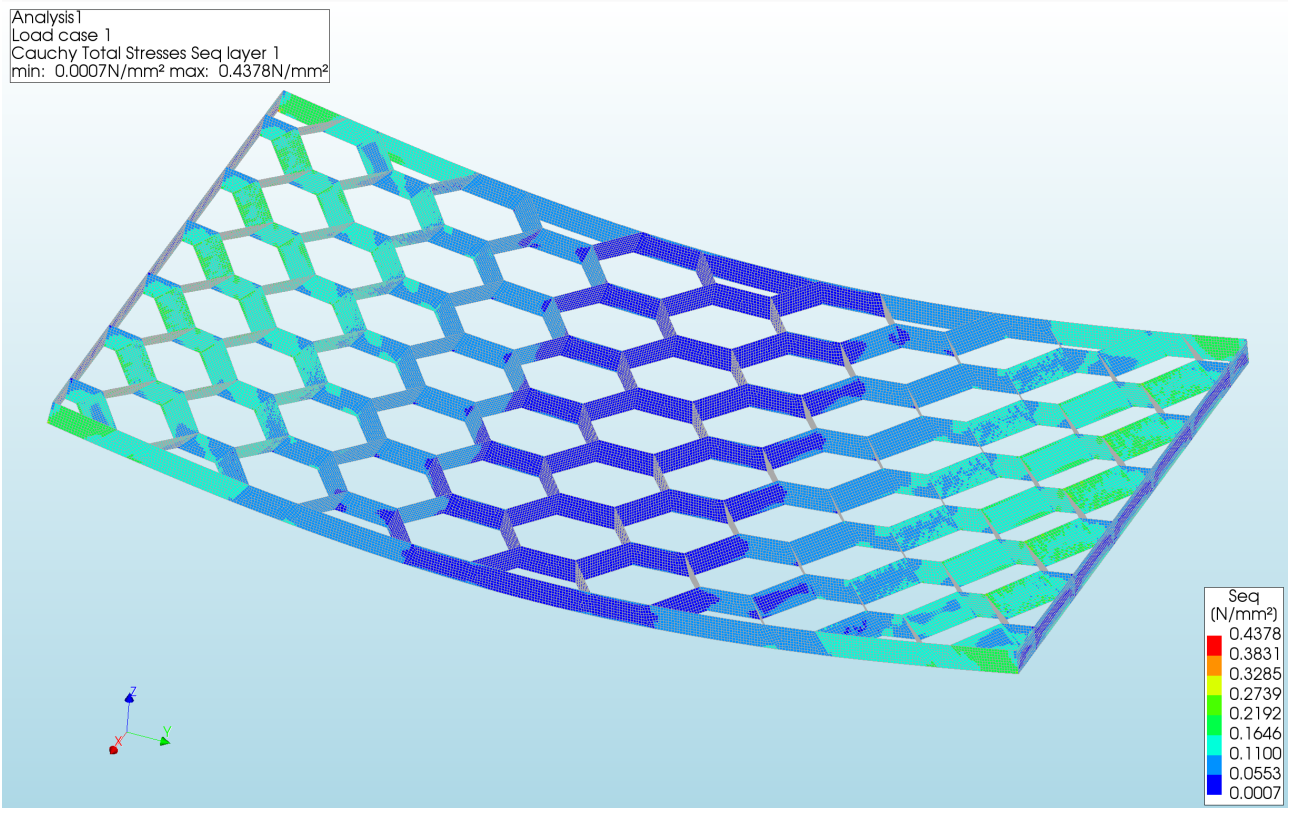


Figure 143: The Von Mises stresses of the Core of Pattern 2 with an element size of 2 millimetre

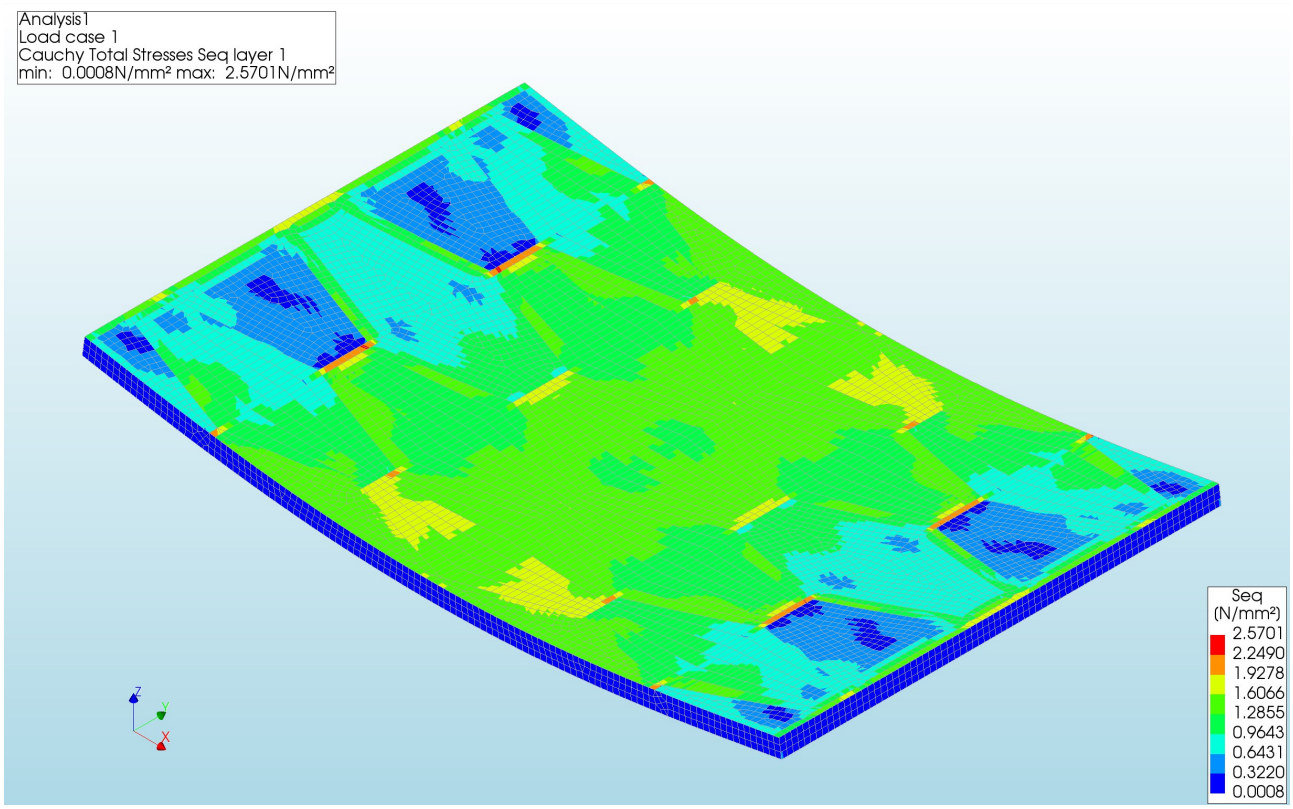


Figure 144: The Von Mises stresses of the top side of Pattern 3 with an element size of 6 millimetre

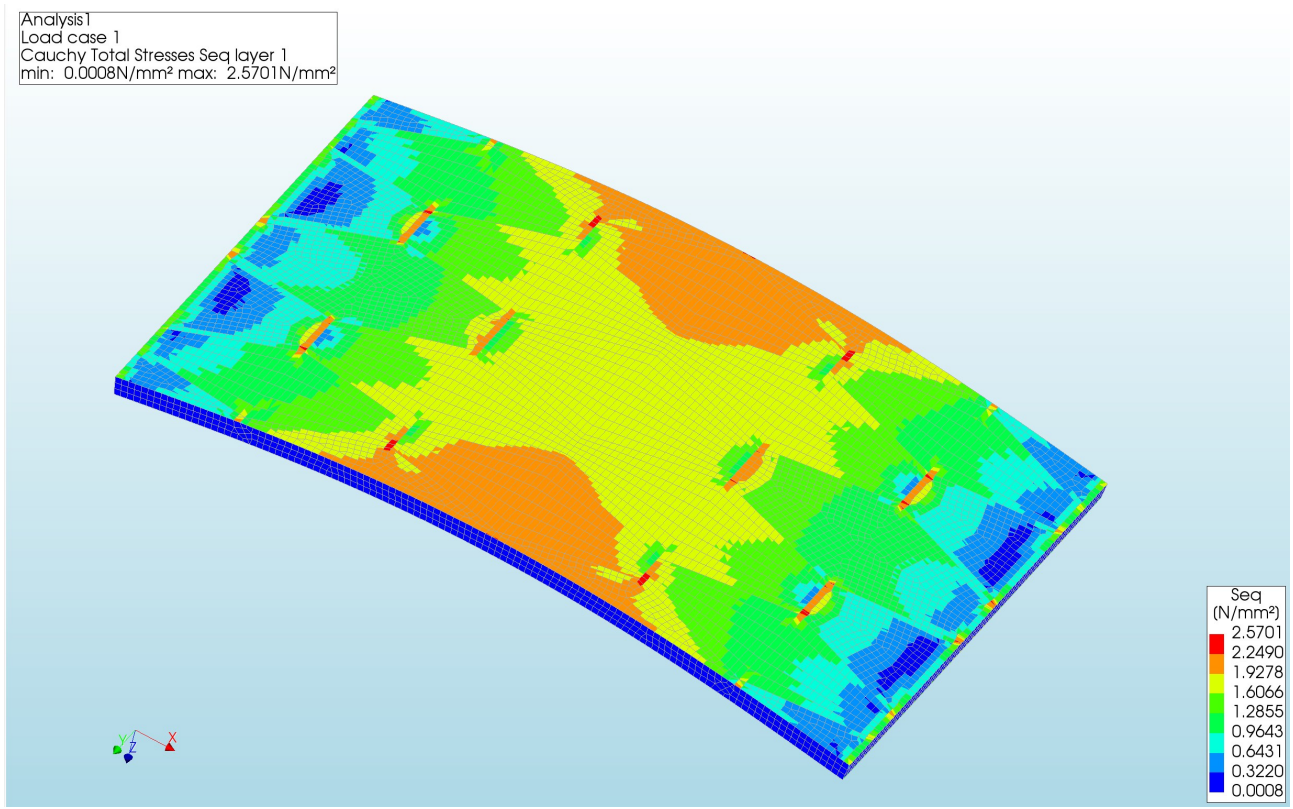


Figure 145: The Von Mises stresses of the bottom side of Pattern 3 with an element size of 6 millimetre

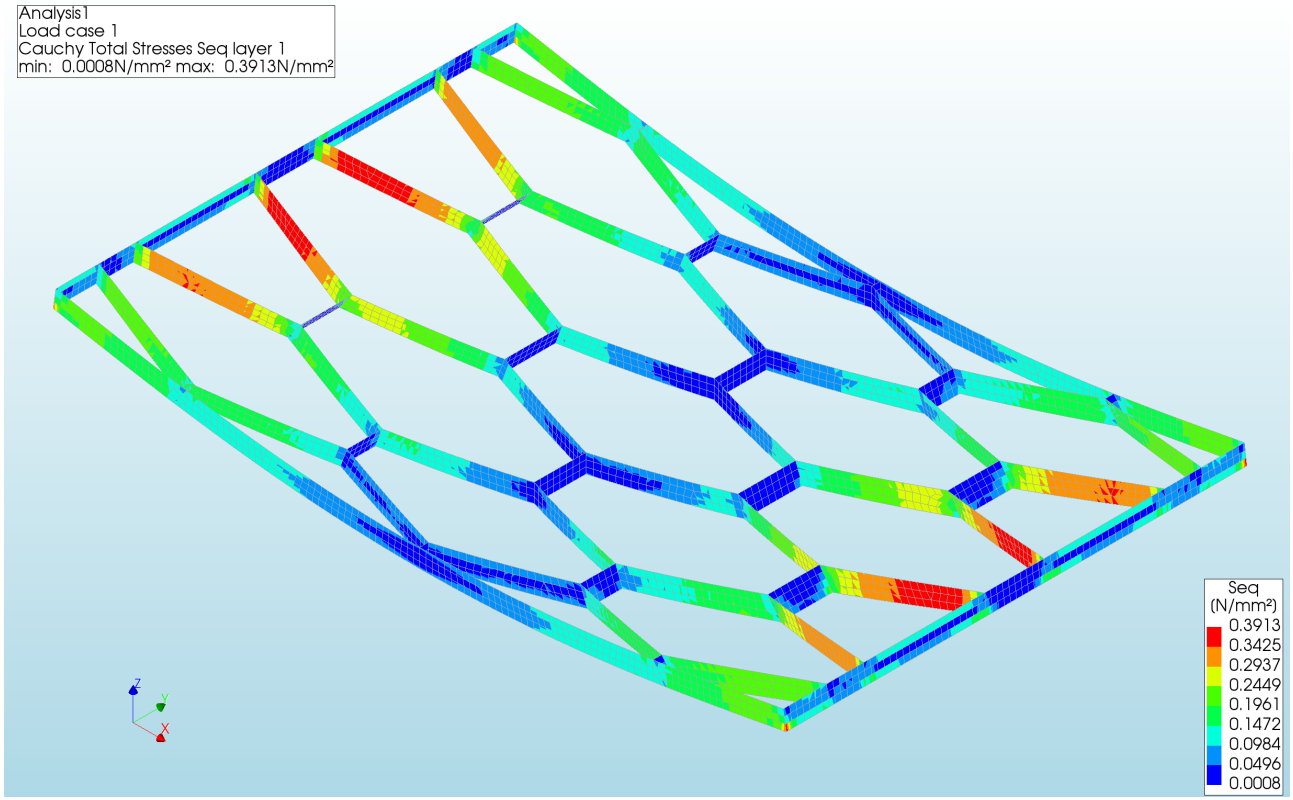


Figure 146: The Von Mises stresses of the Core of Pattern 3 with an element size of 6 millimetre

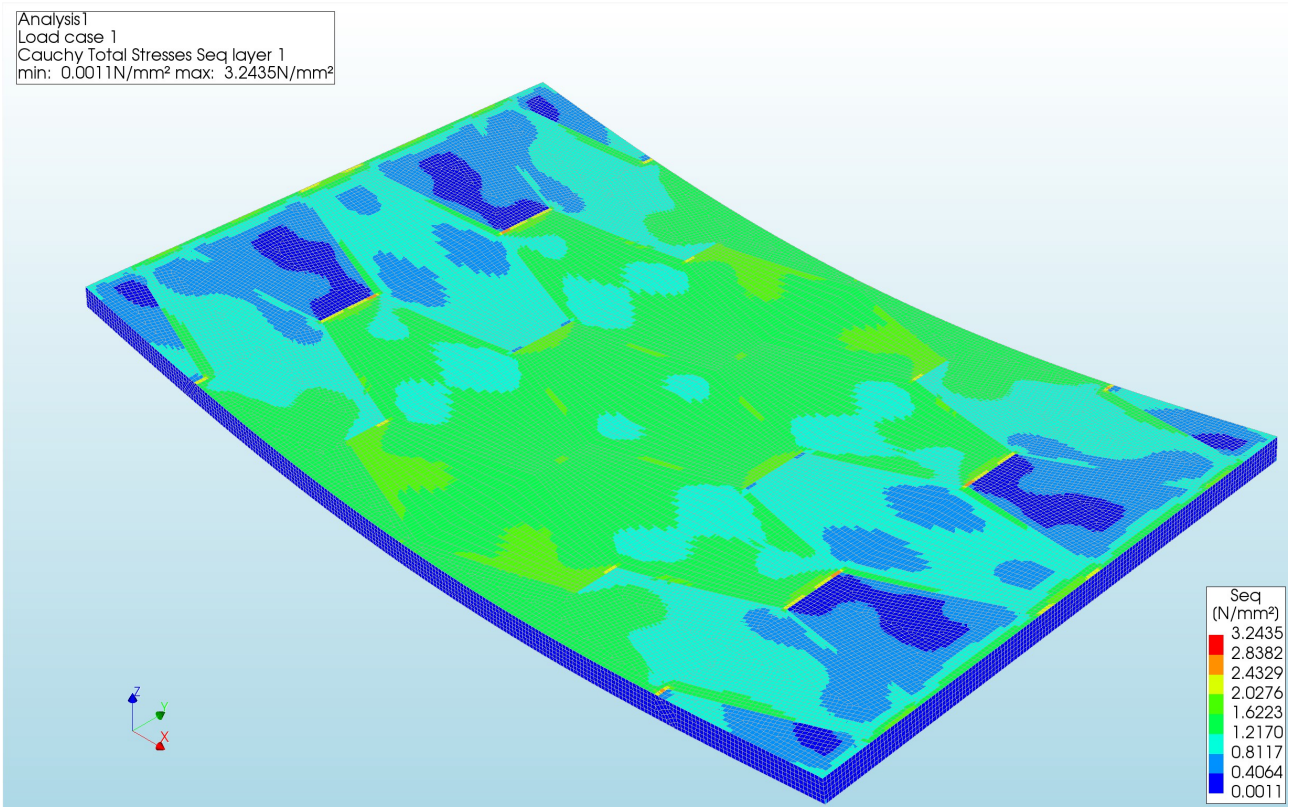


Figure 147: The Von Mises stresses of the top side of Pattern 3 with an element size of 3 millimetre

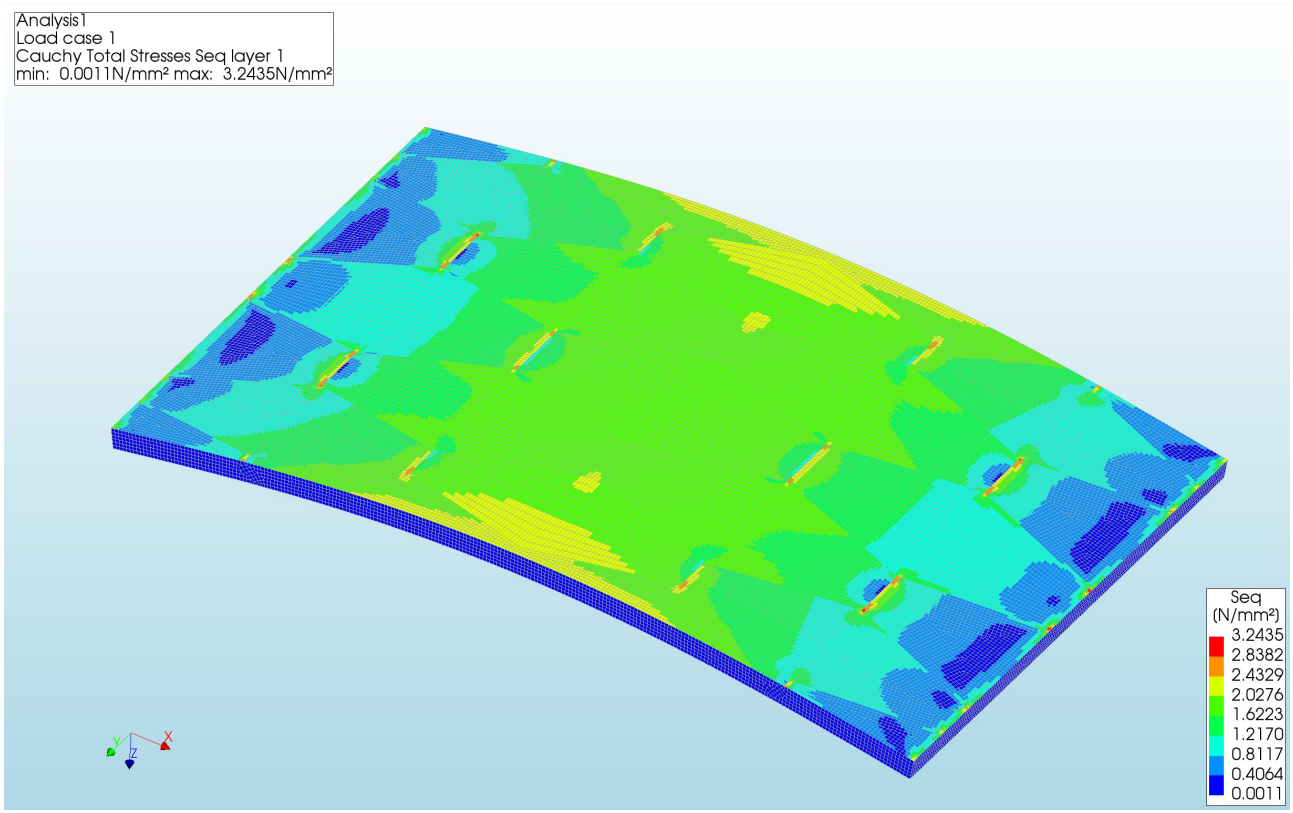


Figure 148: The Von Mises stresses of the bottom side of Pattern 3 with an element size of 3 millimetre

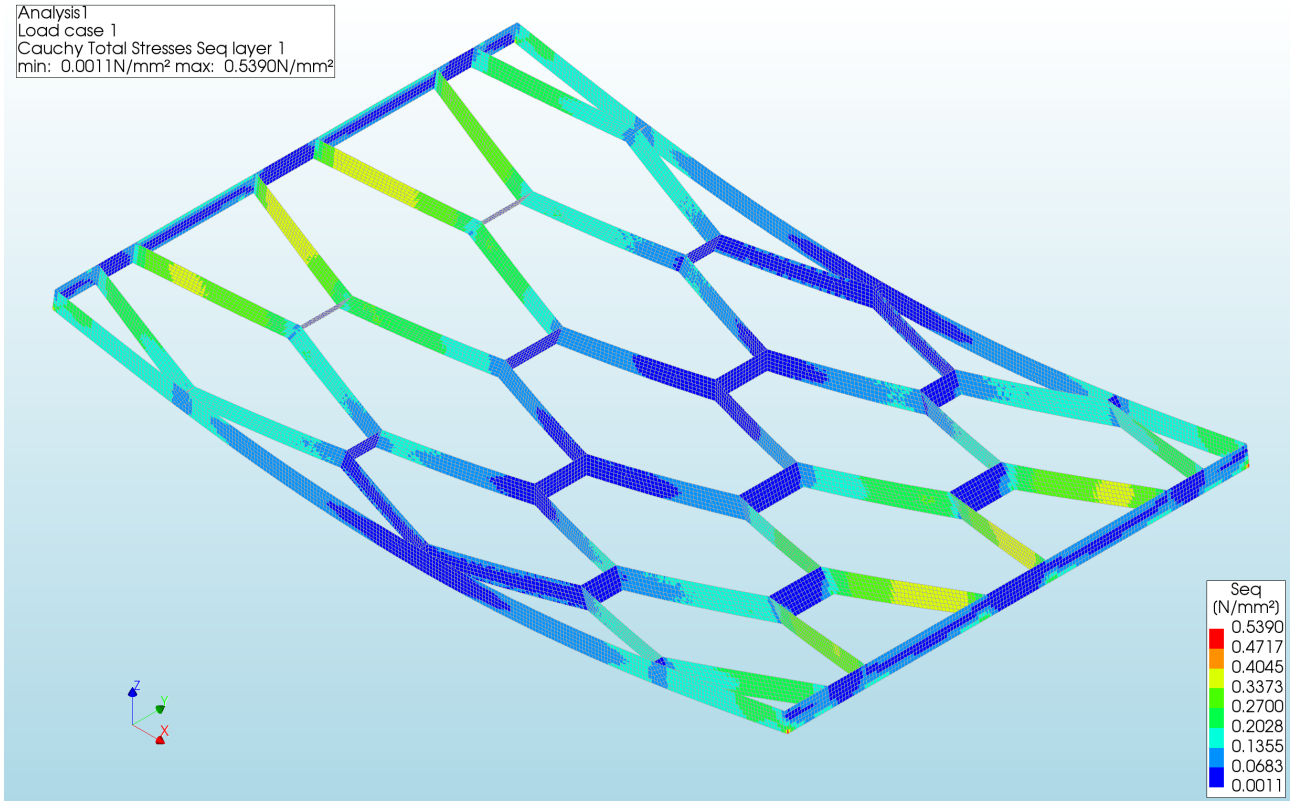


Figure 149: The Von Mises stresses of the Core of Pattern 3 with an element size of 3 millimetre

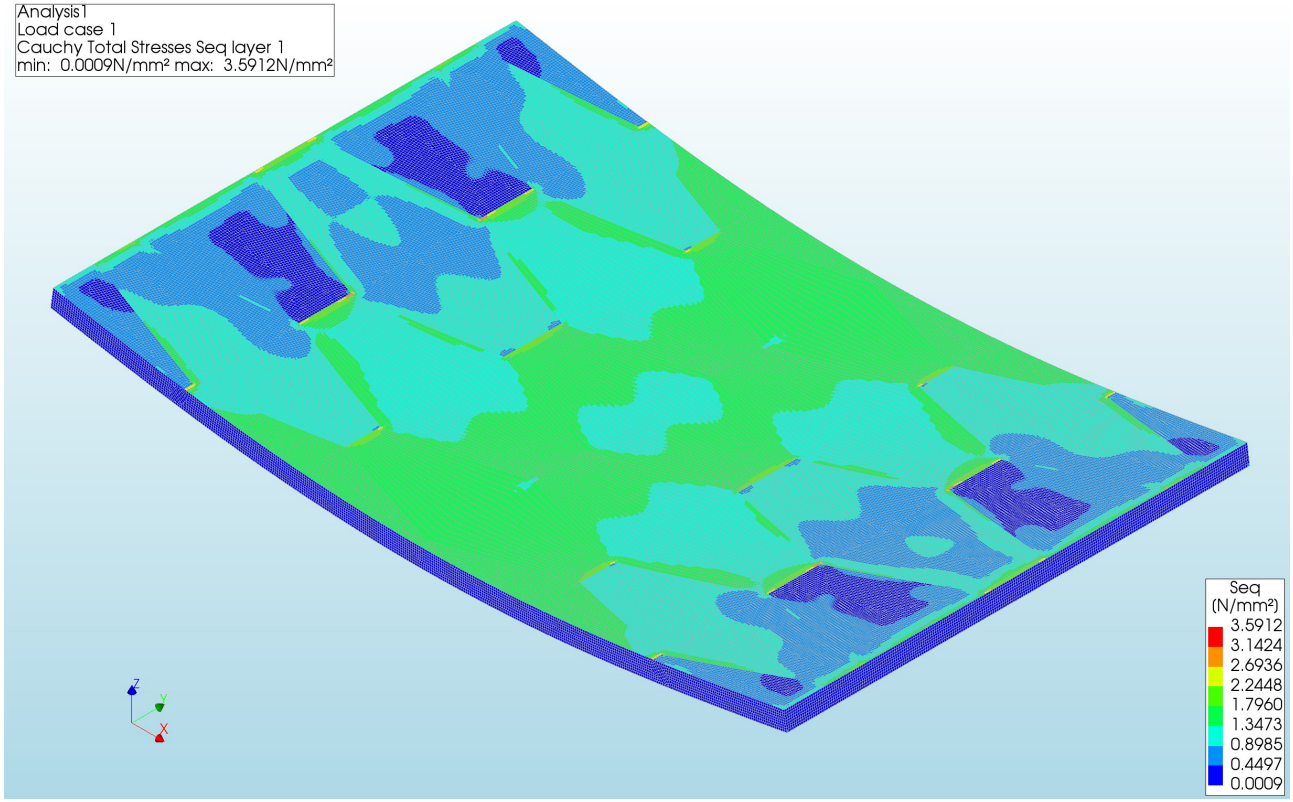


Figure 150: The Von Mises stresses of the top side of Pattern 3 with an element size of 2 millimetre

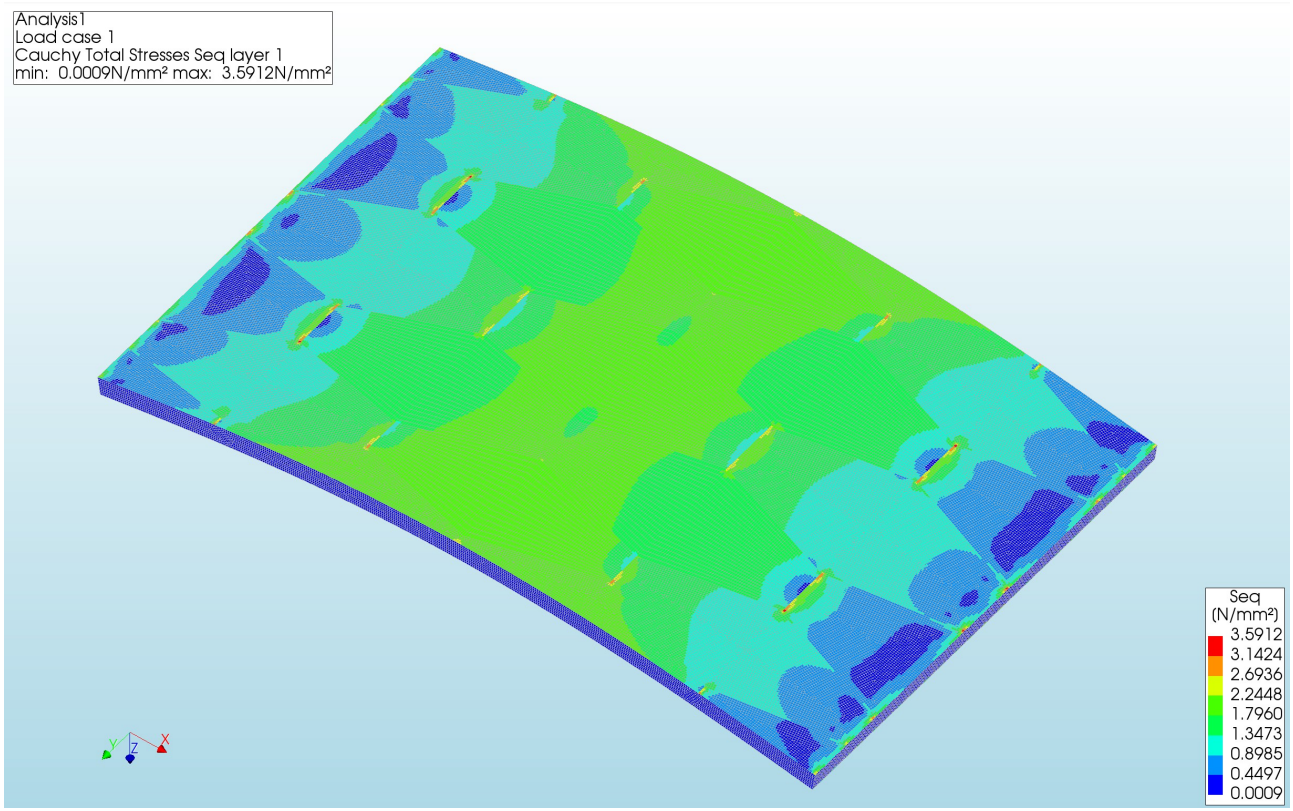


Figure 151: The Von Mises stresses of the bottom side of Pattern 3 with an element size of 2 millimetre

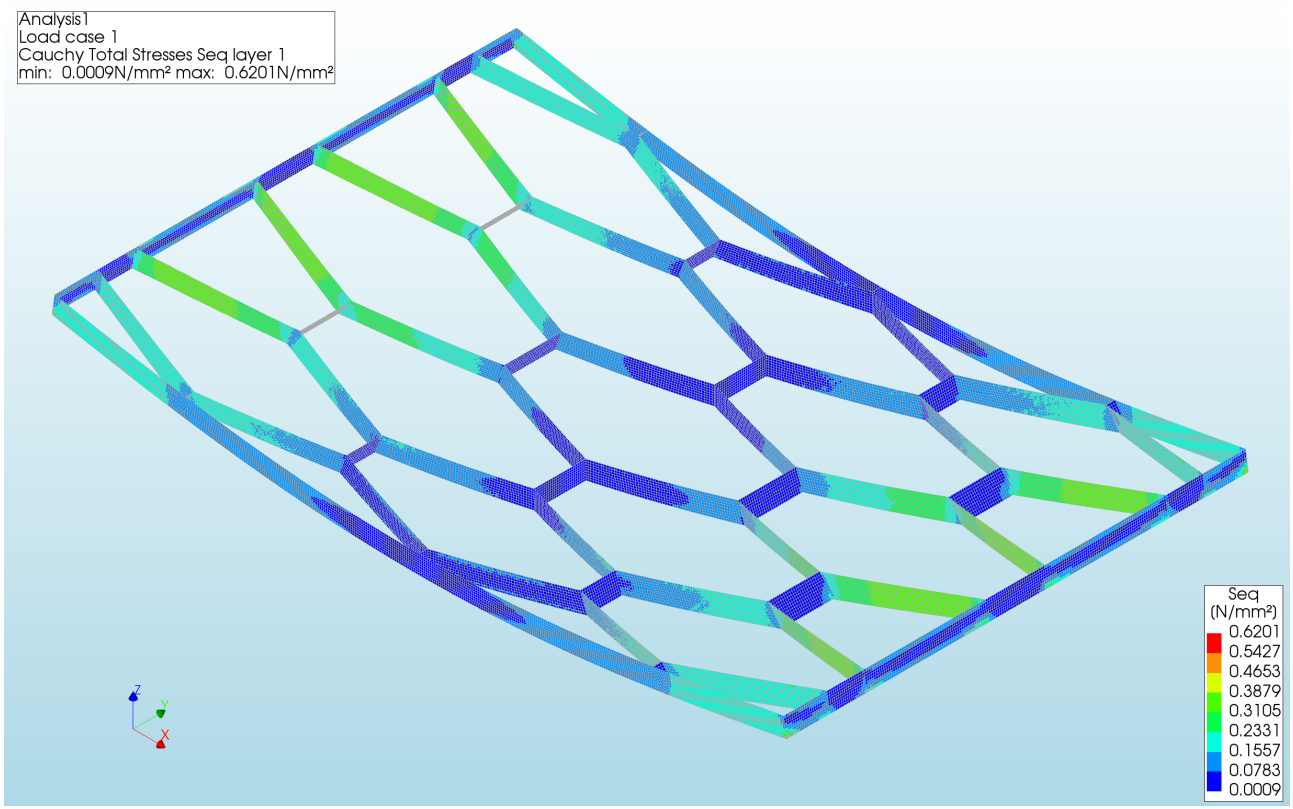


Figure 152: The Von Mises stresses of the Core of Pattern 3 with an element size of 2 millimetre

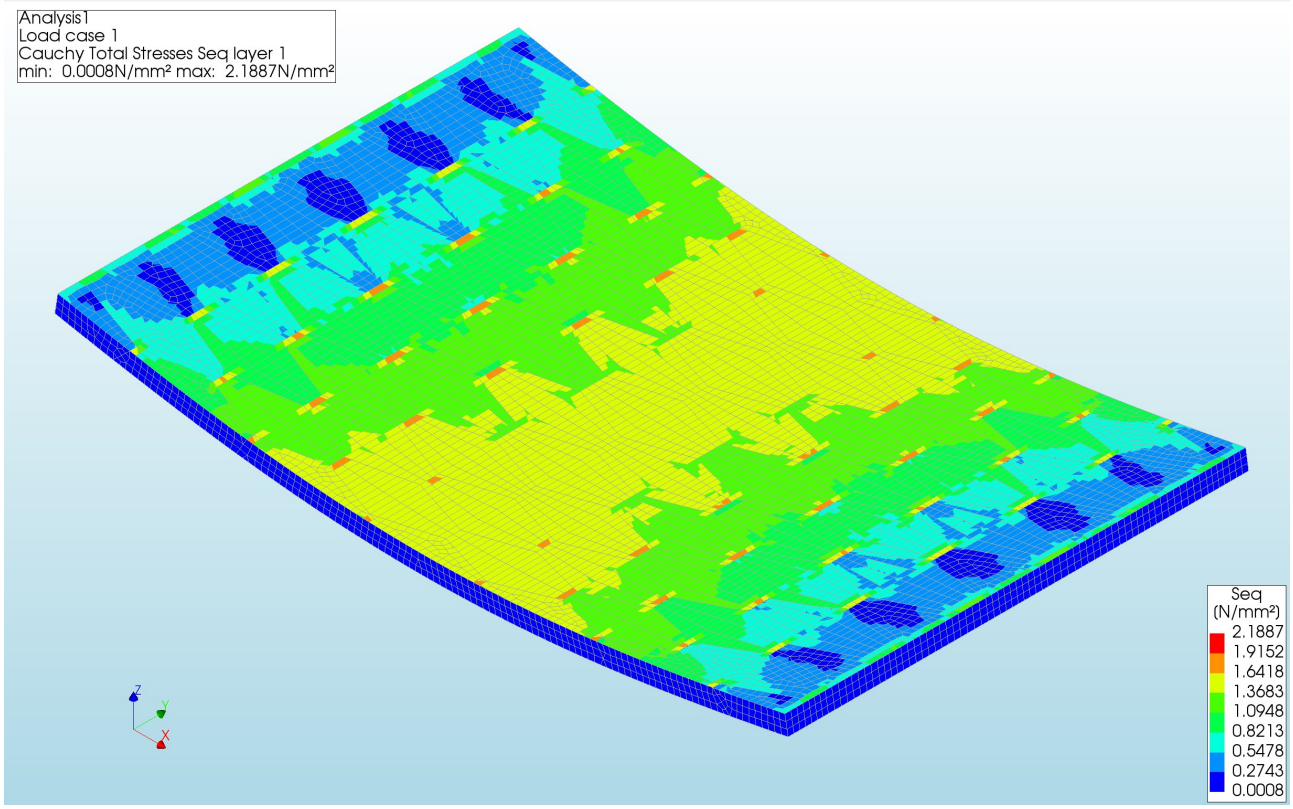


Figure 153: The Von Mises stresses of the top side of Pattern 4 with an element size of 6 millimetre

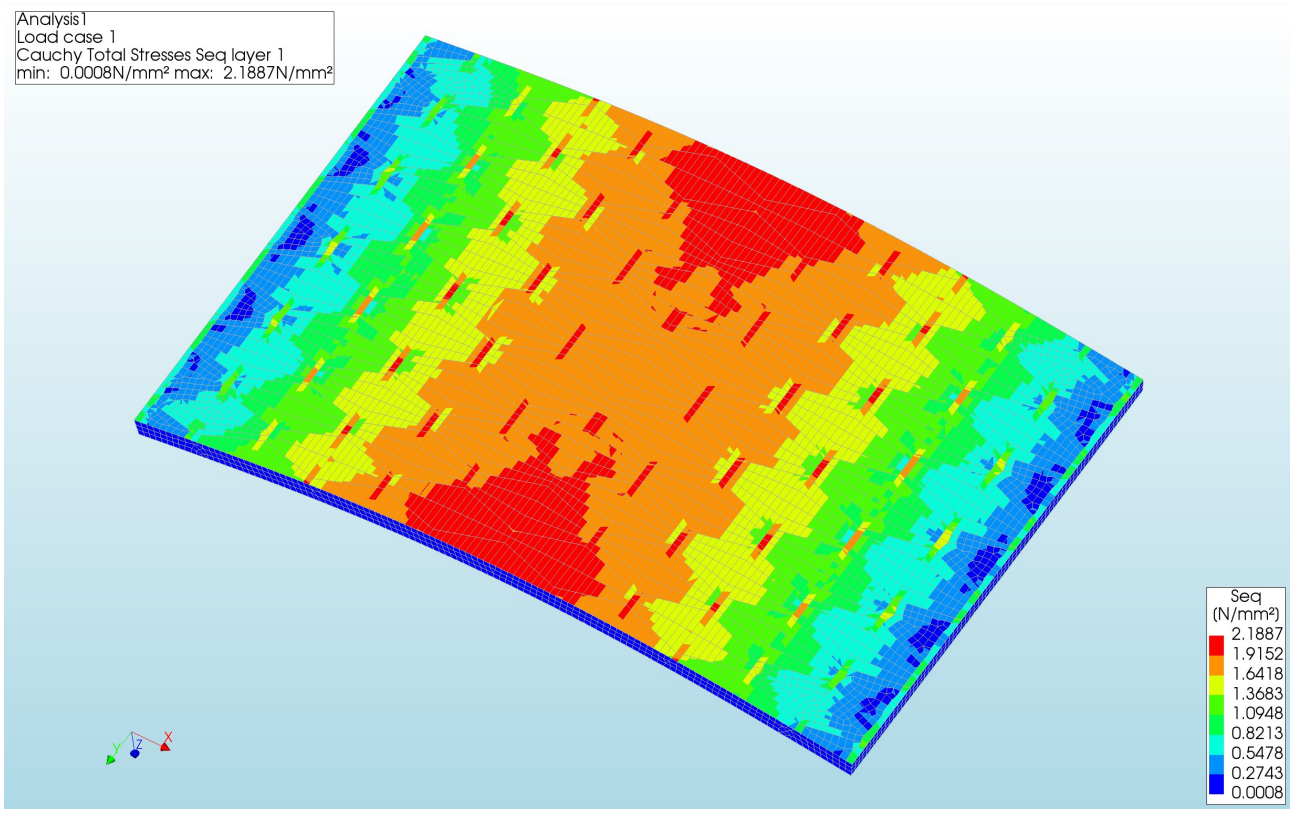


Figure 154: The Von Mises stresses of the bottom side of Pattern 4 with an element size of 6 millimetre

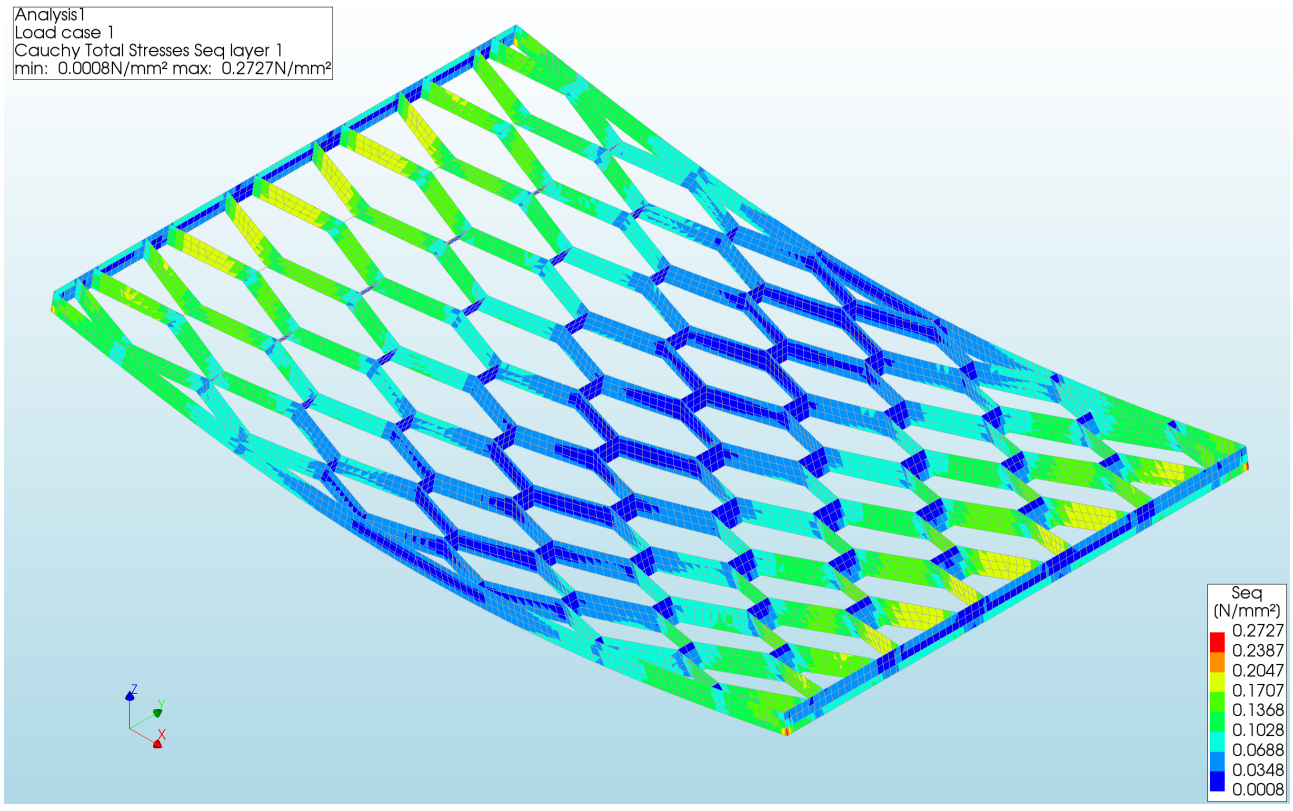


Figure 155: The Von Mises stresses of the Core of Pattern 4 with an element size of 6 millimetre

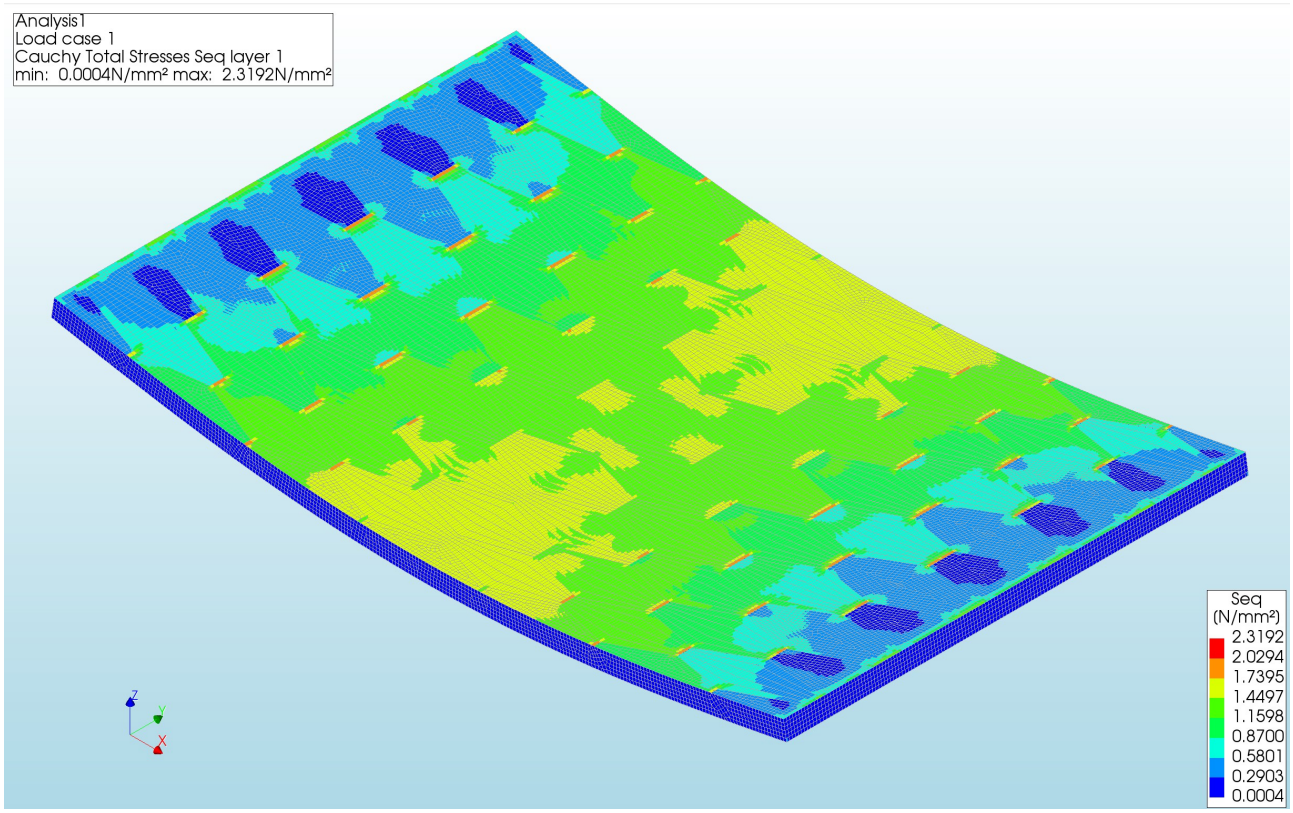


Figure 156: The Von Mises stresses of the top side of Pattern 4 with an element size of 3 millimetre

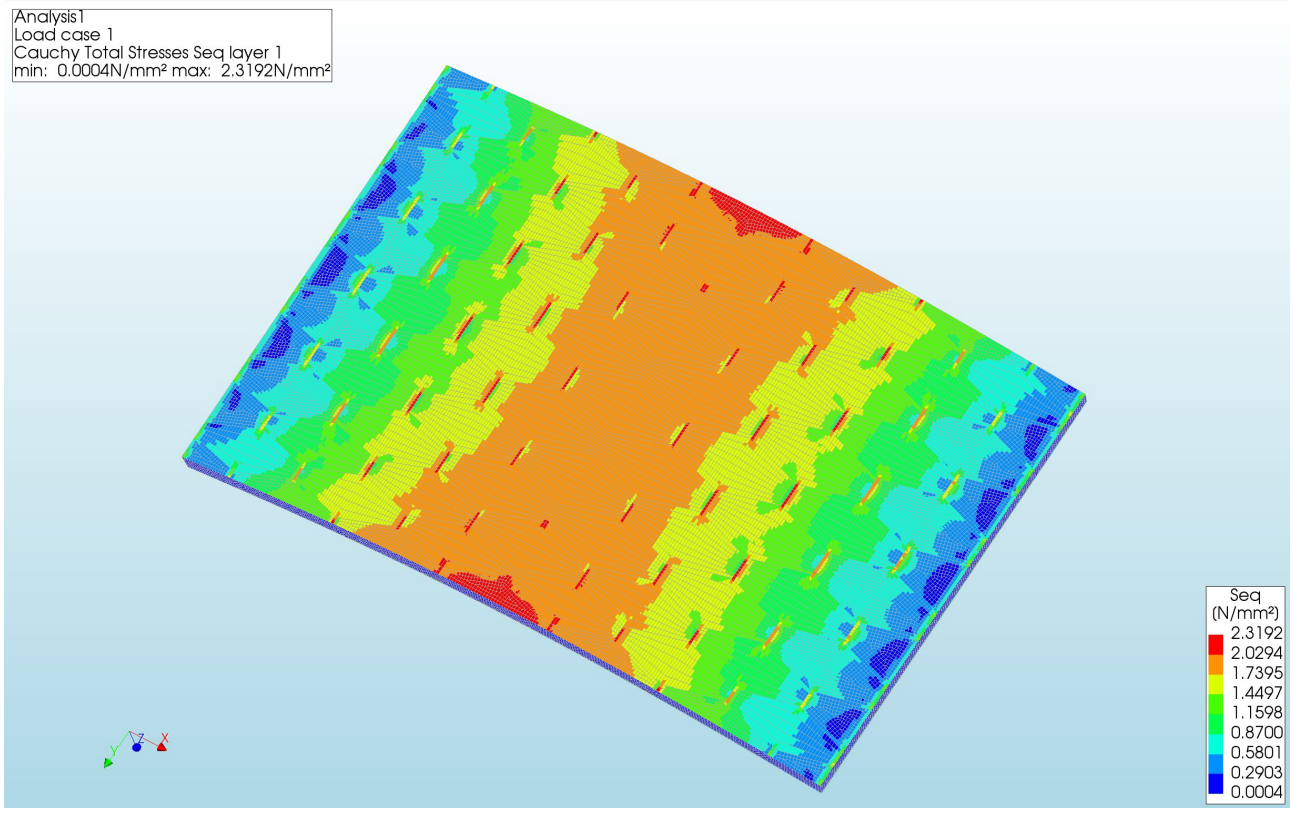


Figure 157: The Von Mises stresses of the bottom side of Pattern 4 with an element size of 3 millimetre

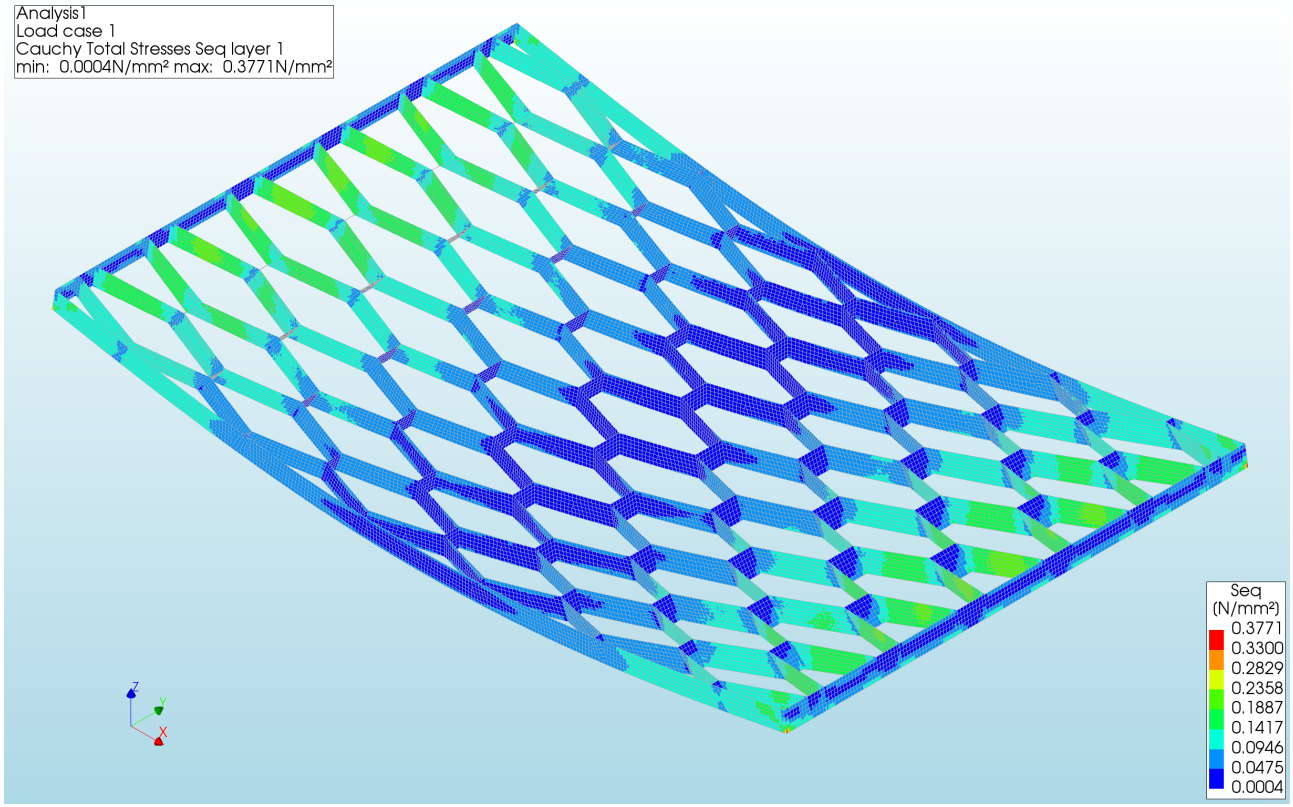


Figure 158: The Von Mises stresses of the Core of Pattern 4 with an element size of 3 millimetre

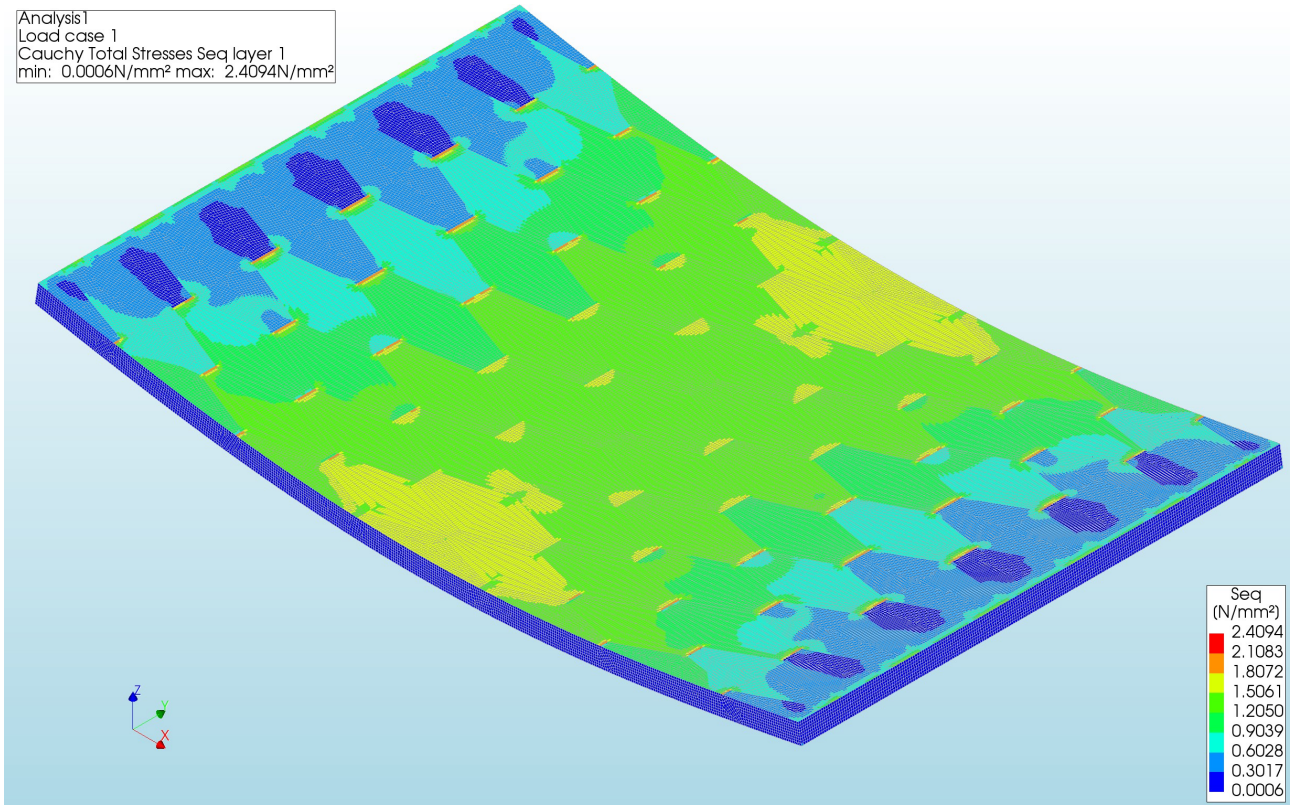


Figure 159: The Von Mises stresses of the top side of Pattern 4 with an element size of 2 millimetre

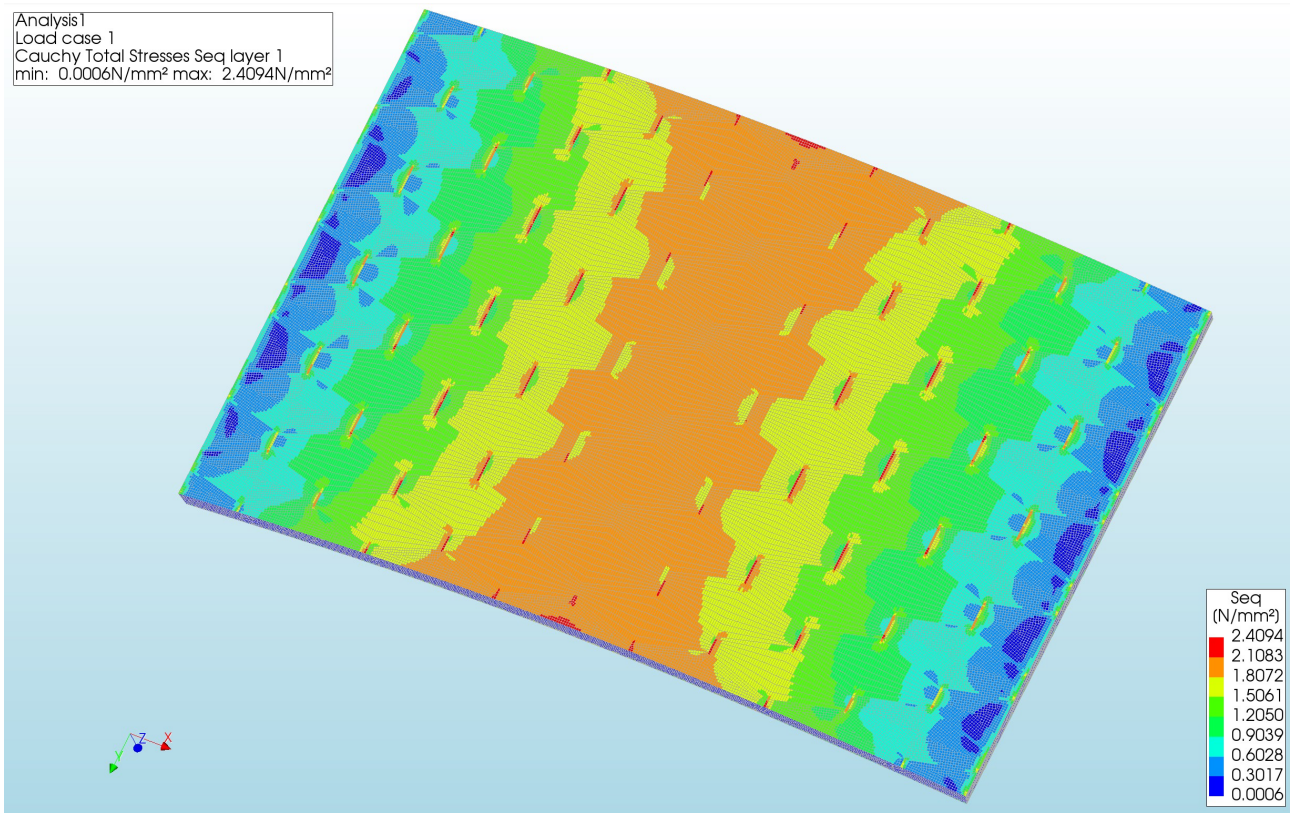


Figure 160: The Von Mises stresses of the bottom side of Pattern 4 with an element size of 2 millimetre

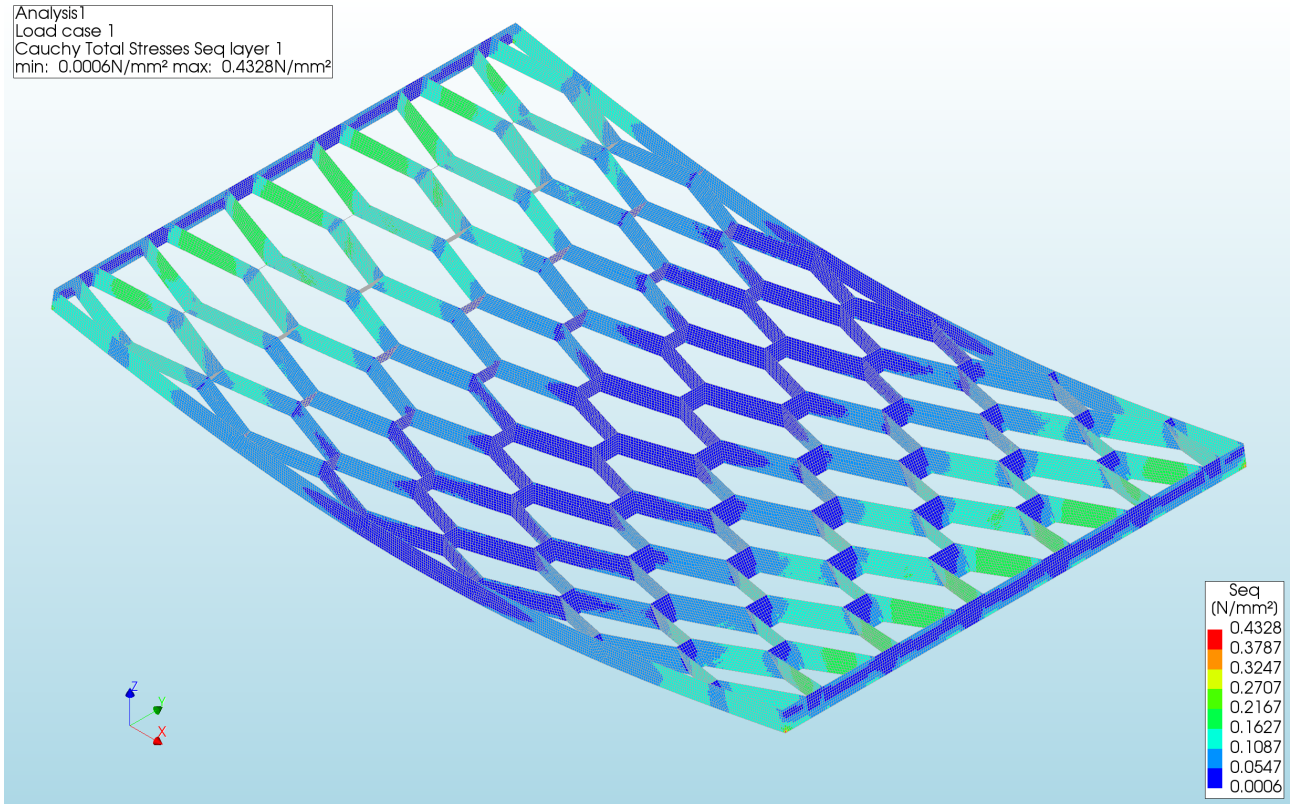


Figure 161: The Von Mises stresses of the Core of Pattern 4 with an element size of 2 millimetre

Appendix E: Visual representation stresses of DIANA with solid elements

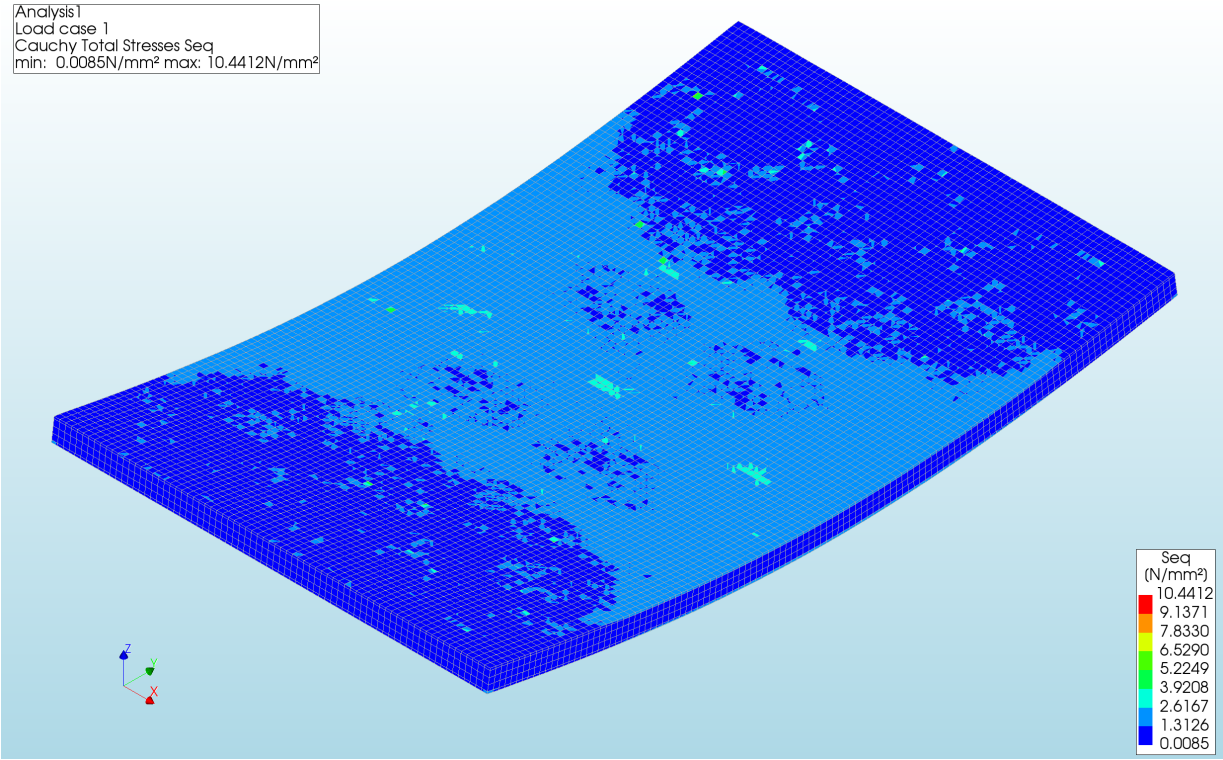


Figure 162: The Von Mises stresses of the top side of Pattern 1 with an element size of 6 millimetre

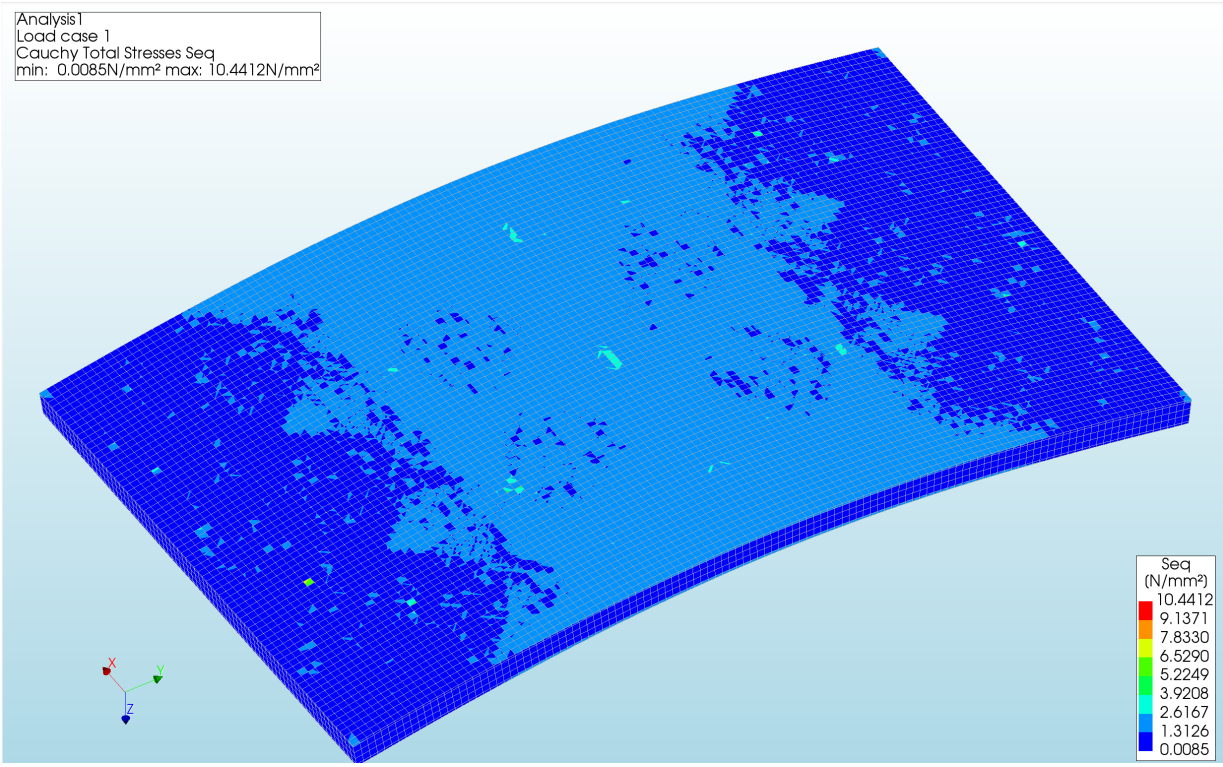


Figure 163: The Von Mises stresses of the bottom side of Pattern 1 with an element size of 6 millimetre

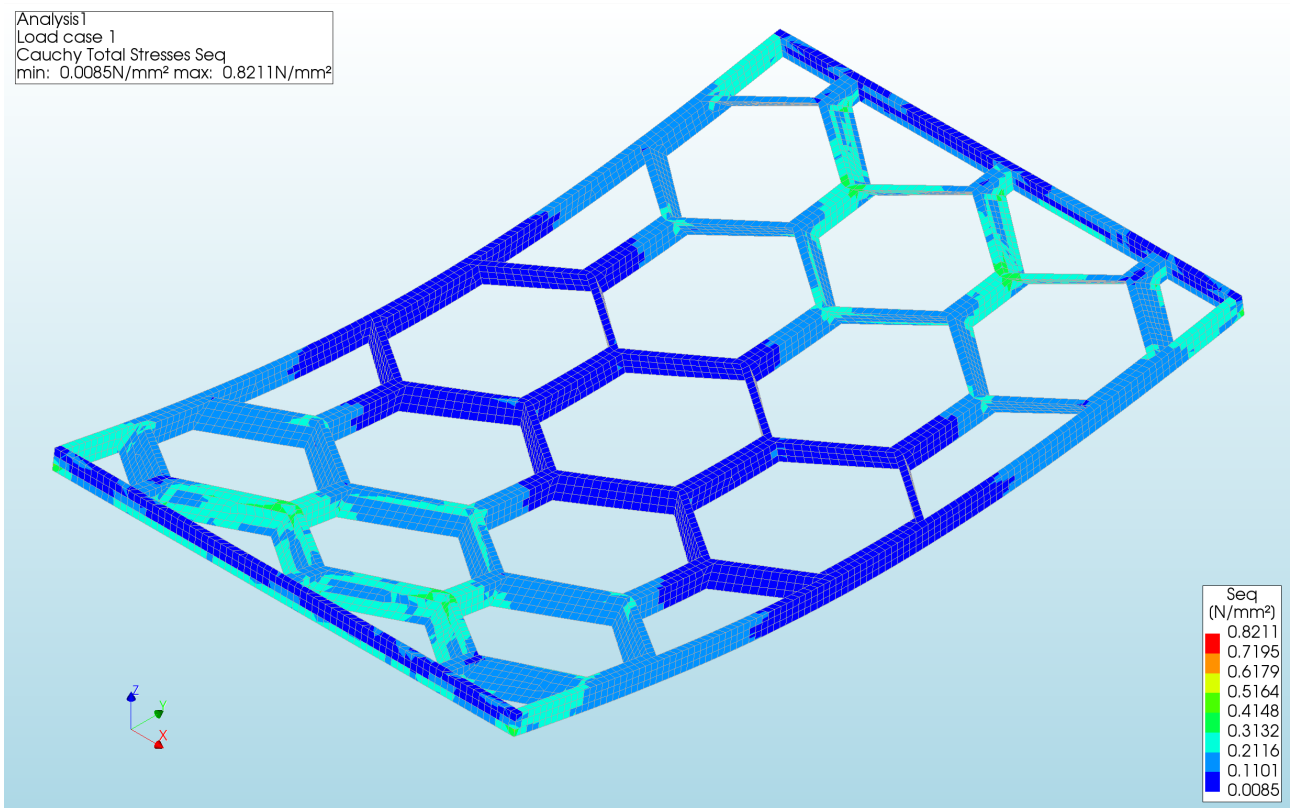


Figure 164: The Von Mises stresses of the Core of Pattern 1 with an element size of 6 millimetre

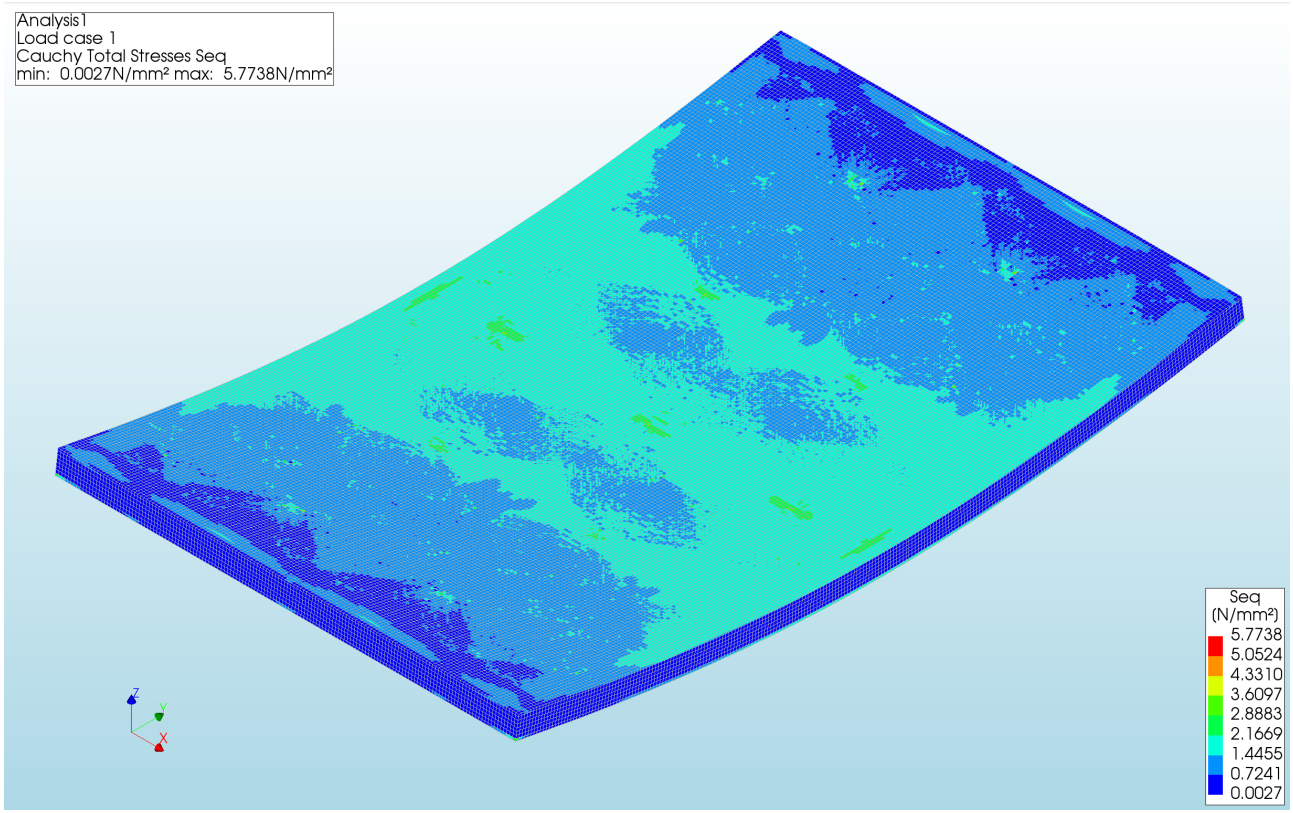


Figure 165: The Von Mises stresses of the top side of Pattern 1 with an element size of 3 millimetre

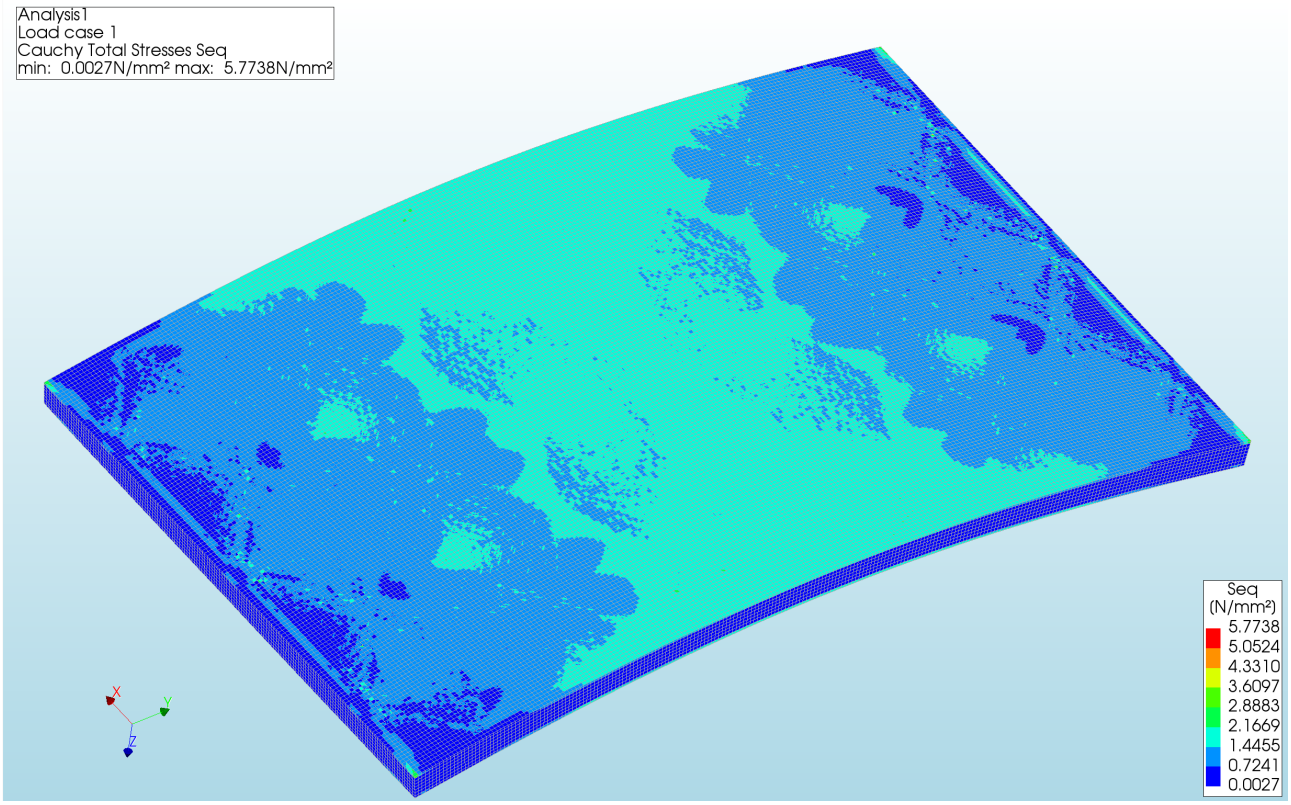


Figure 166: The Von Mises stresses of the bottom side of Pattern 1 with an element size of 3 millimetre

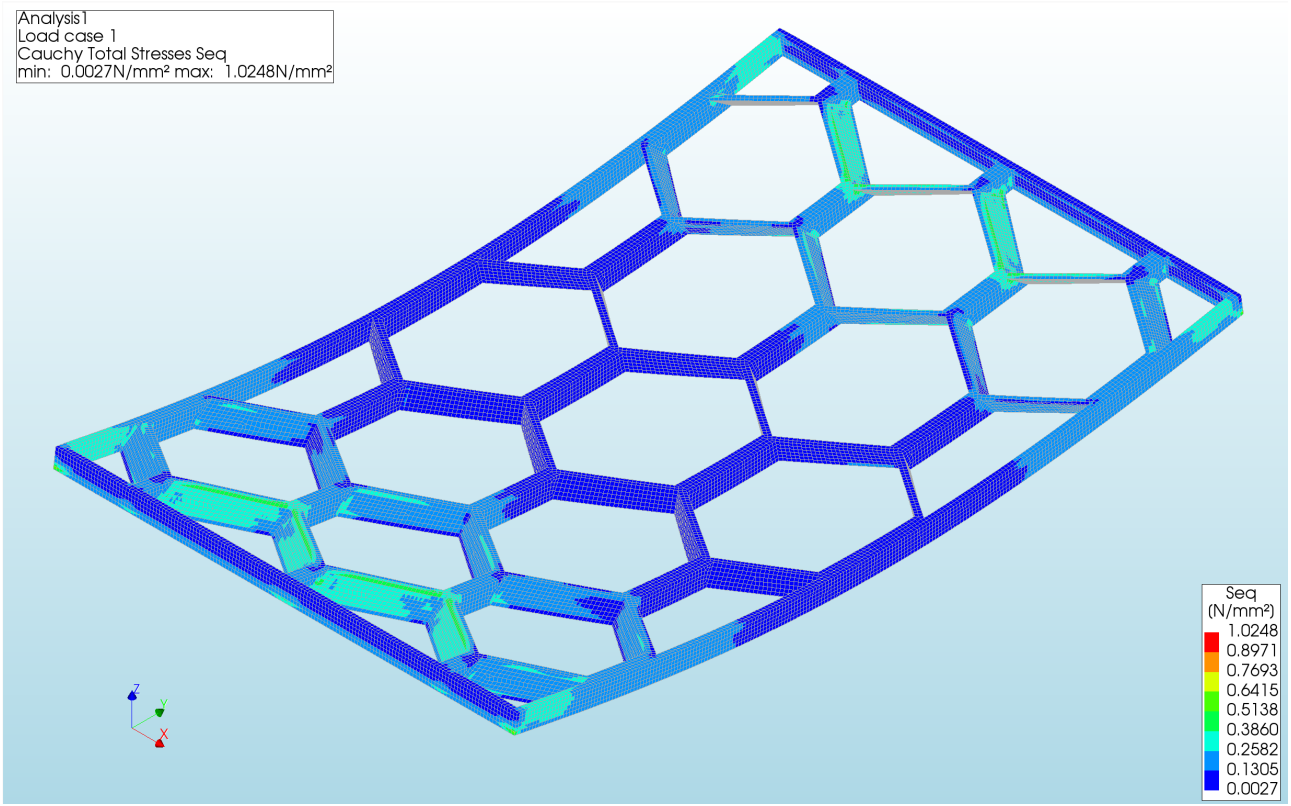


Figure 167: The Von Mises stresses of the Core of Pattern 1 with an element size of 3 millimetre

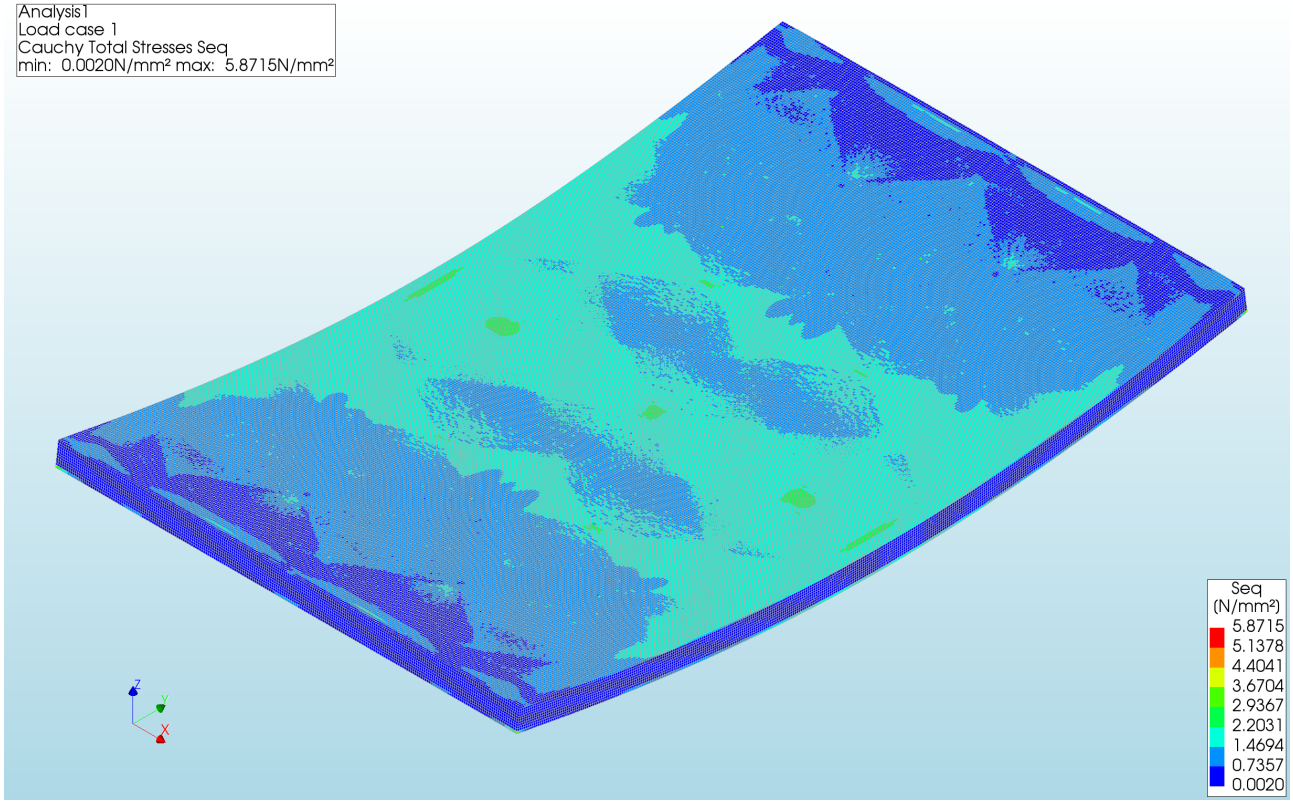


Figure 168: The Von Mises stresses of the top side of Pattern 1 with an element size of 2 millimetre

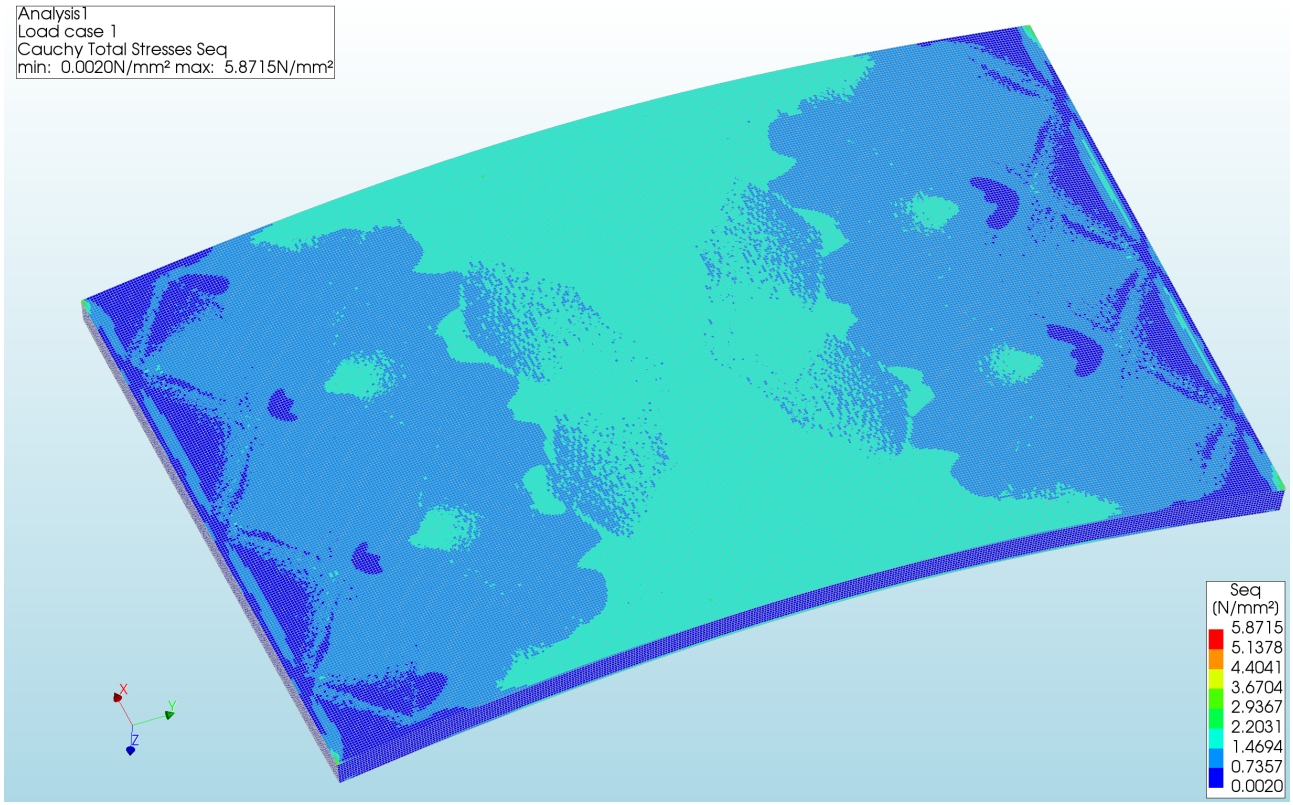


Figure 169: The Von Mises stresses of the bottom side of Pattern 1 with an element size of 2 millimetre

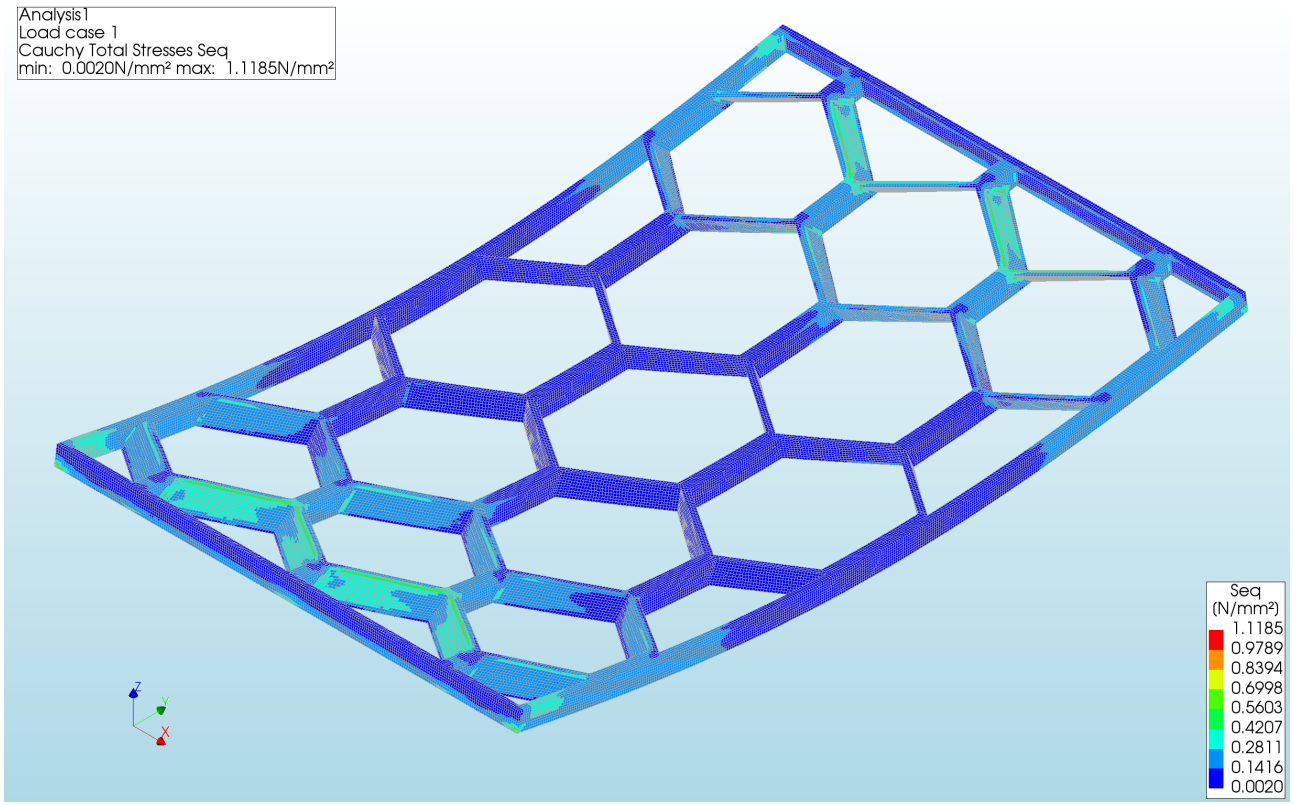


Figure 170: The Von Mises stresses of the Core of Pattern 1 with an element size of 2 millimetre

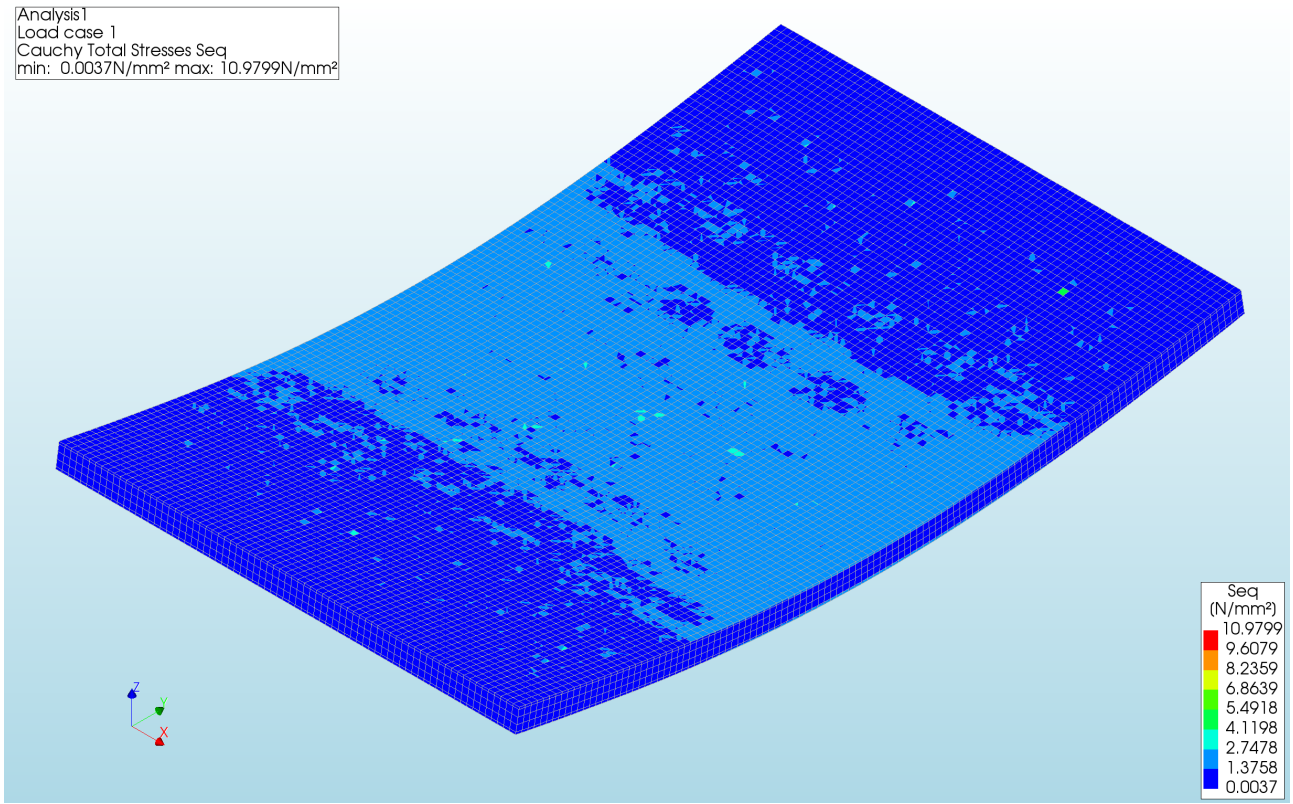


Figure 171: The Von Mises stresses of the top side of Pattern 2 with an element size of 6 millimetre

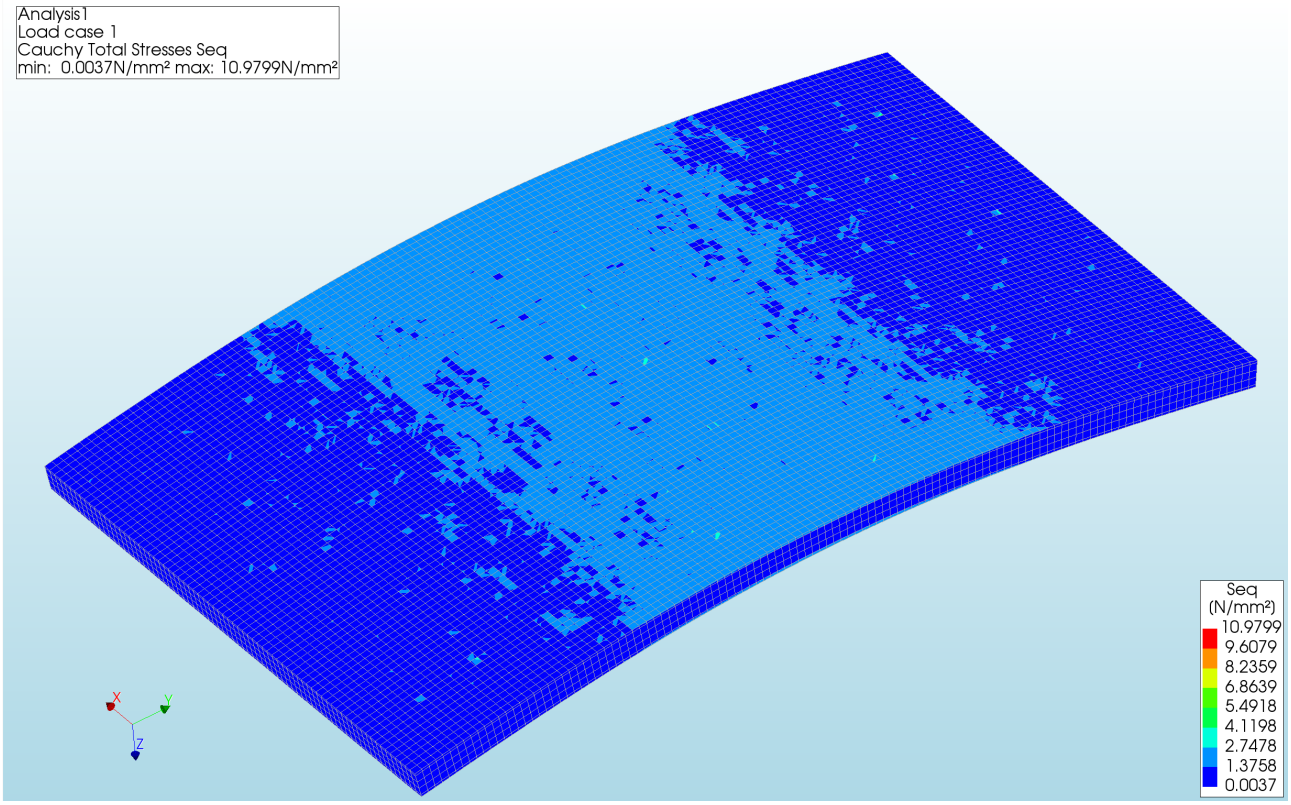


Figure 172: The Von Mises stresses of the bottom side of Pattern 2 with an element size of 6 millimetre

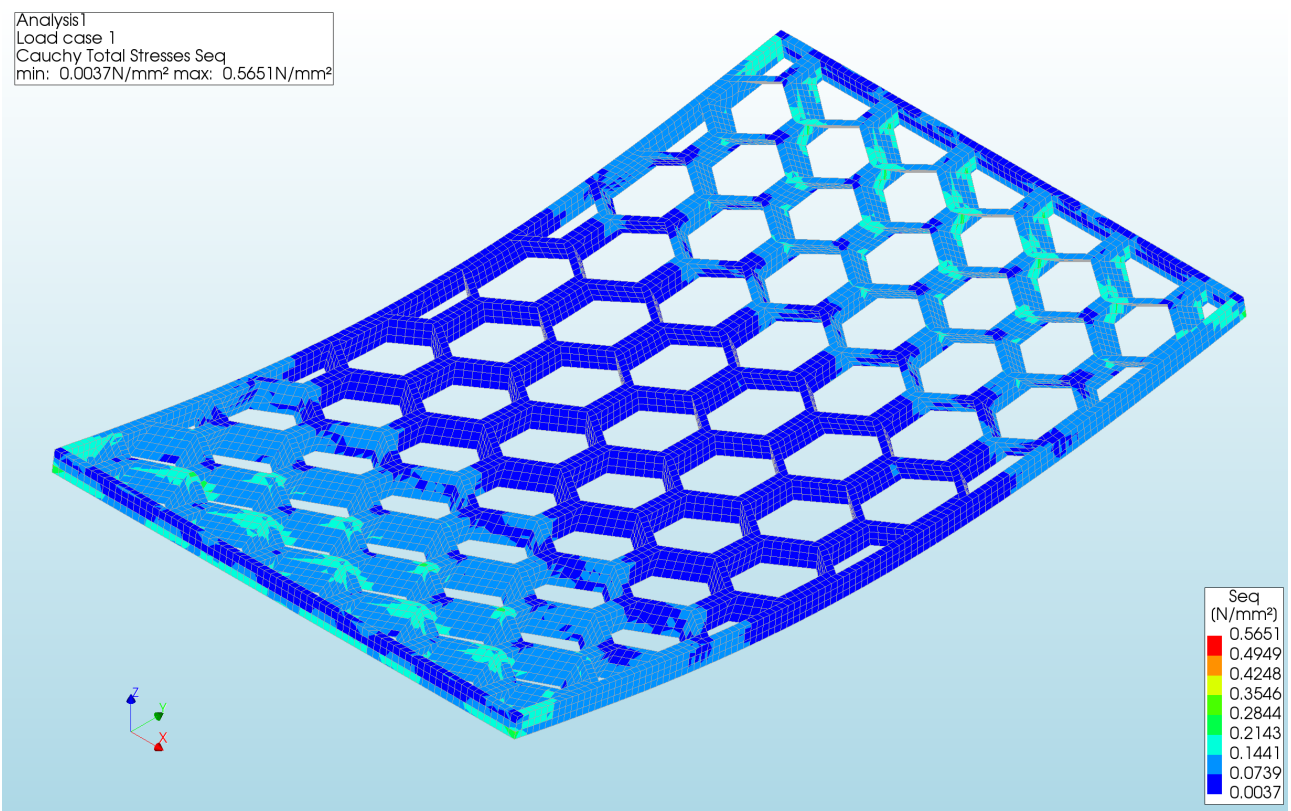


Figure 173: The Von Mises stresses of the Core of Pattern 2 with an element size of 6 millimetre

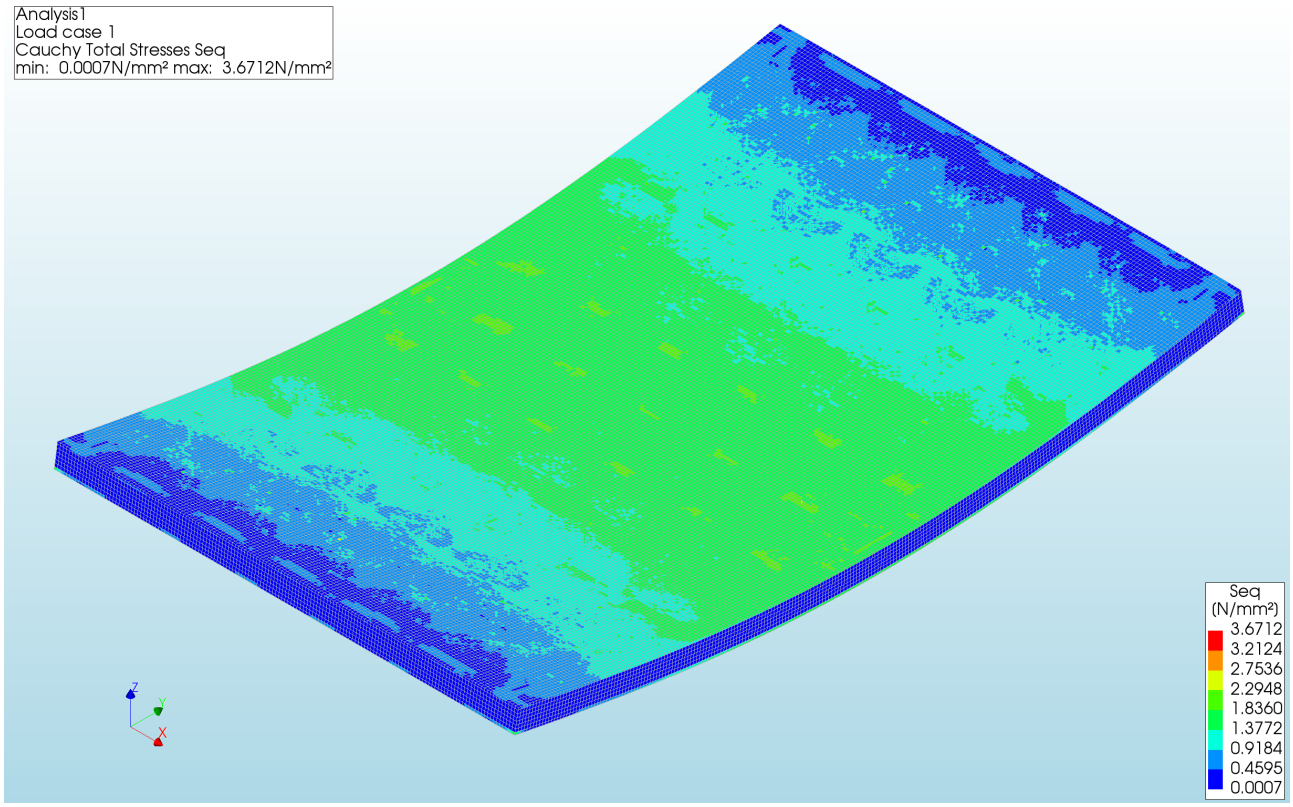


Figure 174: The Von Mises stresses of the top side of Pattern 2 with an element size of 3 millimetre

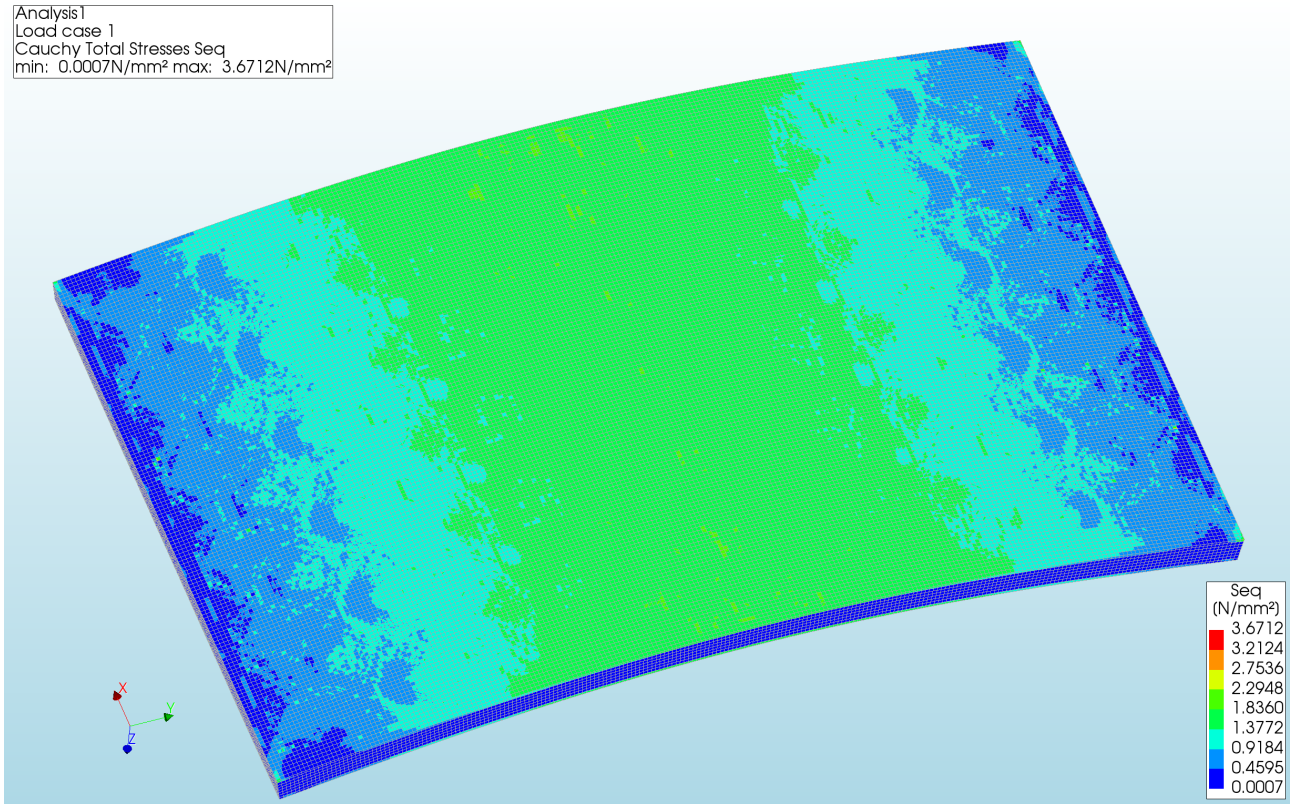


Figure 175: The Von Mises stresses of the bottom side of Pattern 2 with an element size of 3 millimetre

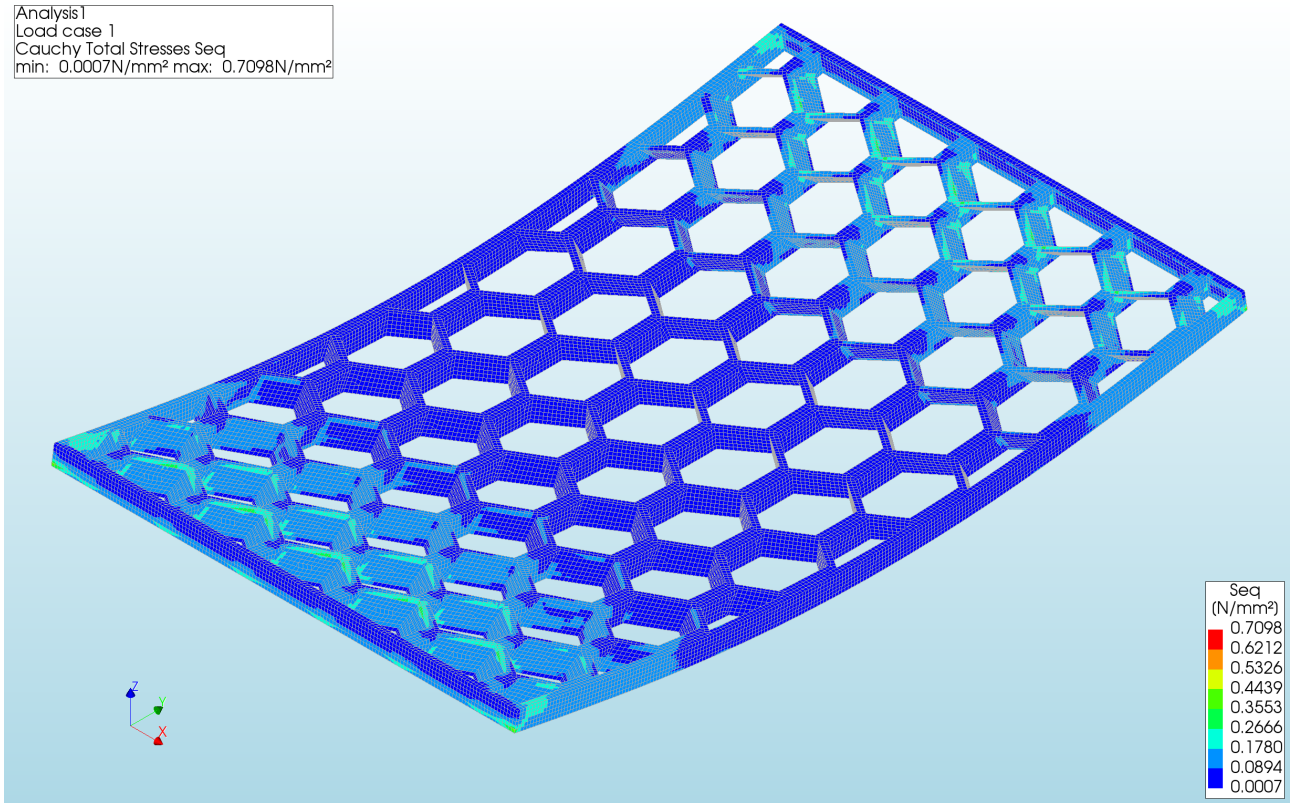


Figure 176: The Von Mises stresses of the Core of Pattern 2 with an element size of 3 millimetre

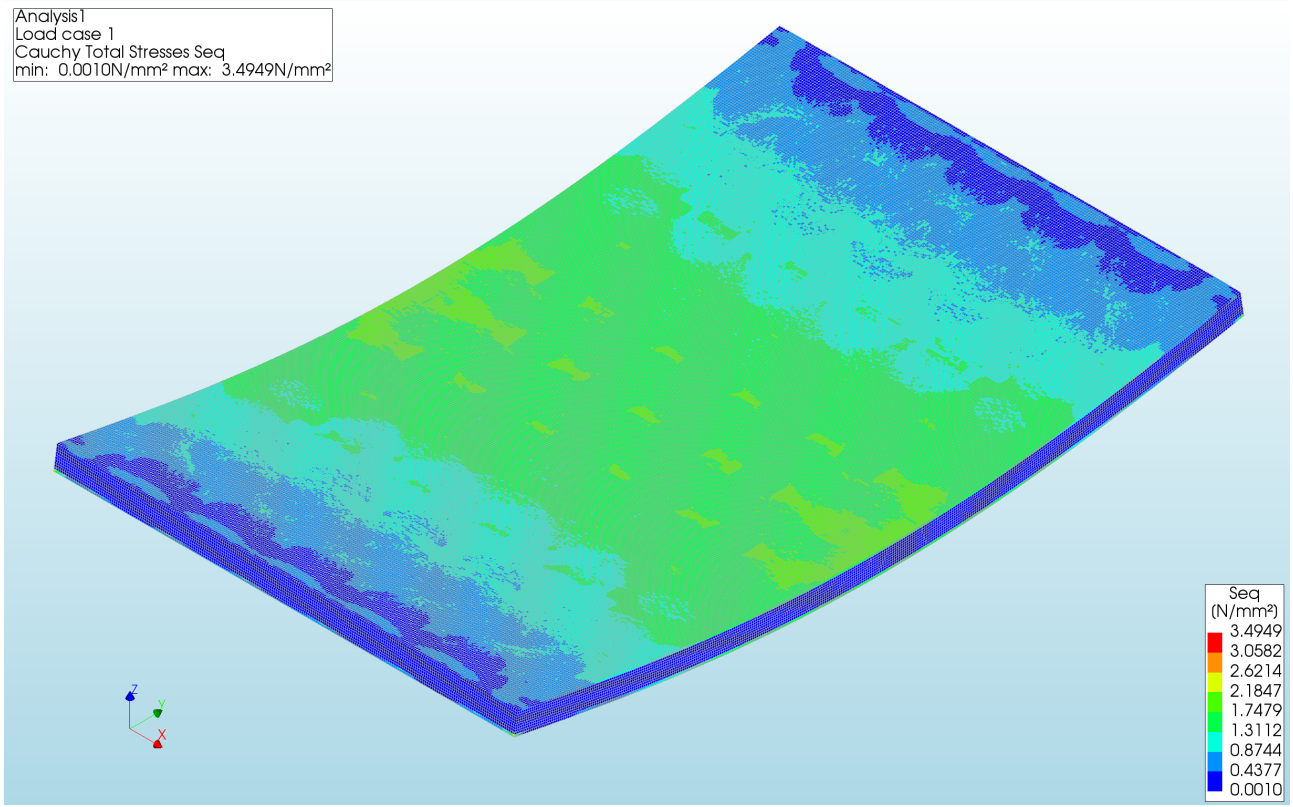


Figure 177: The Von Mises stresses of the top side of Pattern 2 with an element size of 2 millimetre

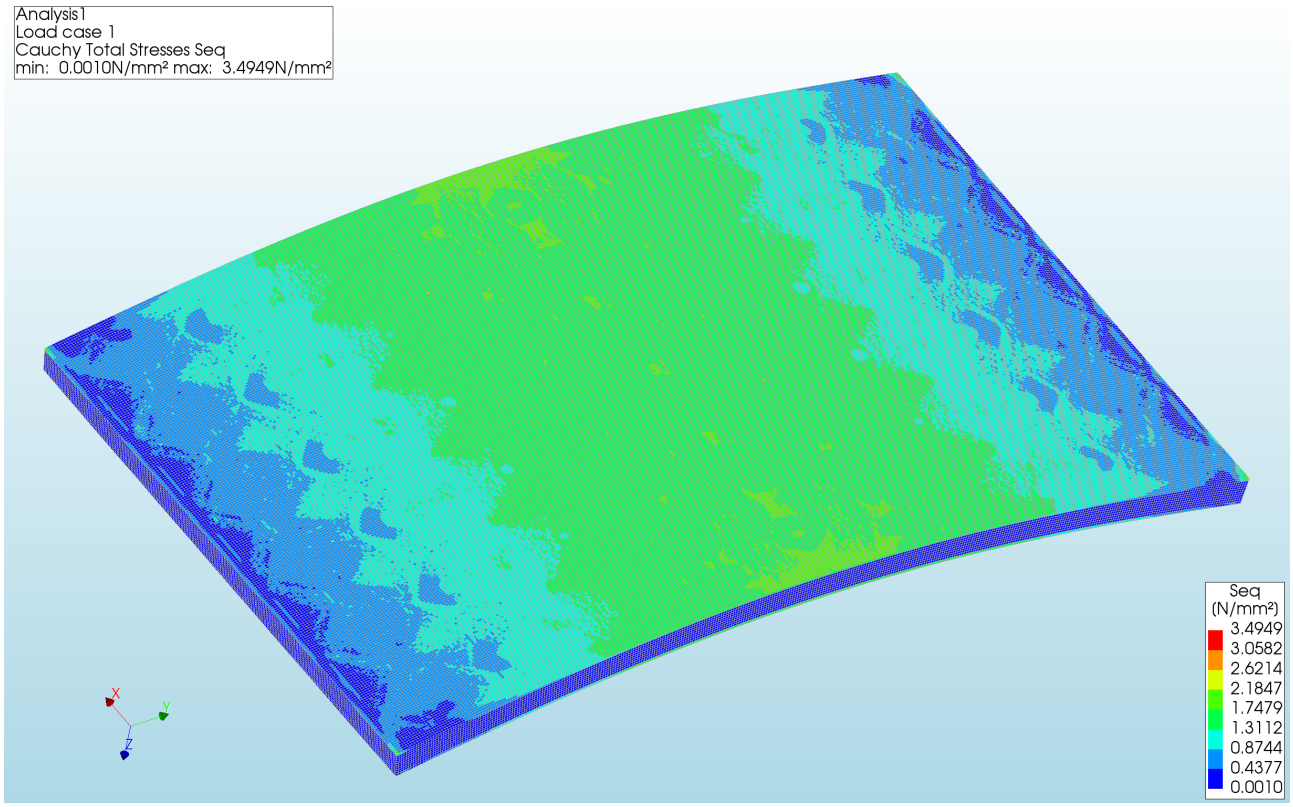


Figure 178: The Von Mises stresses of the bottom side of Pattern 2 with an element size of 2 millimetre

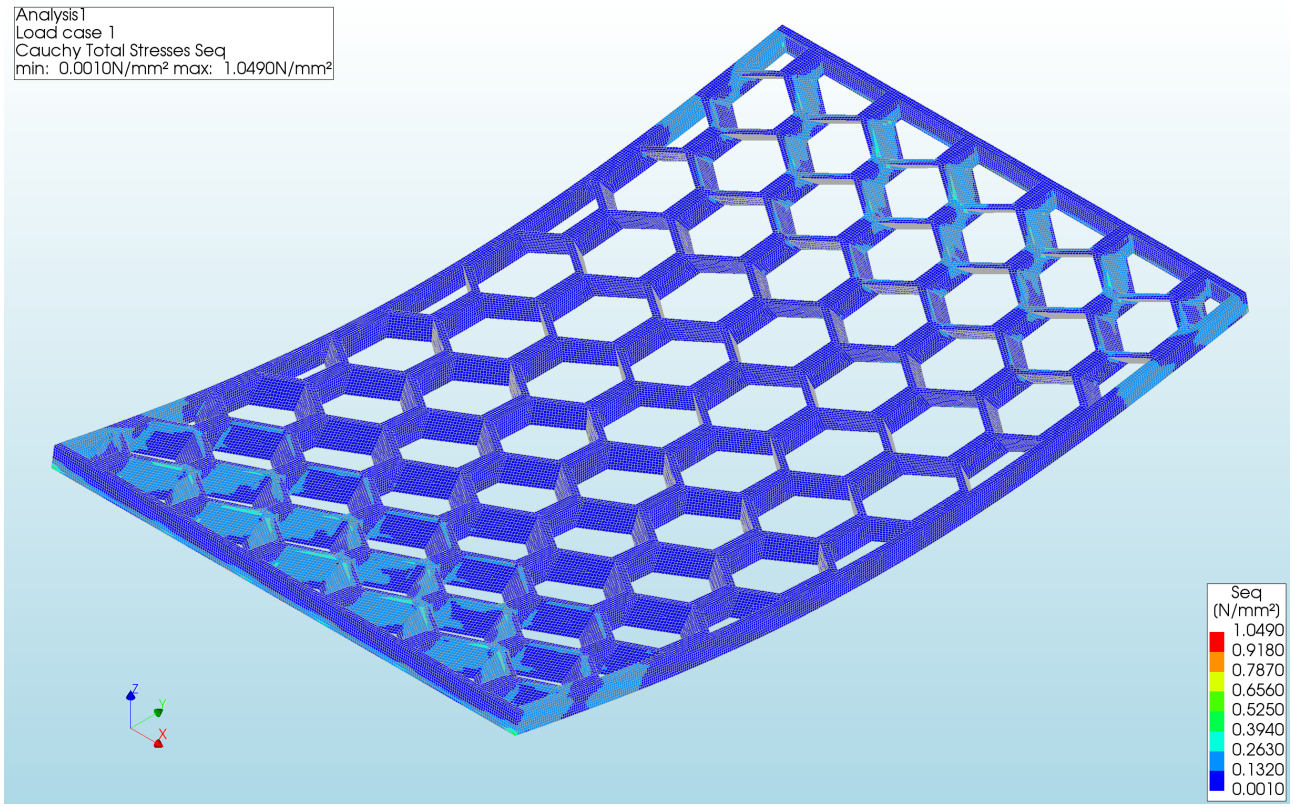


Figure 179: The Von Mises stresses of the Core of Pattern 2 with an element size of 2 millimetre

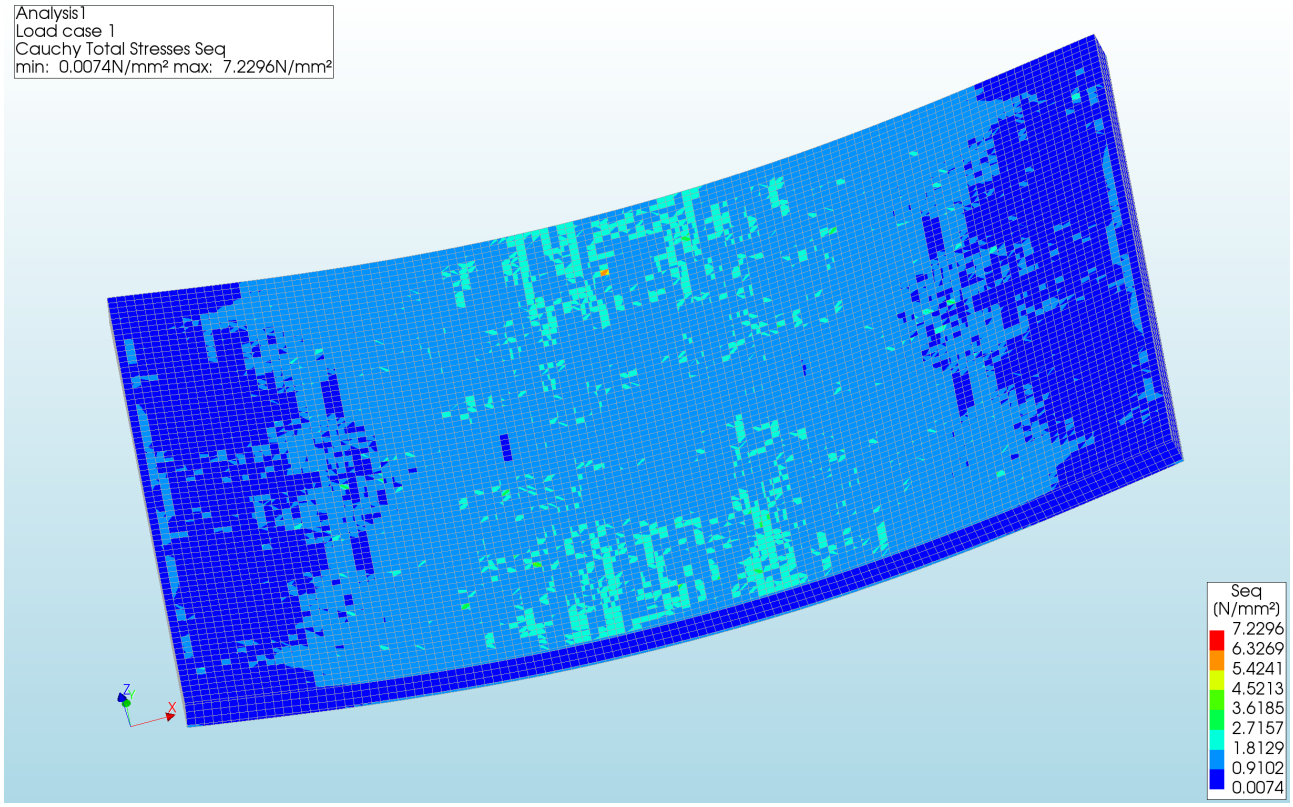


Figure 180: The Von Mises stresses of the top side of Pattern 3 with an element size of 6 millimetre

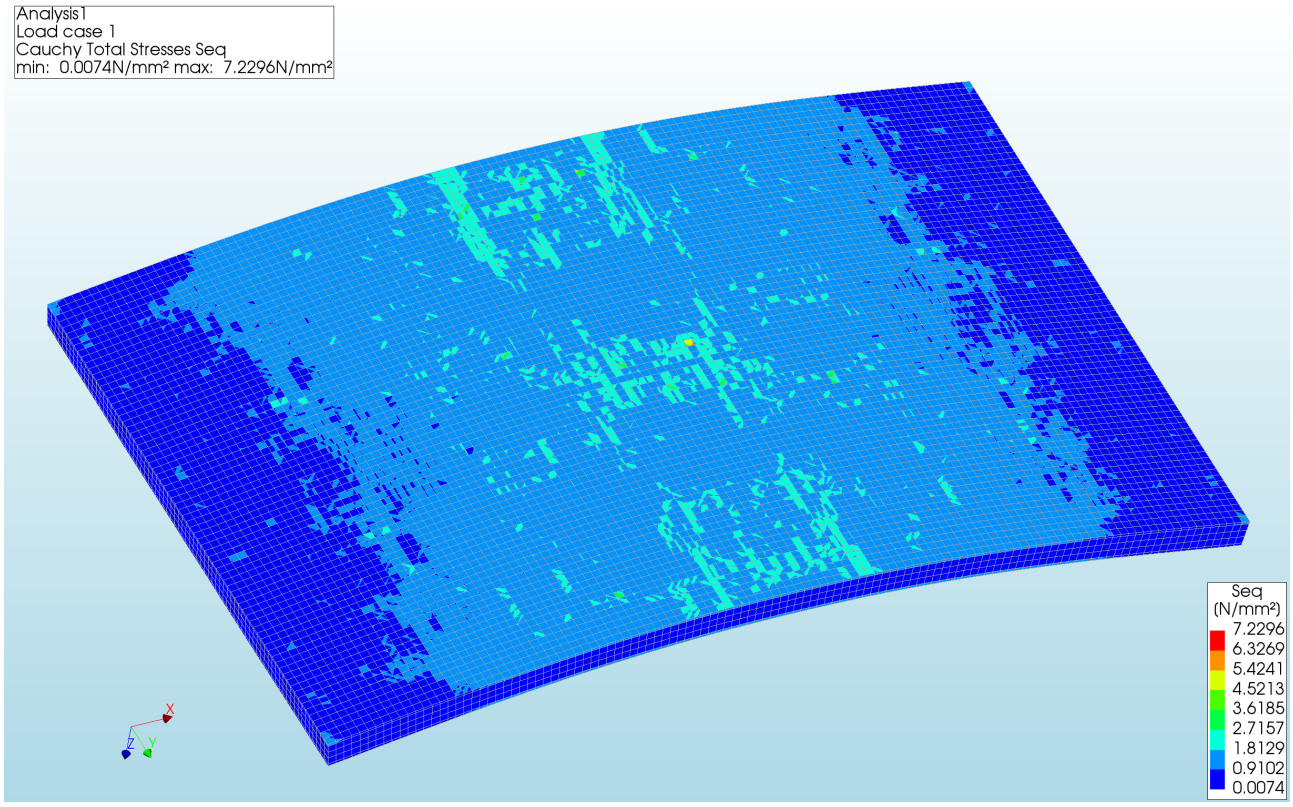


Figure 181: The Von Mises stresses of the bottom side of Pattern 3 with an element size of 6 millimetre

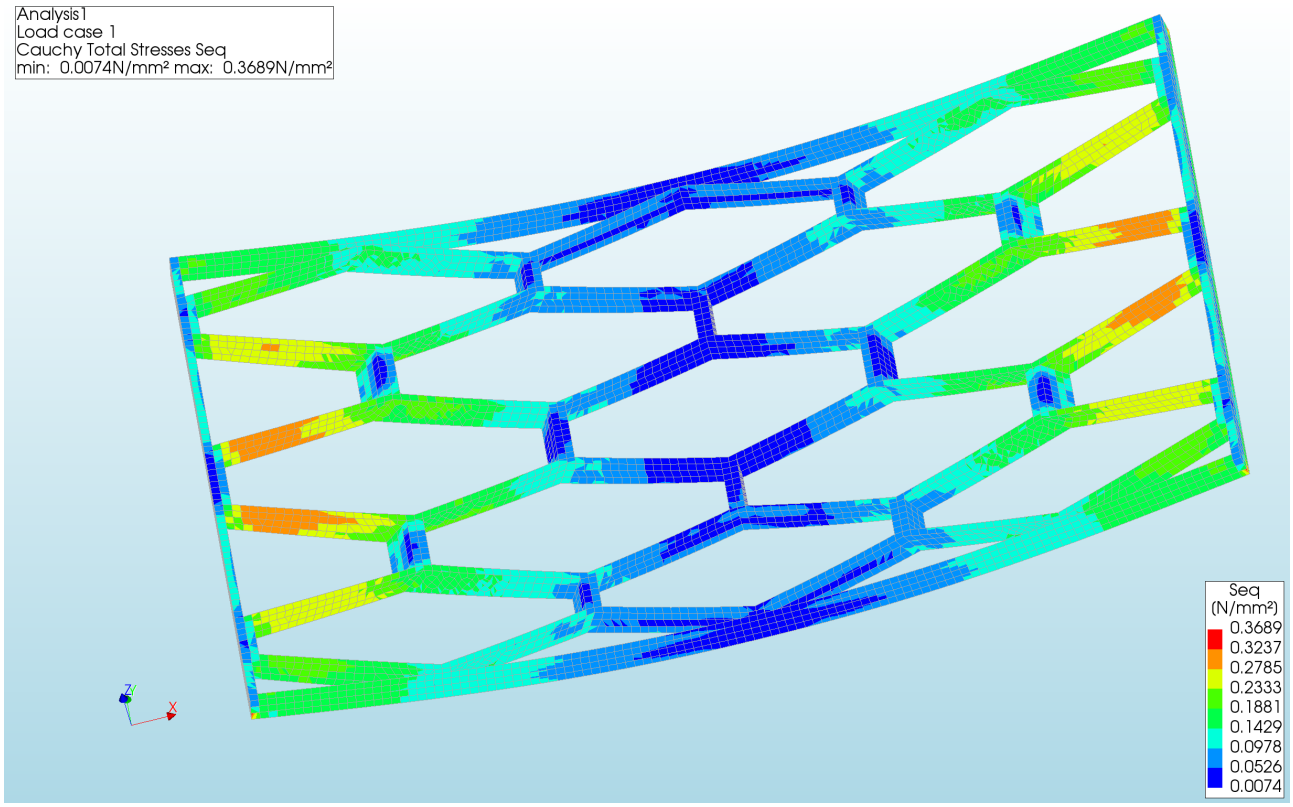


Figure 182: The Von Mises stresses of the Core of Pattern 3 with an element size of 6 millimetre

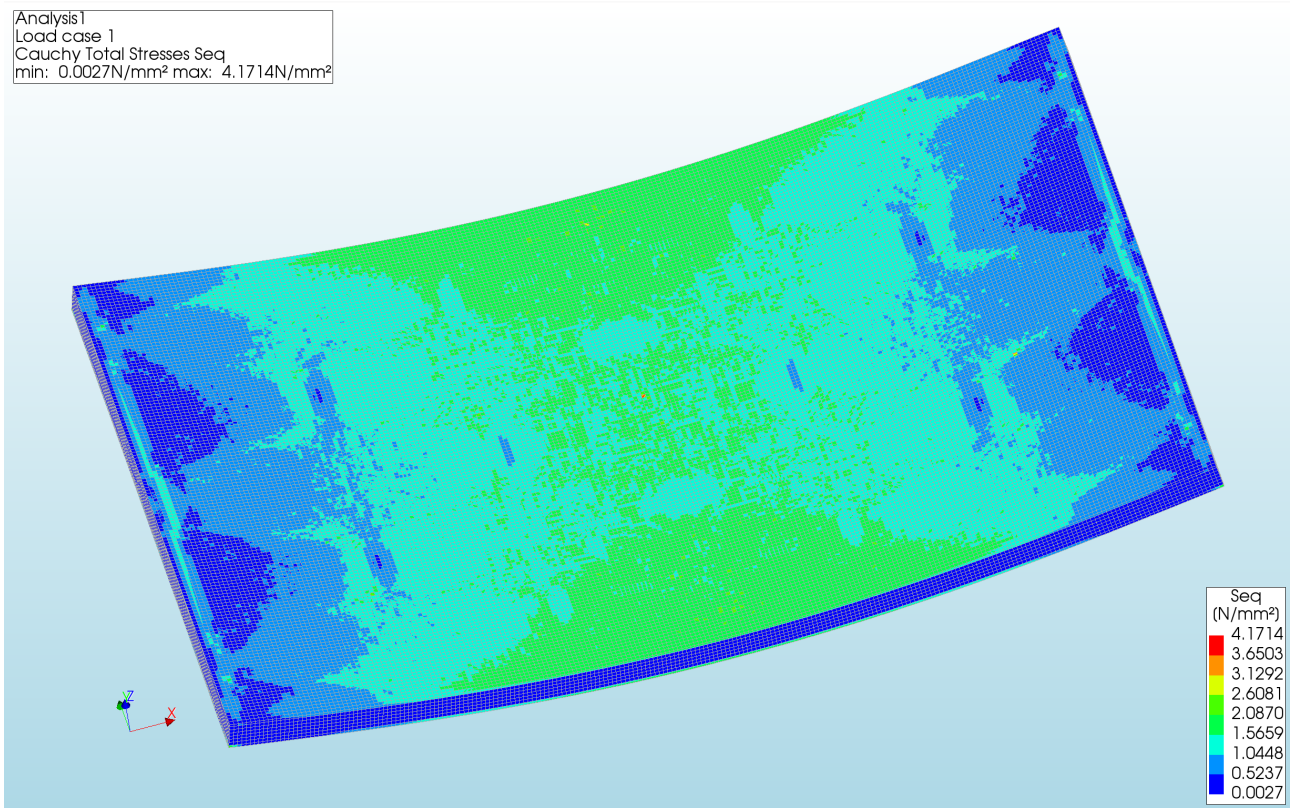


Figure 183: The Von Mises stresses of the top side of Pattern 3 with an element size of 3 millimetre

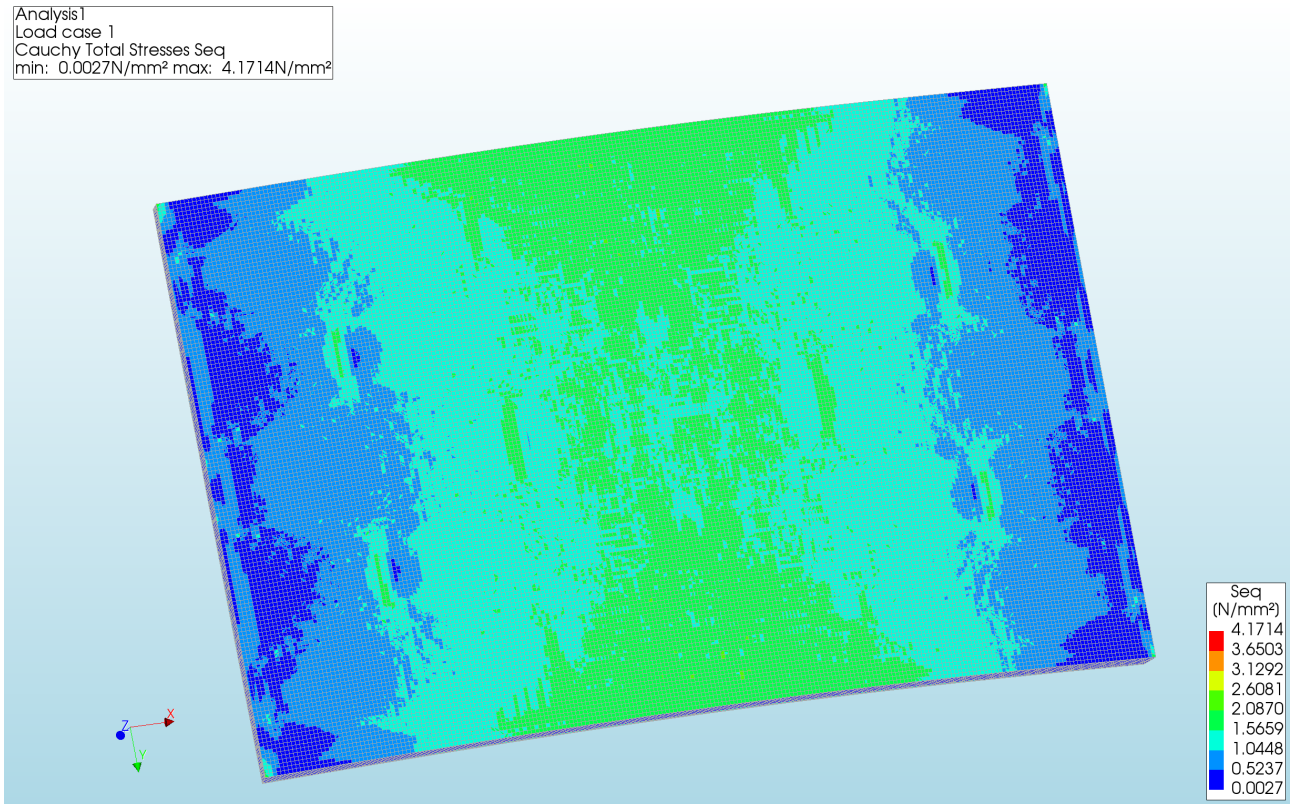


Figure 184: The Von Mises stresses of the bottom side of Pattern 3 with an element size of 3 millimetre

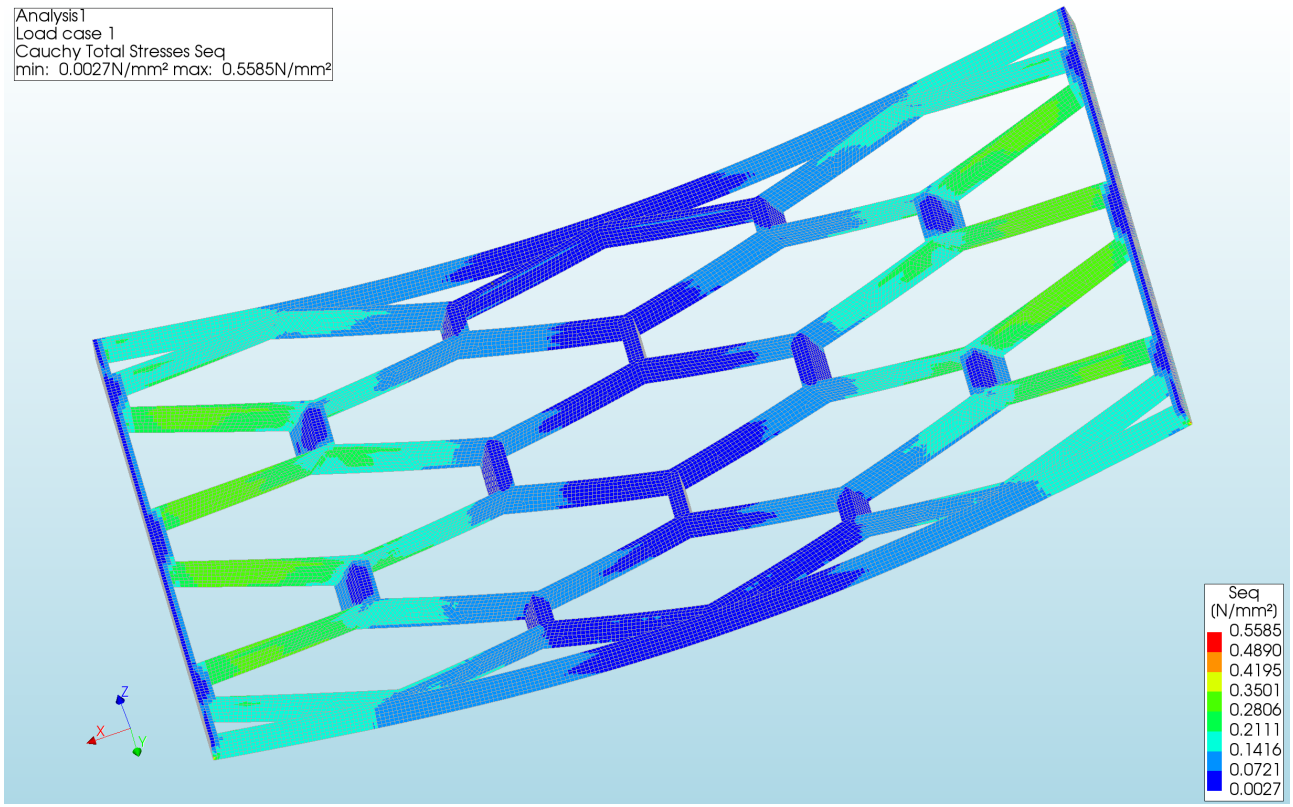


Figure 185: The Von Mises stresses of the Core of Pattern 3 with an element size of 3 millimetre

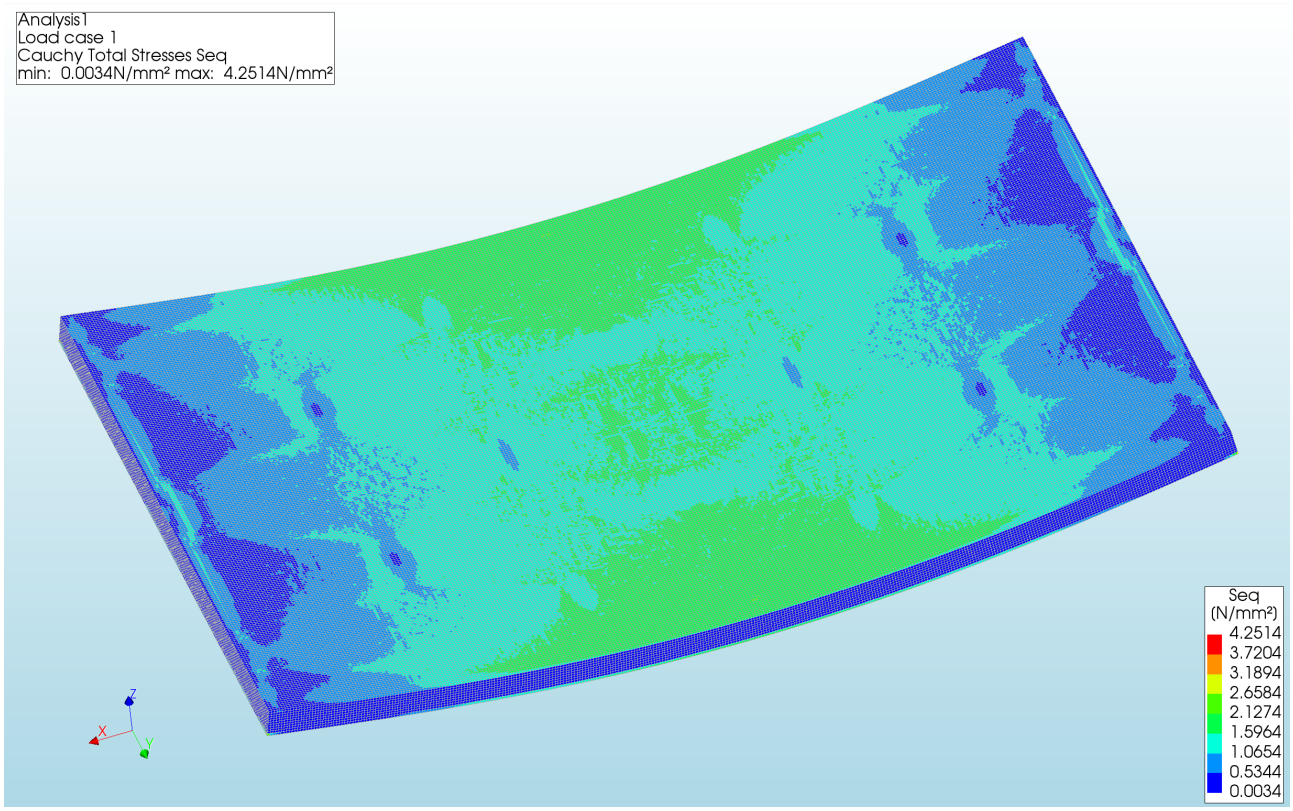


Figure 186: The Von Mises stresses of the top side of Pattern 3 with an element size of 2 millimetre

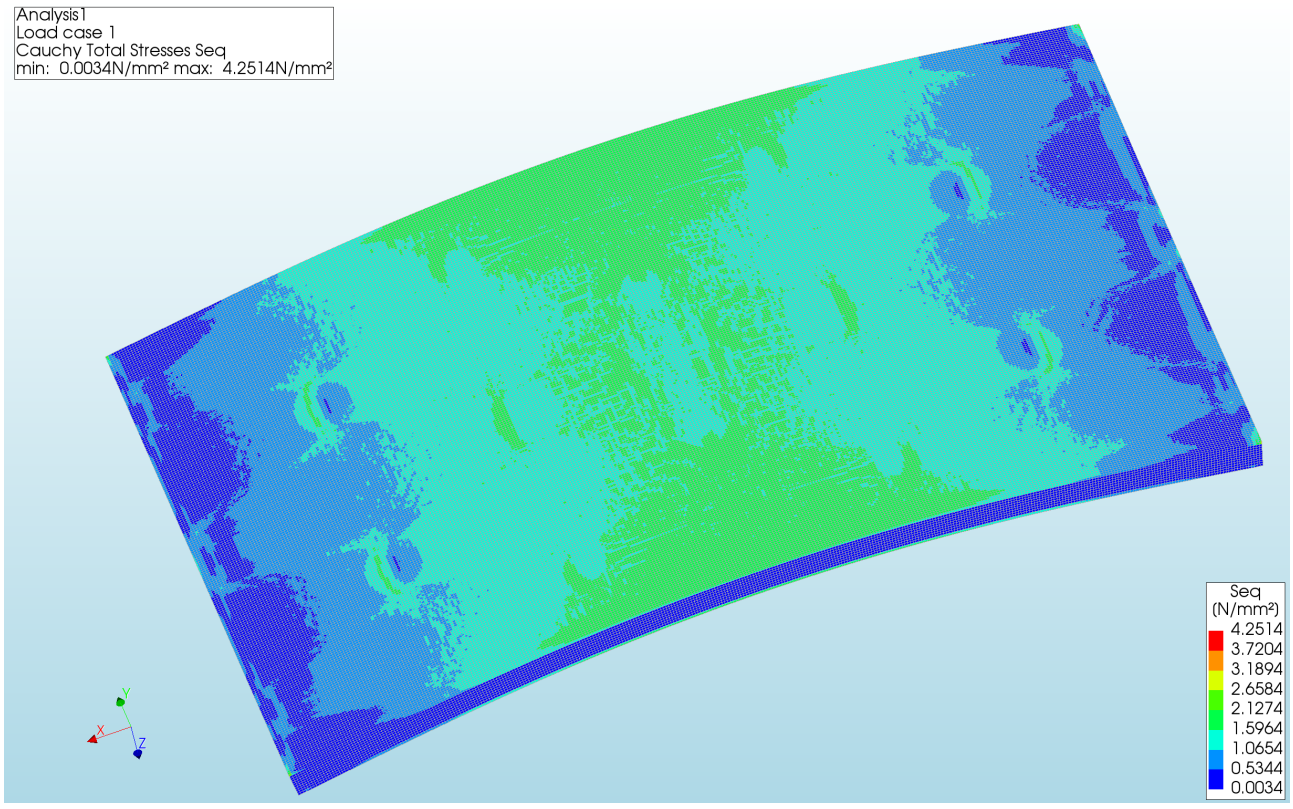


Figure 187: The Von Mises stresses of the bottom side of Pattern 3 with an element size of 2 millimetre

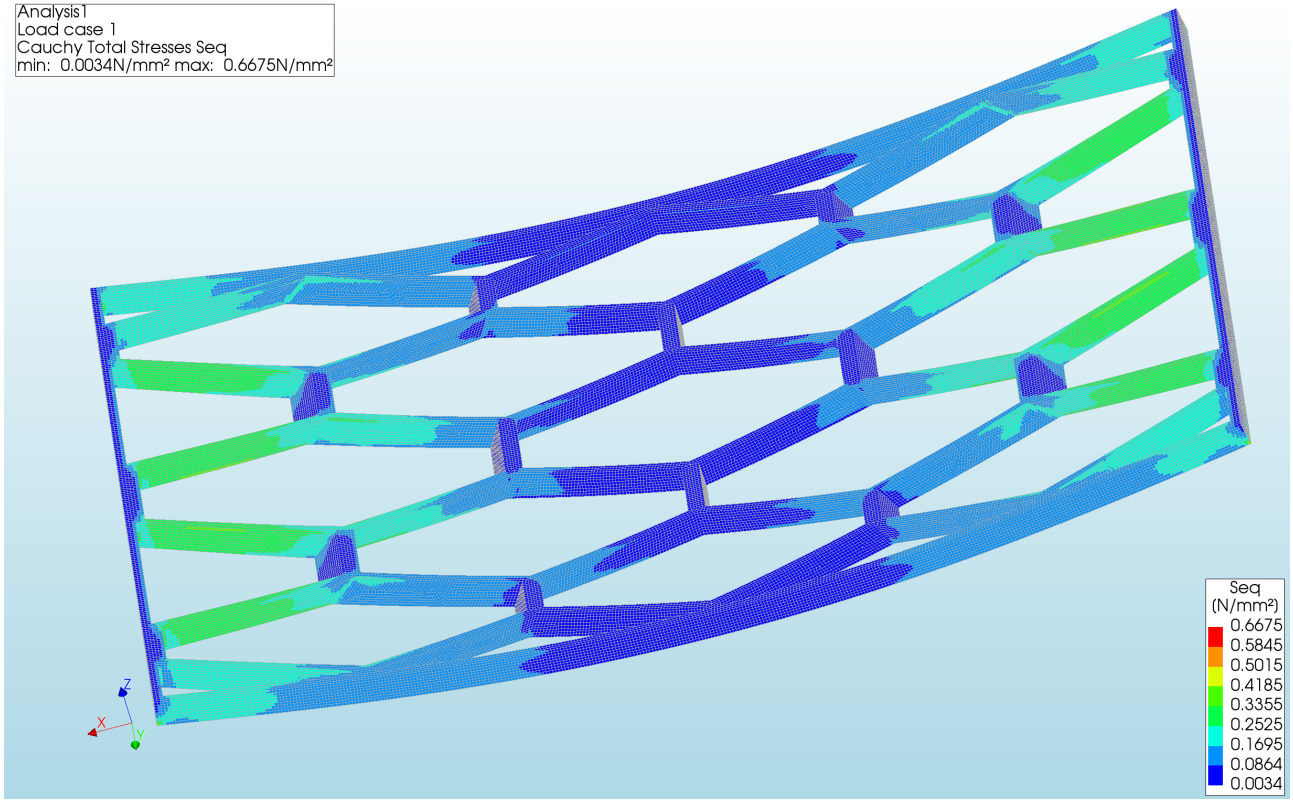


Figure 188: The Von Mises stresses of the Core of Pattern 3 with an element size of 2 millimetre

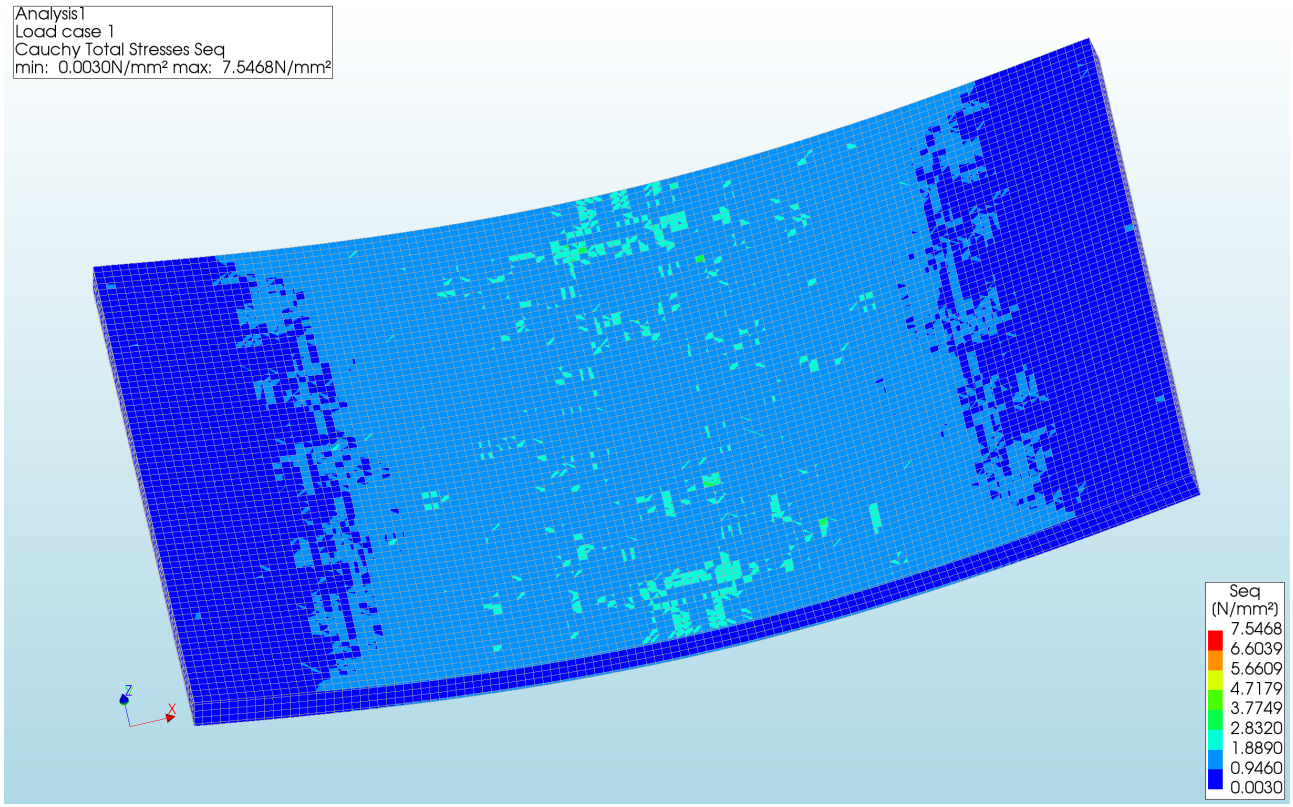


Figure 189: The Von Mises stresses of the top side of Pattern 4 with an element size of 6 millimetre

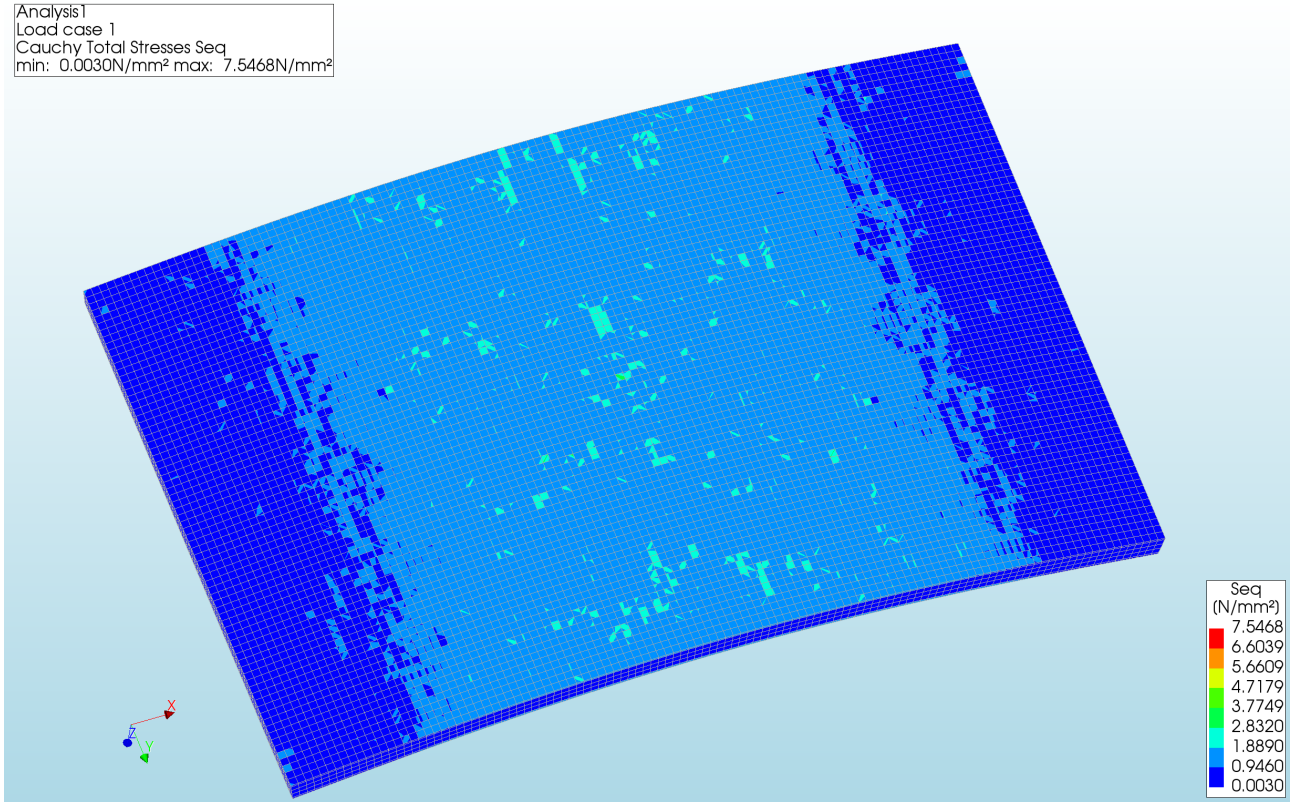


Figure 190: The Von Mises stresses of the bottom side of Pattern 4 with an element size of 6 millimetre

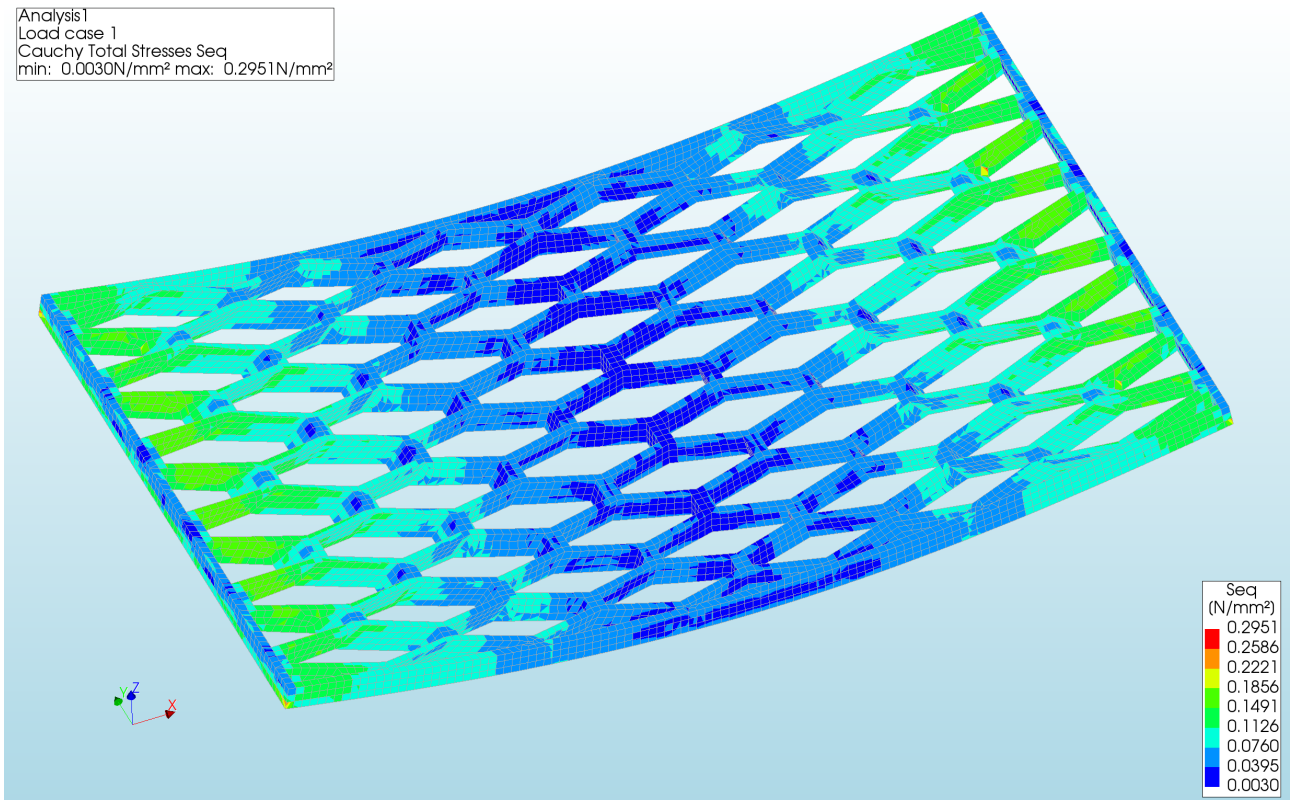


Figure 191: The Von Mises stresses of the Core of Pattern 4 with an element size of 6 millimetre

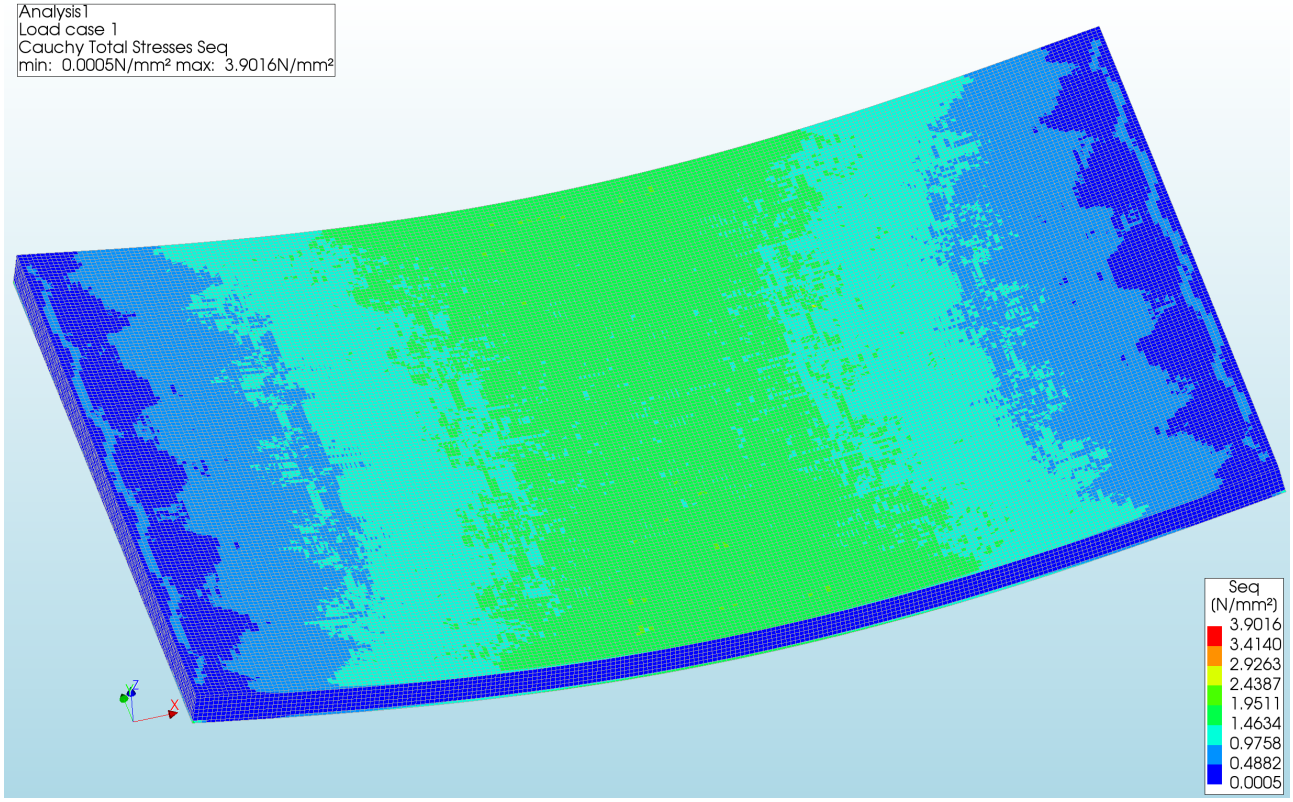


Figure 192: The Von Mises stresses of the top side of Pattern 4 with an element size of 3 millimetre

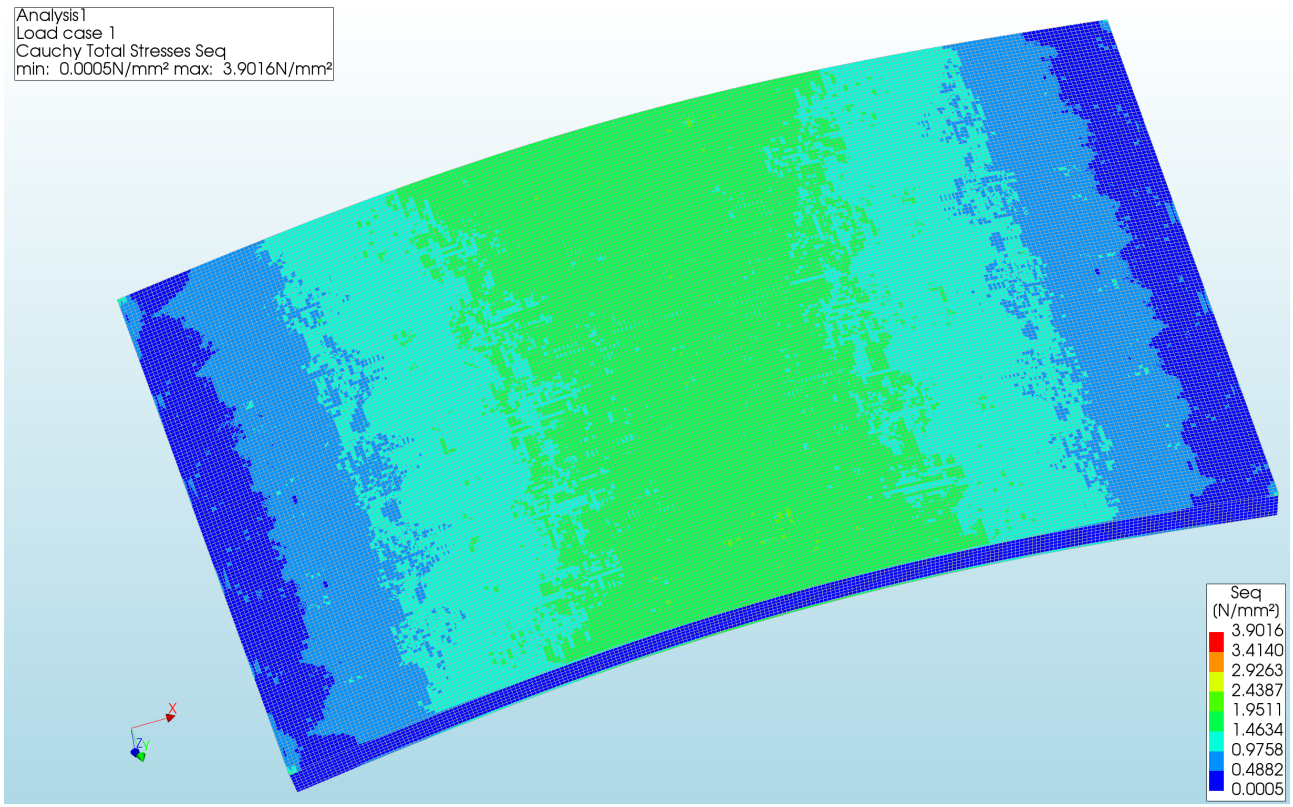


Figure 193: The Von Mises stresses of the bottom side of Pattern 4 with an element size of 3 millimetre

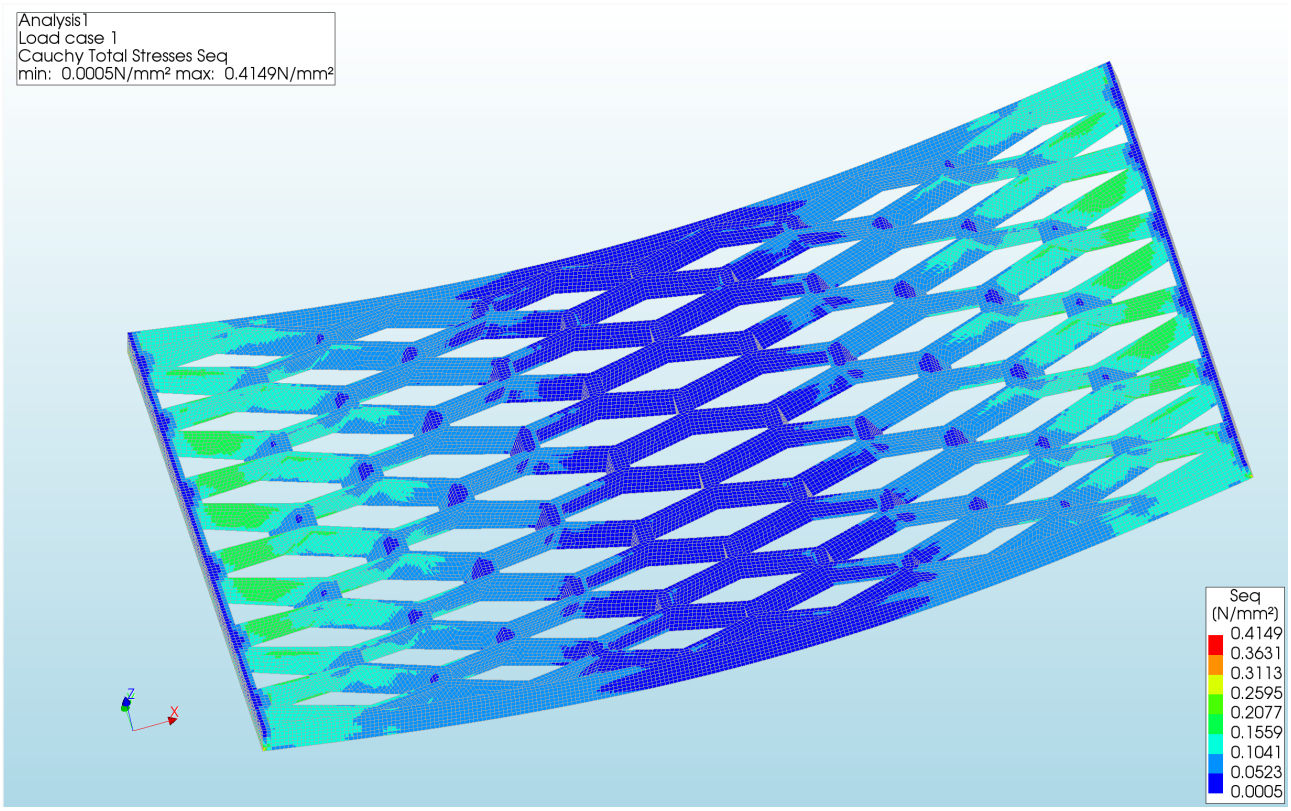


Figure 194: The Von Mises stresses of the Core of Pattern 4 with an element size of 3 millimetre

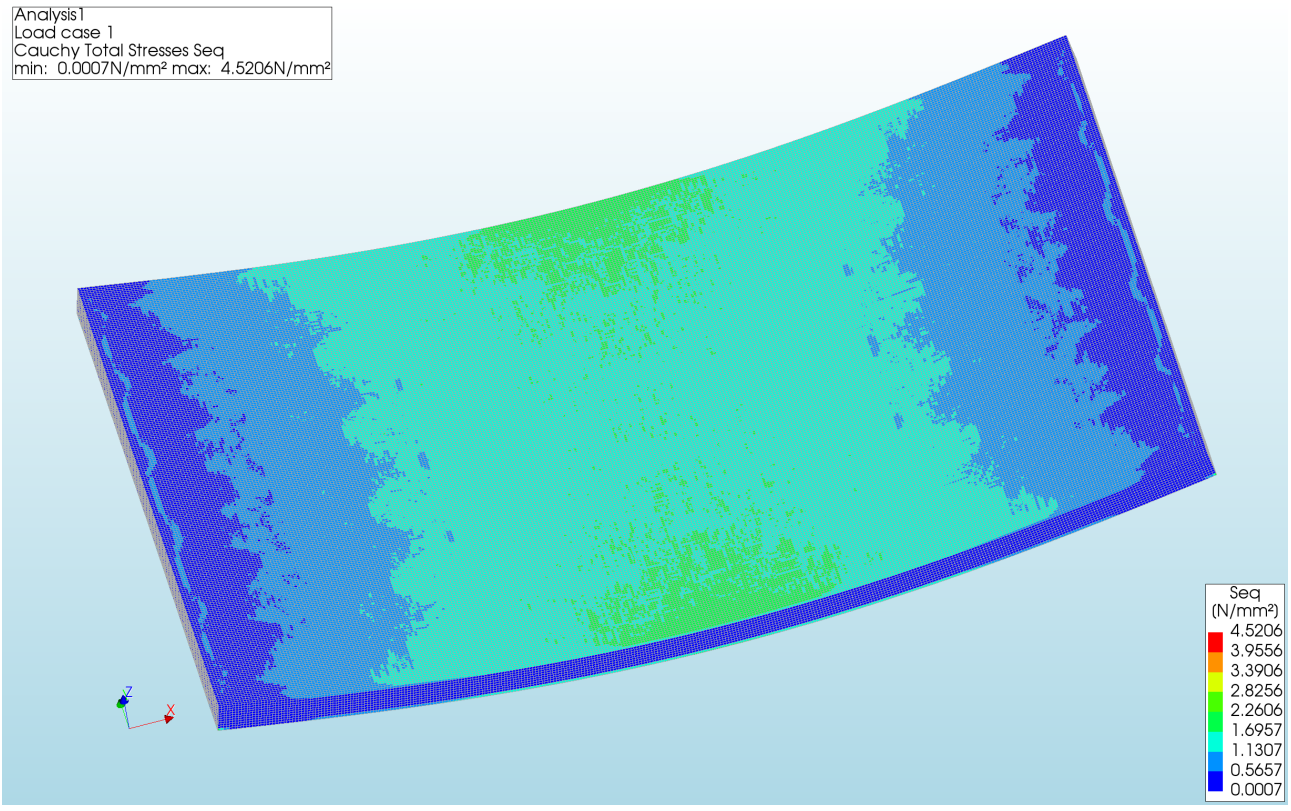


Figure 195: The Von Mises stresses of the top side of Pattern 4 with an element size of 2 millimetre

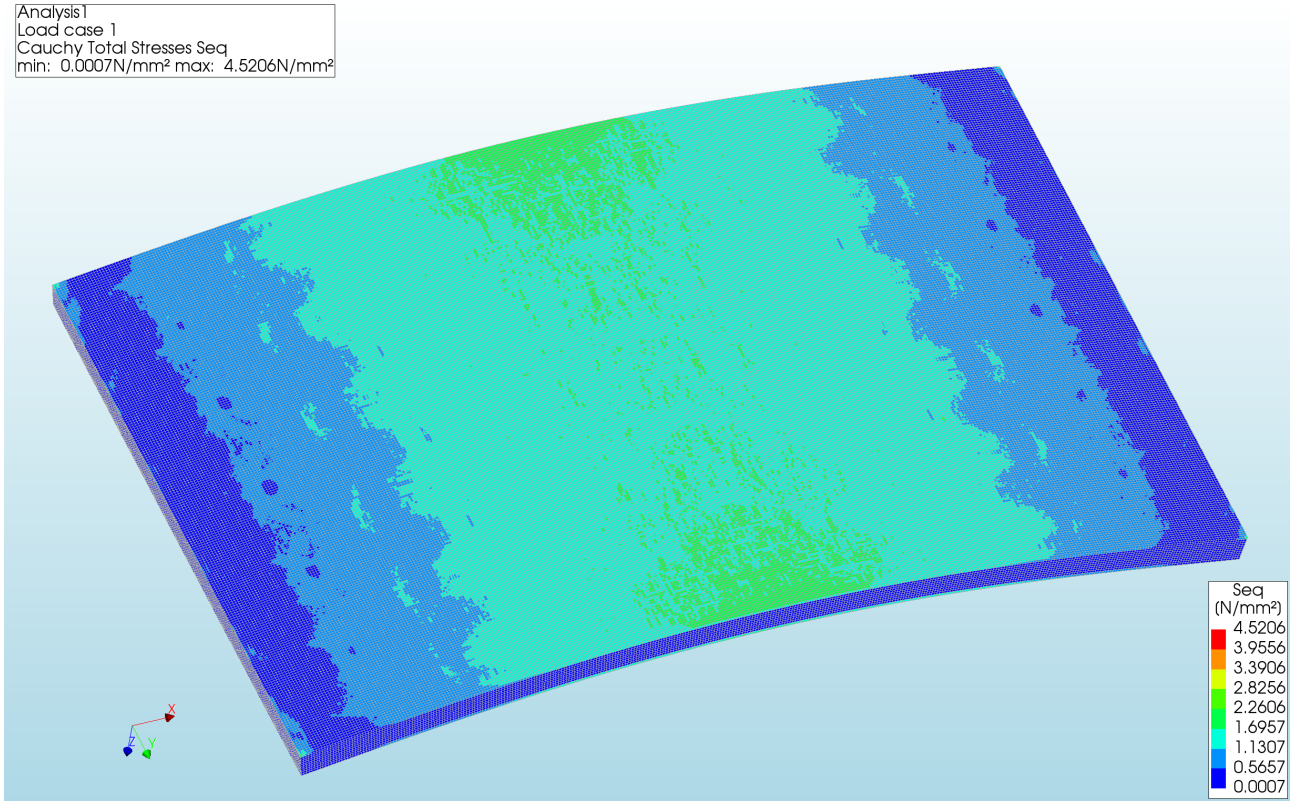


Figure 196: The Von Mises stresses of the bottom side of Pattern 4 with an element size of 2 millimetre

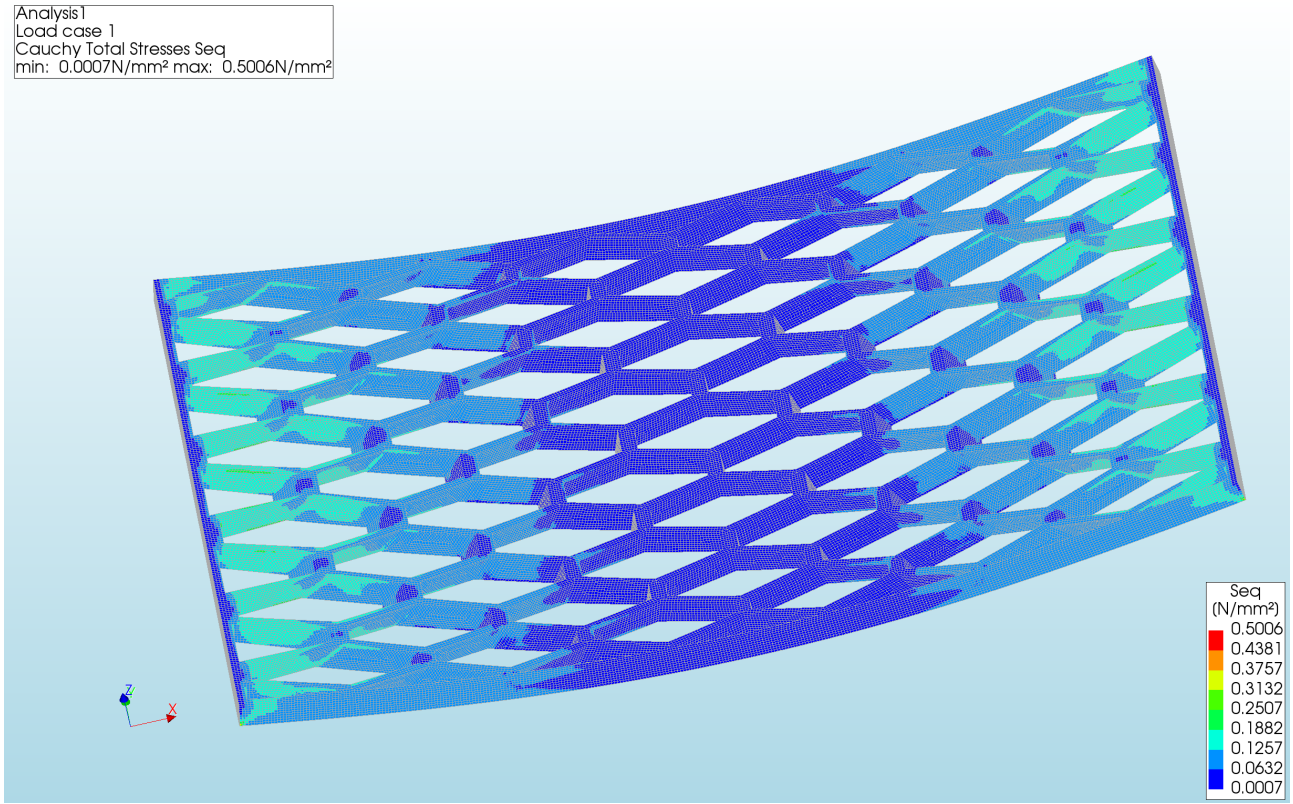


Figure 197: The Von Mises stresses of the Core of Pattern 4 with an element size of 2 millimetre

Appendix F: Data Validation Karamba3D

Karamba3D results

	Maximum deflection [mm]	Maximum Unity Check	Computation time [s]
A0	0.1963	0.0575	1.61
A1	0.2041	0.0720	2.22
A2	0.2062	0.0865	2.54
A3	0.2080	0.113	8.57
A4	0.2077	0.149	76.2
B0	0.1930	0.0587	2.21
B1	0.2031	0.0717	2.93
B2	0.2057	0.0825	4.09
B3	0.2086	0.103	10.1
B4	0.2082	0.158	75.8
C0	0.1900	0.0602	2.94
C1	0.2025	0.0741	3.72
C2	0.2057	0.0804	4.97
C3	0.2093	0.106	11.3
C4	0.2095	0.168	78.9

Table 48: Results of Pattern 1 in Karamba3D

	Maximum deflection [mm]	Maximum Unity Check	Computation time [s]
A0	0.1616	0.0528	12.7
A1	0.1672	0.0537	15.2
A2	0.1687	0.0552	17.3
A3	0.1697	0.0643	28.2
A4	0.1692	0.102	130
B0	0.1578	0.0531	18.8
B1	0.1660	0.0544	21.0
B2	0.1681	0.0563	23.4
B3	0.1700	0.0616	35.4
B4	0.1694	0.103	131
C0	0.1538	0.0546	26.0
C1	0.1648	0.0566	28.5
C2	0.1677	0.0566	31.5
C3	0.1700	0.0667	45.1
C4	0.1702	0.113	146

Table 49: Results of Pattern 2 in Karamba3D

	Maximum deflection [mm]	Maximum Unity Check	Computation time [s]
A0	0.1872	0.0495	1.76
A1	0.1948	0.0519	2.58
A2	0.1966	0.0544	3.56
A3	0.1972	0.0769	9.90
A4	0.1979	0.114	86.4
B0	0.1840	0.0493	2.40
B1	0.1931	0.0510	3.42
B2	0.1957	0.0530	4.57
B3	0.1971	0.0933	11.4
B4	0.1975	0.110	86.9
C0	0.1806	0.0484	3.11
C1	0.1915	0.0508	4.16
C2	0.1948	0.0549	5.47
C3	0.1966	0.0672	13.4
C4	0.1974	0.107	89.5

Table 50: Results of Pattern 3 in Karamba3D

	Maximum deflection [mm]	Maximum Unity Check	Computation time [s]
A0	0.1618	0.04745	13.7
A1	0.1657	0.0489	15.9
A2	0.1680	0.0506	28.7
A3	0.1685	0.0611	30.8
A4	0.1693	0.0825	152
B0	0.1578	0.0470	19.4
B1	0.1649	0.0492	22.3
B2	0.1671	0.0508	25.4
B3	0.1682	0.0622	39.2
B4	0.1689	0.0756	152
C0	0.1535	0.0459	26.6
C1	0.1631	0.0482	29.9
C2	0.1662	0.0500	33.9
C3	0.1683	0.0625	49.2
C4	0.1687	0.0715	169

Table 51: Results of Pattern 4 in Karamba3D

Diana shell elements results

Element size [mm]	Maximum Deflection [mm]	Von-Mises Stress [N/mm2]	Unity Check
6	0.2232	3.115	0.0779
3	0.2239	3.540	0.0885
2	0.2241	3.725	0.0931

Table 52: Diana Shell results Pattern 1

Element size [mm]	Maximum Deflection [mm]	Von-Mises Stress [N/mm2]	Unity Check
6	0.1816	2.304	0.0576
3	0.1819	2.335	0.0584
2	0.1821	2.348	0.0587

Table 53: Diana Shell results Pattern 2

Element size [mm]	Maximum Deflection [mm]	Von-Mises Stress [N/mm2]	Unity Check
6	0.2052	2.570	0.06425
3	0.2059	3.244	0.0811
2	0.2061	3.591	0.0898

Table 54: Diana Shell results Pattern 3

Element size [mm]	Maximum Deflection [mm]	Von-Mises Stress [N/mm2]	Unity Check
6	0.17365	2.189	0.0547
3	0.17380	2.319	0.0580
2	0.17381	2.409	0.0602

Table 55: Diana Shell results Pattern 4

Diana solid elements results

Element size [mm]	Maximum Deflection [mm]	Von-Mises Stress [N/mm2]	Unity check
6	0.1647	10.441	0.261
3	0.1742	5.7738	0.144
2	0.1774	5.872	0.147

Table 56: Diana Solid results Pattern 1

Element size [mm]	Maximum Deflection [mm]	Von-Mises Stress [N/mm2]	Unity Check
6	0.1384	10.980	0.275
3	0.1451	3.671	0.0918
2	0.1461	3.495	0.0874

Table 57: Diana Solid results Pattern 2

Element size [mm]	Maximum Deflection [mm]	Von-Mises Stress [N/mm2]	Unity Check
6	0.1491	7.230	0.181
3	0.1567	4.171	0.104
2	0.1596	4.251	0.0854

Table 58: Diana Solid results Pattern 3

Element size [mm]	Maximum Deflection [mm]	Von-Mises Stress [N/mm2]	Unity Check
6	0.1336	7.547	0.189
3	0.1385	3.902	0.0975
2	0.1393	4.52	0.113

Table 59: Diana Solid results Pattern 4

Appendix G: Data daylight grid size determination

Daylight without reflection

	Model 1	Computation time [s]
Grid 1	0.0079232 0.0037686	3.4
Grid 2	0.0074504 0.0040759	4.1
Grid 3	0.0074504 0.0033069	5.6
Grid 4	0.0078099 0.0036500	8.6
Grid 5	0.0067429 0.0032142	13.6
Grid 6	0.0074486 0.0035766	26.4
Grid 7	0.0075561 0.0036013	42.4
Grid 8	0.0069325 0.0032559	75.6
Grid 9	0.0068051 0.0032624	168
Grid 10	0.00672235 0.0030917	222

Table 60: Overview results Model 1 with different grid sizes for both summer and winter solstice.

	Model 2	Computation time [s]
Grid 1	0.012674 0.0064834	3.6
Grid 2	0.012861 0.0068812	4.4
Grid 3	0.012499 0.0065009	5.7
Grid 4	0.013023 0.0066773	8.3
Grid 5	0.011577 0.0057157	13.9
Grid 6	0.012457 0.0065381	27
Grid 7	0.012760 0.0064376	41.8
Grid 8	0.0120291 0.0061720	74.2
Grid 9	0.011628 0.0060807	168
Grid 10	0.11691 0.0057981	222

Table 61: Overview results Model 2 with different grid sizes for both summer and winter solstice.

	Model 3	Computation time [s]
Grid 1	0.020054 0.010217	3.6
Grid 2	0.020654 0.011091	4.4
Grid 3	0.019706 0.0097048	5.7
Grid 4	0.021201 0.010732	8.3
Grid 5	0.018389 0.0094388	13.9
Grid 6	0.019841 0.010413	27
Grid 7	0.020276 0.010511	41.8
Grid 8	0.019484 0.010180	74.2
Grid 9	0.018974 0.010092	168
Grid 10	0.018805 0.0095635	222

Table 62: Overview results Model 3 with different grid sizes for both summer and winter solstice.

	Model 4	Computation time [s]
Grid 1	0.047252 0.022444	4.0
Grid 2	0.043587 0.021229	4.7
Grid 3	0.040846 0.014813	5.9
Grid 4	0.048074 0.019685	8.8
Grid 5	0.036300 0.014340	14.2
Grid 6	0.041553 0.019041	27.4
Grid 7	0.043464 0.016730	42.6
Grid 8	0.038119 0.014438	76.9
Grid 9	0.037322 0.014750	174
Grid 10	0.036437 0.013982	228

Table 63: Overview results Model 4 with different grid sizes for both summer and winter solstice.

Daylight with Reflection

Summer	Grid A	Grid B	Grid C	Grid D	Grid E	Grid F	Grid G
Test 1	0.081766	0.081547	0.081507	0.081461	0.081421	0.081382	0.0813627
Test 2	0.081771	0.081583	0.081498	0.081458	0.081396	0.081386	0.081365
Test 3	0.081817	0.081563	0.081564	0.081472	0.081392	0.081366	0.0813729
Test 4	0.081588	0.081662	0.081500	0.081467	0.081398	0.081361	0.0813651
Test 5	0.081829	0.081576	0.081532	0.081461	0.081444	0.081389	0.0813664
Test 6	0.081941	0.081624	0.081476	0.081409	0.081399	0.081381	0.0813675
Test 7	0.081852	0.081559	0.081576	0.081460	0.081387	0.081369	0.0813759
Test 8	0.081839	0.081637	0.081480	0.081500	0.081419	0.081382	0.0813752
Test 9	0.081807	0.081646	0.081515	0.081454	0.081412	0.081381	0.0813755
Test 10	0.081801	0.081596	0.081556	0.081420	0.081418	0.081388	0.0813642
Test 11	0.081784	0.081644	0.081639	0.081486	0.081395	0.081382	0.081358
Test 12	0.081854	0.081602	0.081487	0.081473	0.081380	0.081396	0.081380
Test 13	0.081789	0.081679	0.081561	0.081446	0.081389	0.081394	0.081366
Test 14	0.081943	0.081625	0.081529	0.081414	0.081421	0.081397	0.081366
Test 15	0.081933	0.081603	0.081518	0.081448	0.081423	0.081373	0.081370
Test 16	0.081730	0.081592	0.081530	0.081402	0.081391	0.081383	0.081369
Test 17	0.081808	0.081674	0.081550	0.081459	0.081409	0.081368	0.081368
Test 18	0.081870	0.081548	0.081578	0.081487	0.081378	0.081389	0.081377
Test 19	0.081869	0.081540	0.081458	0.081455	0.081421	0.081391	0.081361
Test 20	0.082010	0.081583	0.081541	0.081463	0.081401	0.081387	0.081368
Test 21	0.081708	0.081630	0.081482	0.081439	0.081405	0.081380	0.081368
Test 22	0.082019	0.081601	0.081534	0.081463	0.081397	0.081378	0.081367
Test 23	0.081902	0.081666	0.081532	0.081448	0.081379	0.081375	0.081363
Test 24	0.081891	0.081636	0.081458	0.081441	0.081400	0.081390	0.081364
Test 25	0.081679	0.081626	0.081560	0.081454	0.081360	0.081367	0.081369
Test 26	0.081783	0.081555	0.081472	0.081447	0.081414	0.081373	0.081373
Test 27	0.081652	0.081586	0.081487	0.081456	0.081402	0.081378	0.081368
Test 28	0.081690	0.081594	0.081529	0.081467	0.081383	0.081391	0.081362
Test 29	0.081942	0.081586	0.081514	0.081450	0.081404	0.081373	0.081371
Test 30	0.081791	0.081591	0.081508	0.081456	0.081388	0.081394	0.081367
Max. value	0.082019	0.081679	0.081639	0.081500	0.081444	0.081397	0.081380
Min. Value	0.081588	0.081540	0.081458	0.081402	0.081360	0.081361	0.081358
Max. change	0.528%	0.170%	0.222%	0.120%	0.103%	0.044%	0.027%
Average	0.081822	0.081605	0.081522	0.081454	0.081401	0.081381	0.081368

Table 64: Overview of all conducted tests of Model 1 with reflection at summer solstice

Winter	Grid A	Grid B	Grid C	Grid D	Grid E	Grid F	Grid G
Test 1	0.043086	0.043290	0.043248	0.043147	0.043105	0.043124	0.0431075
Test 2	0.043339	0.043065	0.043169	0.043127	0.043141	0.043097	0.0431168
Test 3	0.043247	0.043097	0.043133	0.043149	0.043130	0.043117	0.0431163
Test 4	0.043208	0.043328	0.043081	0.043143	0.043124	0.043128	0.04310
Test 5	0.043360	0.043080	0.043237	0.043133	0.043110	0.043146	0.04310
Test 6	0.043410	0.043188	0.043179	0.043175	0.043104	0.043130	0.043122
Test 7	0.043095	0.043064	0.043160	0.043234	0.043129	0.043100	0.043112
Test 8	0.043359	0.043170	0.043124	0.043214	0.043162	0.043080	0.0430949
Test 9	0.043185	0.043281	0.043040	0.043152	0.043095	0.043103	0.0431065
Test 10	0.043148	0.043164	0.043137	0.043174	0.043123	0.043100	0.0431119
Test 11	0.043067	0.043160	0.043273	0.043049	0.043133	0.043098	0.043096
Test 12	0.043262	0.043295	0.043202	0.043131	0.043081	0.043092	0.043109
Test 13	0.043311	0.043043	0.043166	0.043082	0.043097	0.043117	0.043097
Test 14	0.043366	0.043347	0.043184	0.043106	0.043157	0.043147	0.043100
Test 15	0.043066	0.043302	0.043124	0.043028	0.043107	0.043085	0.043109
Test 16	0.043344	0.043119	0.043219	0.043115	0.043090	0.043106	0.043117
Test 17	0.043113	0.043213	0.043071	0.043061	0.043139	0.043081	0.043103
Test 18	0.043432	0.043255	0.043108	0.043144	0.043086	0.043107	0.043098
Test 19	0.043371	0.043076	0.043081	0.043100	0.043092	0.043133	0.043102
Test 20	0.043341	0.043111	0.043163	0.043096	0.043120	0.043102	0.043116
Test 21	0.043261	0.043260	0.043134	0.043115	0.043105	0.043097	0.043115
Test 22	0.043425	0.043272	0.043004	0.043052	0.043113	0.043156	0.043090
Test 23	0.043167	0.043287	0.043154	0.043176	0.043114	0.043092	0.043116
Test 24	0.043281	0.043351	0.043195	0.043098	0.043108	0.043135	0.043084
Test 25	0.043411	0.043382	0.043261	0.043185	0.043111	0.043103	0.043099
Test 26	0.043190	0.043346	0.043201	0.043165	0.043134	0.043116	0.043094
Test 27	0.043280	0.043166	0.043144	0.043116	0.043113	0.043096	0.043113
Test 28	0.043260	0.043208	0.043131	0.043092	0.043123	0.043100	0.043104
Test 29	0.043250	0.043276	0.043106	0.043210	0.043115	0.043098	0.043108
Test 30	0.043217	0.043338	0.043172	0.043139	0.043104	0.043089	0.043122
Max. value	0.043432	0.043382	0.043273	0.043234	0.043162	0.043156	0.043122
Min. Value	0.043066	0.043043	0.043004	0.043028	0.043081	0.043080	0.043084
Max. change	0.850%	0.788%	0.626%	0.479%	0.188%	0.177%	0.088%
Average	0.043262	0.043218	0.043153	0.043130	0.043116	0.043109	0.043106

Table 65: Overview of all conducted tests of Model 1 with reflection at winter solstice

Summer	Grid A	Grid B	Grid C	Grid D	Grid E	Grid F	Grid G
Test 1	0.062285	0.062144	0.061890	0.061908	0.061865	0.061838	0.061842
Test 2	0.062384	0.062052	0.061865	0.061907	0.061867	0.061875	0.061838
Test 3	0.061944	0.062086	0.061957	0.061895	0.061864	0.061871	0.061850
Test 4	0.062412	0.062054	0.061938	0.061948	0.061850	0.061854	0.061842
Test 5	0.062276	0.061936	0.061969	0.061923	0.061862	0.061865	0.061847
Test 6	0.062407	0.062033	0.061904	0.061894	0.061864	0.061851	0.061848
Test 7	0.062203	0.061968	0.061872	0.061896	0.061902	0.061849	0.061847
Test 8	0.062287	0.062114	0.061905	0.061928	0.061866	0.061872	0.061846
Test 9	0.062395	0.062022	0.061951	0.061899	0.061881	0.061860	0.061844
Test 10	0.062263	0.062085	0.061934	0.061895	0.061872	0.061846	0.061847
Test 11	0.062144	0.061994	0.062038	0.061872	0.061839	0.061856	0.061840
Test 12	0.062230	0.061994	0.062006	0.061986	0.061868	0.061856	0.061842
Test 13	0.062252	0.062076	0.062055	0.061956	0.061887	0.061853	0.061856
Test 14	0.062205	0.061982	0.061857	0.061933	0.061923	0.061849	0.061847
Test 15	0.062175	0.062066	0.061985	0.061867	0.061849	0.061850	0.061846
Test 16	0.062300	0.062003	0.061962	0.061910	0.061888	0.061850	0.061838
Test 17	0.062188	0.062068	0.061931	0.061936	0.061893	0.061860	0.061847
Test 18	0.062312	0.062092	0.061993	0.061922	0.061870	0.061845	0.061843
Test 19	0.062140	0.062007	0.061955	0.061932	0.061924	0.061838	0.061846
Test 20	0.062131	0.062036	0.061990	0.061927	0.061887	0.061844	0.061841
Test 21	0.062162	0.062063	0.061960	0.061922	0.061855	0.061832	0.061852
Test 22	0.062227	0.062138	0.061965	0.061886	0.061870	0.061854	0.061840
Test 23	0.062145	0.062092	0.061937	0.061879	0.061855	0.061848	0.061840
Test 24	0.062177	0.062098	0.061998	0.061908	0.061876	0.061859	0.061843
Test 25	0.062183	0.062147	0.062025	0.061885	0.061845	0.061852	0.061851
Test 26	0.062289	0.062072	0.061939	0.061882	0.061862	0.061854	0.061845
Test 27	0.062193	0.061981	0.062025	0.061960	0.061863	0.061853	0.061844
Test 28	0.062307	0.062142	0.061945	0.061872	0.061857	0.061867	0.061831
Test 29	0.062171	0.061951	0.061955	0.061944	0.061881	0.061853	0.061851
Test 30	0.062048	0.062135	0.061934	0.061907	0.061895	0.061840	0.061842
Max. value	0.062412	0.062147	0.062055	0.061986	0.061924	0.061875	0.061856
Min. Value	0.061944	0.061936	0.061857	0.061867	0.061839	0.061832	0.061831
Max. change	0.756%	0.341%	0.320%	0.192%	0.137%	0.070%	0.040%
Average	0.062228	0.062054	0.061955	0.061913	0.061873	0.061853	0.061845

Table 66: Overview of all conducted tests of Model 2 with reflection at summer solstice

Winter	Grid A	Grid B	Grid C	Grid D	Grid E	Grid F	Grid G
Test 1	0.032427	0.032501	0.032578	0.032344	0.032475	0.032444	0.032459
Test 2	0.032739	0.032492	0.032480	0.032443	0.032493	0.032455	0.032457
Test 3	0.032337	0.032609	0.032504	0.032427	0.032481	0.032467	0.032465
Test 4	0.032544	0.032595	0.032438	0.032455	0.032519	0.032463	0.032452
Test 5	0.032594	0.032561	0.032381	0.032488	0.032451	0.032471	0.032475
Test 6	0.032514	0.032503	0.032559	0.032523	0.032442	0.032460	0.032458
Test 7	0.032634	0.032733	0.032576	0.032474	0.032506	0.032461	0.032459
Test 8	0.032585	0.032586	0.032607	0.032542	0.032497	0.032454	0.032459
Test 9	0.032605	0.032502	0.032467	0.032523	0.032494	0.032499	0.032457
Test 10	0.032470	0.032561	0.032488	0.032461	0.032484	0.032447	0.032443
Test 11	0.032412	0.032622	0.032537	0.032509	0.032459	0.032459	0.032460
Test 12	0.032796	0.032540	0.032412	0.032558	0.032465	0.032474	0.032449
Test 13	0.032445	0.032652	0.032508	0.032458	0.032485	0.032466	0.032450
Test 14	0.032369	0.032540	0.032564	0.032476	0.032438	0.032451	0.032468
Test 15	0.032547	0.032501	0.032539	0.032530	0.032480	0.032454	0.032456
Test 16	0.032618	0.032504	0.032515	0.032522	0.032450	0.032450	0.032455
Test 17	0.032573	0.032593	0.032438	0.032532	0.032510	0.032478	0.032480
Test 18	0.032424	0.032666	0.032596	0.032517	0.032496	0.032489	0.032453
Test 19	0.032674	0.032491	0.032545	0.032494	0.032494	0.032466	0.032453
Test 20	0.032689	0.032732	0.032483	0.032449	0.032458	0.032457	0.032452
Test 21	0.032731	0.032344	0.032469	0.032401	0.032480	0.032482	0.032465
Test 22	0.032605	0.032711	0.032489	0.032475	0.032435	0.032441	0.032446
Test 23	0.032504	0.032594	0.032454	0.032512	0.032461	0.032446	0.032466
Test 24	0.032677	0.032574	0.032571	0.032550	0.032475	0.032489	0.032448
Test 25	0.032603	0.032591	0.032550	0.032554	0.032483	0.032472	0.032453
Test 26	0.032655	0.032445	0.032605	0.032452	0.032446	0.032475	0.032460
Test 27	0.032493	0.032538	0.032532	0.032558	0.032466	0.032482	0.032464
Test 28	0.032614	0.032575	0.032542	0.032525	0.032420	0.032464	0.032455
Test 29	0.032530	0.032556	0.032482	0.032503	0.032488	0.032456	0.032460
Test 30	0.032630	0.032564	0.032492	0.032481	0.032482	0.032452	0.032454
Max. value	0.032796	0.032733	0.032607	0.032558	0.032519	0.032499	0.032480
Min. Value	0.032337	0.032344	0.032381	0.032344	0.032420	0.032441	0.032443
Max. change	1.419%	1.203%	0.698%	0.662%	0.305%	0.179%	0.114%
Average	0.032568	0.032566	0.032513	0.032491	0.032474	0.032464	0.032458

Table 67: Overview of all conducted tests of Model 2 with reflection at winter solstice

Summer	Grid A	Grid B	Grid C	Grid D	Grid E	Grid F	Grid G
Test 1	0.045924	0.045694	0.045729	0.045709	0.045689	0.045697	0.045666
Test 2	0.045973	0.045786	0.045788	0.045696	0.045689	0.045689	0.045673
Test 3	0.046196	0.045874	0.045693	0.045684	0.045702	0.045679	0.045668
Test 4	0.045948	0.045811	0.045862	0.045730	0.045713	0.045670	0.045670
Test 5	0.045937	0.045811	0.045805	0.045692	0.045691	0.045683	0.045662
Test 6	0.046044	0.045742	0.045830	0.045712	0.045710	0.045697	0.045670
Test 7	0.045993	0.045842	0.045796	0.045727	0.045662	0.045693	0.045673
Test 8	0.046137	0.045778	0.045698	0.045739	0.045699	0.045674	0.045669
Test 9	0.045912	0.045907	0.045815	0.045719	0.045690	0.045680	0.045671
Test 10	0.045855	0.045880	0.045778	0.045691	0.045665	0.045677	0.045666
Test 11	0.045877	0.045872	0.045826	0.045706	0.045715	0.045680	0.045662
Test 12	0.045934	0.045785	0.045740	0.045728	0.045679	0.045687	0.045670
Test 13	0.045847	0.045860	0.045786	0.045691	0.045692	0.045669	0.045669
Test 14	0.045988	0.045804	0.045803	0.045731	0.045692	0.045682	0.045670
Test 15	0.045970	0.045833	0.045708	0.045739	0.045705	0.045686	0.045677
Test 16	0.045934	0.045947	0.045785	0.045728	0.045723	0.045684	0.045674
Test 17	0.046077	0.045772	0.045733	0.045759	0.045715	0.045680	0.045663
Test 18	0.045954	0.045823	0.045747	0.045707	0.045673	0.045685	0.045667
Test 19	0.045939	0.045798	0.045718	0.045767	0.045673	0.045668	0.045682
Test 20	0.046000	0.045809	0.045772	0.045688	0.045696	0.045672	0.045671
Test 21	0.045992	0.045896	0.045754	0.045676	0.045705	0.045679	0.045670
Test 22	0.045843	0.045772	0.045714	0.045666	0.045706	0.045678	0.045667
Test 23	0.046088	0.045826	0.045768	0.045739	0.045680	0.045679	0.045665
Test 24	0.046016	0.045826	0.045809	0.045658	0.045680	0.045692	0.045671
Test 25	0.046147	0.045747	0.045811	0.045734	0.045700	0.045675	0.045679
Test 26	0.045776	0.045844	0.045798	0.045730	0.045696	0.045693	0.045673
Test 27	0.045855	0.045878	0.045819	0.045693	0.045680	0.045684	0.045661
Test 28	0.046014	0.045715	0.045738	0.045732	0.045700	0.045691	0.045671
Test 29	0.045936	0.045898	0.045752	0.045695	0.045661	0.045659	0.045660
Test 30	0.045966	0.045834	0.045738	0.045715	0.045674	0.045676	0.045672
Max. value	0.046196	0.045947	0.045862	0.045767	0.045723	0.045697	0.045682
Min. Value	0.045776	0.045694	0.045693	0.045658	0.045661	0.045659	0.045660
Max. change	0.918%	0.554%	0.370%	0.239%	0.136%	0.083%	0.048%
Average	0.045969	0.045822	0.045770	0.045713	0.045692	0.045681	0.045669

Table 68: Overview of all conducted tests of Model 3 with reflection at summer solstice

Winter	Grid A	Grid B	Grid C	Grid D	Grid E	Grid F	Grid G
Test 1	0.023468	0.023451	0.023351	0.023442	0.023336	0.023339	0.023349
Test 2	0.023587	0.023384	0.023189	0.023398	0.023323	0.023340	0.023338
Test 3	0.023615	0.023353	0.023350	0.023403	0.023344	0.023330	0.023340
Test 4	0.023577	0.023357	0.023440	0.023352	0.023381	0.023341	0.023346
Test 5	0.023548	0.023423	0.023380	0.023351	0.023376	0.023357	0.023344
Test 6	0.023355	0.023362	0.023432	0.023427	0.023403	0.023355	0.023341
Test 7	0.023390	0.023454	0.023355	0.023404	0.023325	0.023328	0.023334
Test 8	0.023304	0.023332	0.023369	0.023432	0.023343	0.023340	0.023339
Test 9	0.023727	0.023360	0.023468	0.023354	0.023370	0.023363	0.023337
Test 10	0.023555	0.023297	0.023315	0.023385	0.023353	0.023357	0.023327
Test 11	0.023625	0.023320	0.023446	0.023394	0.023345	0.023344	0.023339
Test 12	0.023514	0.023381	0.023410	0.023356	0.023399	0.023358	0.023349
Test 13	0.023631	0.023346	0.023389	0.023301	0.023355	0.023339	0.023334
Test 14	0.023451	0.023408	0.023417	0.023417	0.023397	0.023338	0.023332
Test 15	0.023517	0.023362	0.023349	0.023367	0.023347	0.023350	0.023348
Test 16	0.023583	0.023421	0.023388	0.023364	0.023347	0.023328	0.023339
Test 17	0.023300	0.023436	0.023413	0.023322	0.023378	0.023342	0.023328
Test 18	0.023393	0.023516	0.023328	0.023383	0.023339	0.023337	0.023341
Test 19	0.023533	0.023372	0.023449	0.023342	0.023362	0.023338	0.023338
Test 20	0.023463	0.023339	0.023392	0.023320	0.023385	0.023339	0.023337
Test 21	0.023443	0.023478	0.023383	0.023480	0.023348	0.023363	0.023335
Test 22	0.023520	0.023377	0.023429	0.023412	0.023319	0.023351	0.023333
Test 23	0.023361	0.023359	0.023379	0.023340	0.023347	0.023372	0.023332
Test 24	0.023491	0.023438	0.023379	0.023385	0.023341	0.023345	0.023355
Test 25	0.023570	0.023366	0.023375	0.023408	0.023370	0.023363	0.023344
Test 26	0.023593	0.023412	0.023445	0.023423	0.023338	0.023349	0.023333
Test 27	0.023372	0.023313	0.023421	0.023405	0.023348	0.023339	0.023332
Test 28	0.023594	0.023393	0.023412	0.023339	0.023340	0.023337	0.023337
Test 29	0.023388	0.023300	0.023309	0.023368	0.023343	0.023338	0.023352
Test 30	0.023339	0.023476	0.023381	0.023335	0.023340	0.023349	0.023347
Max. value	0.023727	0.023516	0.023468	0.023480	0.023403	0.023372	0.023355
Min. Value	0.023300	0.023297	0.023189	0.023301	0.023319	0.023328	0.023327
Max. change	1.833%	0.940%	1.203%	0.768%	0.360%	0.189%	0.120%
Average	0.023494	0.023386	0.023385	0.023380	0.023355	0.023346	0.023339

Table 69: Overview of all conducted tests of Model 3 with reflection at winter solstice

Summer	Grid A	Grid B	Grid C	Grid D	Grid E	Grid F	Grid G
Test 1	0.033140	0.033010	0.032994	0.033017	0.032970	0.032977	0.032949
Test 2	0.033231	0.033039	0.032994	0.032990	0.032963	0.032960	0.032945
Test 3	0.033214	0.033120	0.032975	0.033026	0.032965	0.032983	0.032950
Test 4	0.033233	0.033226	0.033031	0.033005	0.032979	0.032969	0.032955
Test 5	0.033192	0.033083	0.032974	0.033005	0.032972	0.032962	0.032951
Test 6	0.033312	0.033188	0.033000	0.033028	0.033000	0.032962	0.032950
Test 7	0.033168	0.033136	0.033043	0.033034	0.032954	0.032951	0.032945
Test 8	0.033147	0.033074	0.032999	0.033017	0.032946	0.032958	0.032947
Test 9	0.033227	0.033042	0.033051	0.033018	0.032983	0.032958	0.032946
Test 10	0.033273	0.033088	0.033014	0.033016	0.032963	0.032969	0.032958
Test 11	0.033140	0.033251	0.033070	0.032975	0.032964	0.032959	0.032949
Test 12	0.033234	0.033017	0.033019	0.033012	0.032965	0.032954	0.032946
Test 13	0.033190	0.032969	0.033043	0.032950	0.032993	0.032968	0.032947
Test 14	0.033163	0.033044	0.032997	0.033028	0.032945	0.032955	0.032950
Test 15	0.033189	0.033085	0.033031	0.033032	0.032965	0.032968	0.032953
Test 16	0.033146	0.033057	0.033027	0.032992	0.032971	0.032955	0.032949
Test 17	0.033154	0.033054	0.032994	0.032972	0.032951	0.032968	0.032954
Test 18	0.033162	0.033127	0.032997	0.032982	0.032973	0.032943	0.032940
Test 19	0.033178	0.033085	0.033026	0.033034	0.032962	0.032955	0.032948
Test 20	0.033241	0.033047	0.033047	0.033017	0.032975	0.032963	0.032953
Test 21	0.033096	0.033125	0.033054	0.033021	0.032981	0.032942	0.032952
Test 22	0.033123	0.033092	0.033072	0.033007	0.032966	0.032957	0.032955
Test 23	0.033189	0.033066	0.033029	0.033009	0.032980	0.032960	0.032943
Test 24	0.033142	0.033090	0.033069	0.033007	0.032985	0.032967	0.032956
Test 25	0.033249	0.033036	0.033100	0.032983	0.032972	0.032946	0.032953
Test 26	0.033181	0.033159	0.032980	0.032992	0.032964	0.032965	0.032949
Test 27	0.033217	0.033093	0.033032	0.032978	0.032953	0.032966	0.032954
Test 28	0.033171	0.033053	0.033034	0.033002	0.032985	0.032960	0.032956
Test 29	0.033206	0.033073	0.033033	0.032987	0.032971	0.032943	0.032949
Test 30	0.033128	0.033044	0.032988	0.033003	0.032967	0.032957	0.032950
Max. value	0.033312	0.033251	0.033100	0.033034	0.033000	0.032983	0.032958
Min. Value	0.033096	0.032969	0.032974	0.032950	0.032945	0.032942	0.032940
Max. change	0.653%	0.855%	0.382%	0.255%	0.167%	0.124%	0.055%
Average	0.033188	0.033086	0.033024	0.033005	0.032969	0.032960	0.032950

Table 70: Overview of all conducted tests of Model 4 with reflection at summer solstice

Winter	Grid A	Grid B	Grid C	Grid D	Grid E	Grid F	Grid G
Test 1	0.016027	0.016060	0.015877	0.015920	0.015939	0.015909	0.015905
Test 2	0.015969	0.015946	0.016015	0.015933	0.015901	0.015910	0.015912
Test 3	0.016094	0.015937	0.015966	0.015944	0.015940	0.015915	0.015908
Test 4	0.016067	0.015958	0.015945	0.015959	0.015943	0.015912	0.015914
Test 5	0.016018	0.016042	0.015971	0.015909	0.015896	0.015919	0.015907
Test 6	0.015977	0.015974	0.015942	0.015975	0.015923	0.015931	0.015914
Test 7	0.015948	0.016012	0.016023	0.015913	0.015919	0.015902	0.015907
Test 8	0.015993	0.015949	0.015924	0.015930	0.015934	0.015901	0.015907
Test 9	0.016034	0.015928	0.016003	0.015920	0.015910	0.015905	0.015905
Test 10	0.015823	0.015950	0.015951	0.015953	0.015917	0.015918	0.015910
Test 11	0.016055	0.015946	0.015924	0.015912	0.015911	0.015903	0.015908
Test 12	0.016144	0.015898	0.015926	0.015965	0.015921	0.015916	0.015910
Test 13	0.016010	0.015934	0.015931	0.015969	0.015934	0.015919	0.015911
Test 14	0.016000	0.016013	0.015908	0.015916	0.015927	0.015913	0.015895
Test 15	0.015970	0.015966	0.015967	0.015958	0.015906	0.015921	0.015894
Test 16	0.016202	0.015906	0.016006	0.015952	0.015908	0.015918	0.015897
Test 17	0.015970	0.015988	0.015921	0.015892	0.015937	0.015904	0.015916
Test 18	0.016060	0.015980	0.015947	0.015949	0.015911	0.015925	0.015906
Test 19	0.016008	0.016003	0.015948	0.015950	0.015917	0.015917	0.015901
Test 20	0.015970	0.016078	0.015896	0.015886	0.015890	0.015898	0.015908
Test 21	0.015950	0.015909	0.015917	0.015978	0.015938	0.015913	0.015914
Test 22	0.016019	0.015970	0.015971	0.015923	0.015916	0.015907	0.015905
Test 23	0.016064	0.016002	0.015838	0.015969	0.015929	0.015901	0.015908
Test 24	0.015927	0.015953	0.015940	0.015981	0.015891	0.015917	0.015911
Test 25	0.015973	0.015954	0.015956	0.015933	0.015929	0.015910	0.015912
Test 26	0.015976	0.015987	0.015853	0.015970	0.015924	0.015921	0.015910
Test 27	0.016047	0.016009	0.015979	0.015892	0.015909	0.015900	0.015905
Test 28	0.016158	0.015963	0.015940	0.015944	0.015931	0.015930	0.015908
Test 29	0.015967	0.015885	0.015918	0.015946	0.015903	0.015901	0.015911
Test 30	0.016016	0.015805	0.015960	0.015949	0.015931	0.015904	0.015907
Max. value	0.016202	0.016078	0.016023	0.015981	0.015943	0.015931	0.015916
Min. Value	0.015823	0.015805	0.015838	0.015886	0.015890	0.015898	0.015894
Max. change	2.395%	1.727%	1.168%	0.598%	0.334%	0.208%	0.138%
Average	0.016015	0.015964	0.015942	0.015940	0.015920	0.015912	0.015908

Table 71: Overview of all conducted tests of Model 4 with reflection at winter solstice

	Grid A	Grid B	Grid C	Grid D	Grid E	Grid F	Grid G
Model 1	36	37	38	49	54	81	228
Model 2	40	44	46	54	65	98	276
Model 3	66	70	73	83	96	132	342
Model 4	144	156	162	168	192	222	456

Table 72: Computational time in seconds per Grid and Model

Number of sensors	77	150	308	640	2183	4928	19824
Summer Model 1	0.286%	0.091%	0.143%	0.064%	0.053%	0.025%	0.015%
Summer Model 2	0.456%	0.191%	0.162%	0.119%	0.083%	0.035%	0.022%
Summer Model 3	0.494%	0.280%	0.200%	0.120%	0.068%	0.049%	0.028%
Summer Model 4	0.374%	0.499%	0.230%	0.166%	0.093%	0.070%	0.031%
Winter Model 1	0.452%	0.404%	0.346%	0.241%	0.108%	0.109%	0.051%
Winter Model 2	0.709%	0.681%	0.407%	0.453%	0.166%	0.107%	0.069%
Winter Model 3	0.994%	0.555%	0.837%	0.426%	0.207%	0.113%	0.067%
Winter Model 4	1.196%	0.993%	0.653%	0.337%	0.185%	0.119%	0.085%

Table 73: Overview of the relative random error size in all models.

Appendix H: Detailed explanation Grasshopper model

This chapter explains the model in depth. Each of the model components is discussed chronologically. First, the variables and input parameters, which form the basis of subsequent steps, are discussed, and followed by the creation of the core geometry. After the geometry is created, the criteria related to transparency, insulation, daylight control, and the structural constrain are highlighted.

As a small introduction to Grasshopper Rhino, the components in Grasshopper are, in essence, performing some kind of manipulation on the inputs to create the output. For example, a component requires two numerical inputs and determines the maximum value, which is the output.

Input

The first thing that has to be defined in the model are the inputs. The inputs are divided into three groups: variables, geometrical input, and optimisation input. Each of these three groups is discussed in the order mentioned above.

Variables

The variables are shown in Figure 198 where they are represented as so-called sliders. The sliders in Grasshopper require defining the maximum, minimum, and step sizes. For example, the angle of the core lies between zero and forty-five degrees, with steps of five degrees. Also, these sliders are connected to the genetic algorithm component. This component can change the values of the variables during the optimisation process, allowing it to find the most optimal solution for a predefined fitness function.

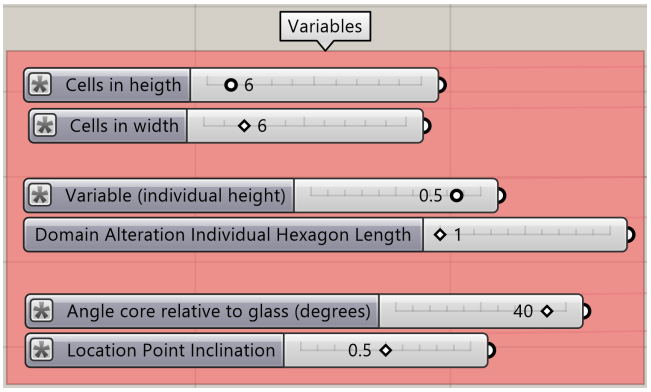


Figure 198: The defined variables in Grasshopper Rhino

Geometrical Input

The geometrical input is displayed in Figure 199, which is also represented by sliders. However, in contrast to the variables, these inputs do not change during the optimisation process. Next, these seven geometrical inputs, together with the variables, define the geometrical properties of the panel. The orientation input has two values, which allows for two rotations of the core design.

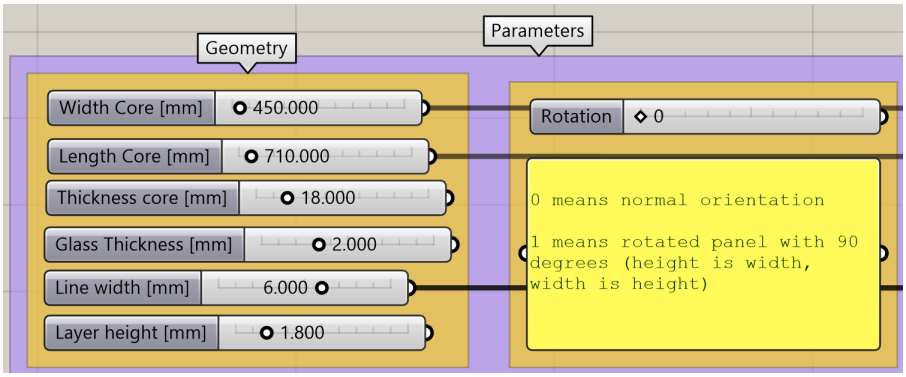


Figure 199: The geometrical inputs in Grasshopper Rhino

Optimisation Input

The optimisation input is the required information for all the criteria and constraints as well as the optimisation process. For example, the structural constraint requires the material properties, the magnitude of the surface load, and the boundary conditions. The boundary conditions can be supported at the bottom and top, left and right, or both. To enhance the user experience, a button is created that by means of a true or false value, controls the boundary conditions.

Subsequently, due to the optional rotation of the core design, it has to be enforced that whatever rotation is used, the true width, length, hexagons vertically, hexagons horizontally, and support placement are used. This process is shown in Figure 201.

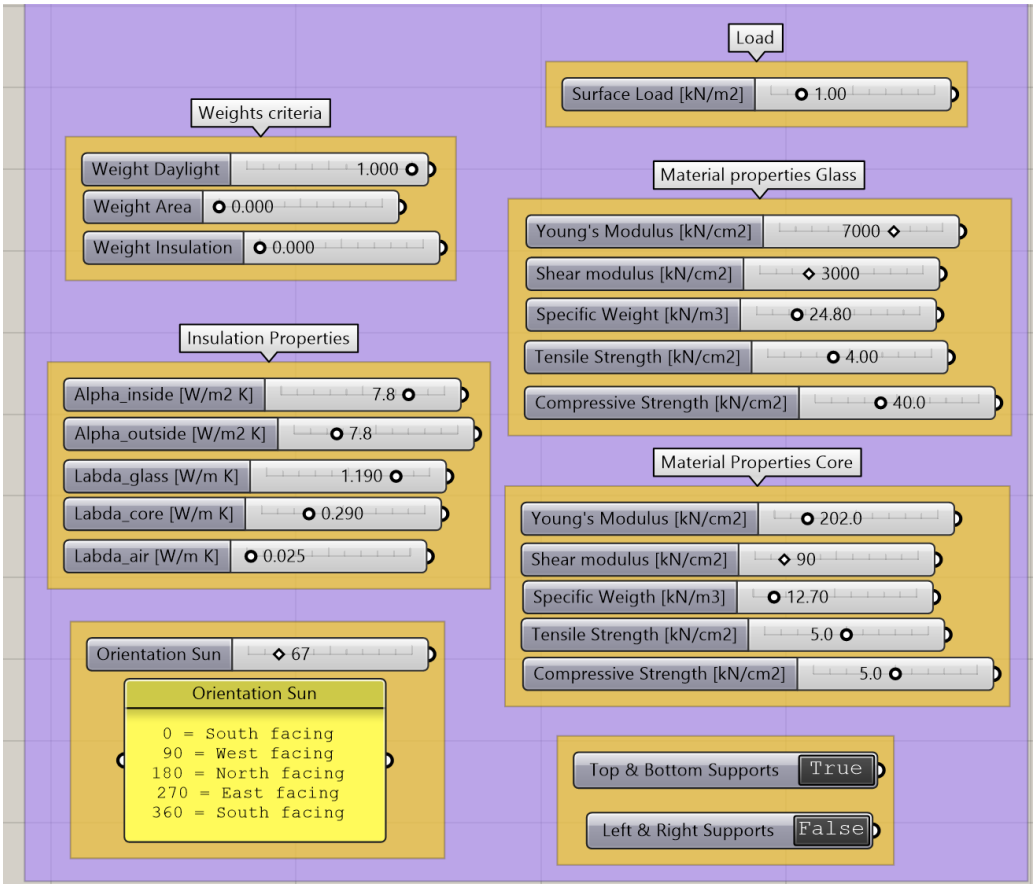


Figure 200: The optimisation inputs in Grasshopper Rhino

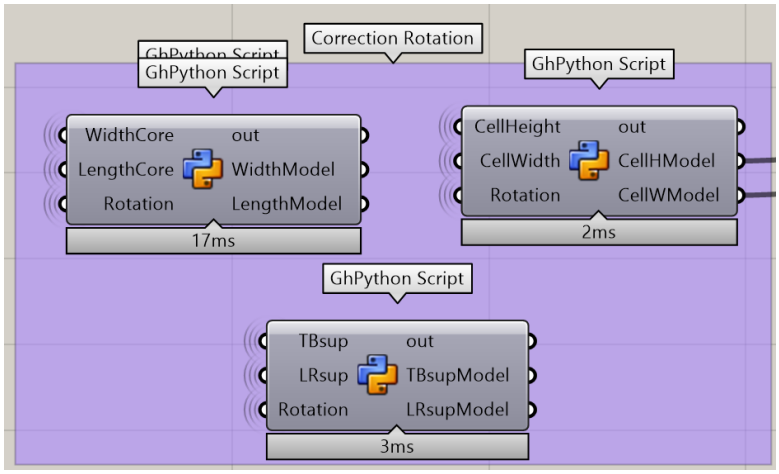


Figure 201: Imposing the correct orientation of the panel in Grasshopper Rhino

Panel Creation

The core geometry is created in a basic form and subsequently altered to reach the final geometry, which includes the various variables, such as inclination and individual hexagon height, as well as the constraint that one layer should be printed in one continuous path. All these steps are discussed in the next sub-chapters.

Creation honeycomb pattern

The honeycomb pattern is created with the Grasshopper component 'Hexagon Cells' from the Lunchbox plug-in. This component requires the number of hexagons vertically and horizontally, the hexagon shape, and the surface area. The surface area is equal to the width of the panel and the height of the panel minus the line width multiplied. The line width is subtracted from the height because the line width is not yet included in the basic honeycomb pattern. The hexagon shape, represented by an hexagon angle, is chosen to be equal to 0.27. This is not considered a general input parameter; however, it can be altered in case a specific hexagon shape is desired. Finally, the internal angle is calculated for the hexagon, which is half of the angle between the two inclined lines. This is used in subsequent components. These steps are displayed in Figure 202, and the obtained shape is shown in Figure 203.

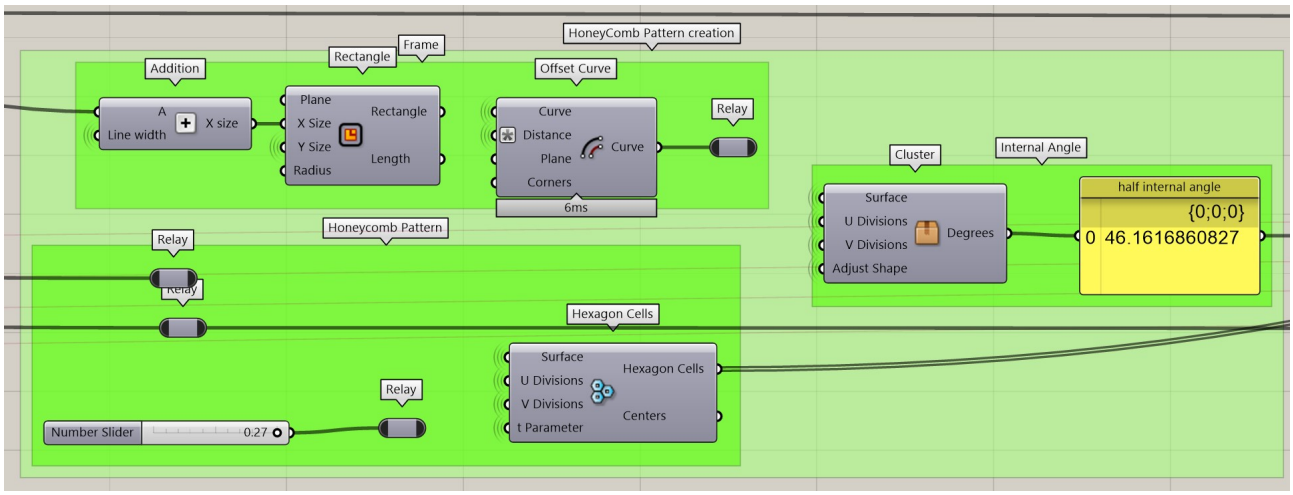


Figure 202: The required steps to create the basic honeycomb pattern

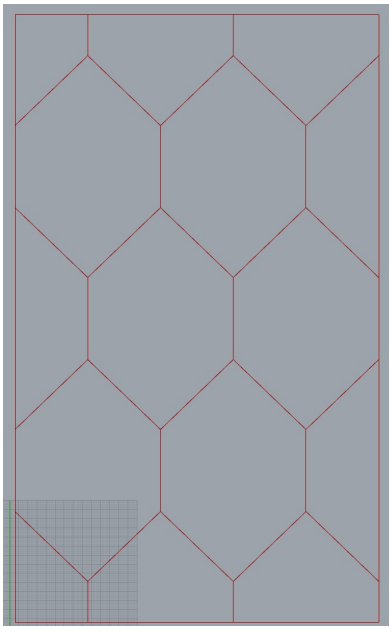


Figure 203: The basic honeycomb pattern

Adding line thickness

The next step is to include the line width in the hexagon pattern, which is shown in Figures 204 and 205. The line width is added by offsetting the center lines, as shown in Figure 203. The lines are offset by dividing the honeycomb pattern into straight lines, grouping them based on position and orientation. Then, the vertical lines in the core are moved to the left and right to obtain the line width as defined in the geometrical inputs.

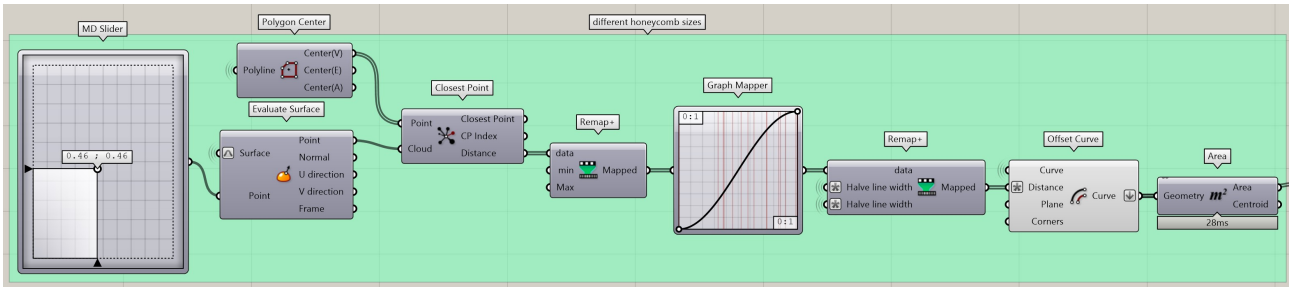


Figure 204: The required steps to add the line width

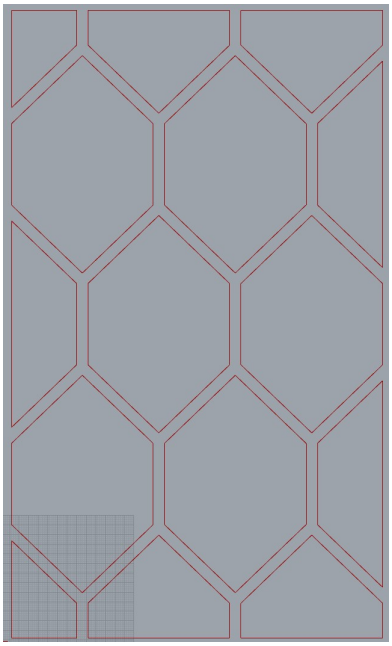


Figure 205: The basic honeycomb pattern with line width

The subsequent alteration of the basic honeycomb pattern is to have a double line width for the vertical hexagon lines. This is a result of the constraint to print a core layer continuously. The first part is to cut the pattern into straight lines and determine the vertical and diagonal lines. The vertical lines are obtained by filtering on their angle, as all vertical lines have either an angle of 0 or 180 degrees. It is required to include both angles because of the different positions of the start and end points of each line. Additionally, the diagonals are filtered based on the previously calculated internal angle. Two different angles are used because of the two different orientations due to the position of the start and end points of each line. Also, a 'Smaller Than' component is used for one of the orientations. This component is introduced because Grasshopper rounds off numbers slightly differently within certain components, which creates a small deviation between them. Thus, in order to prevent errors from occurring, this additional step is introduced. The last segments are the horizontal lines and are also filtered based on their angle, which is equal to 90 degrees. All steps are visible in Figures 206, 207, and 208.

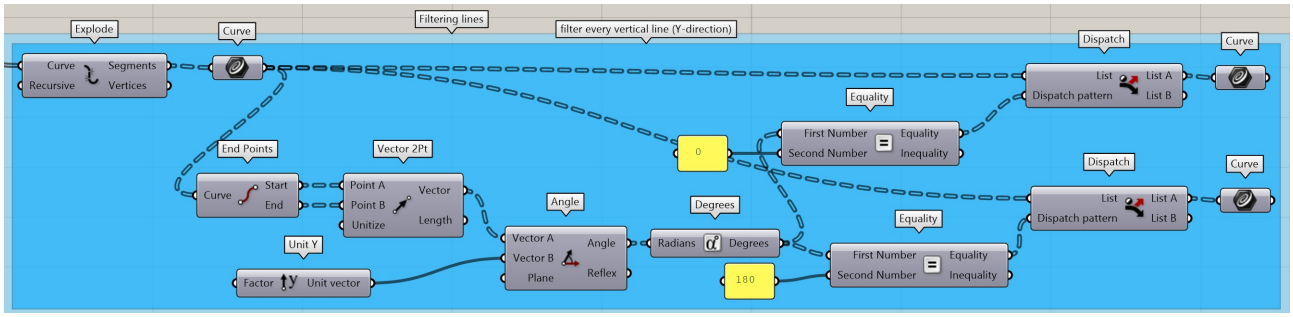


Figure 206: The filtering process to obtain the vertical lines

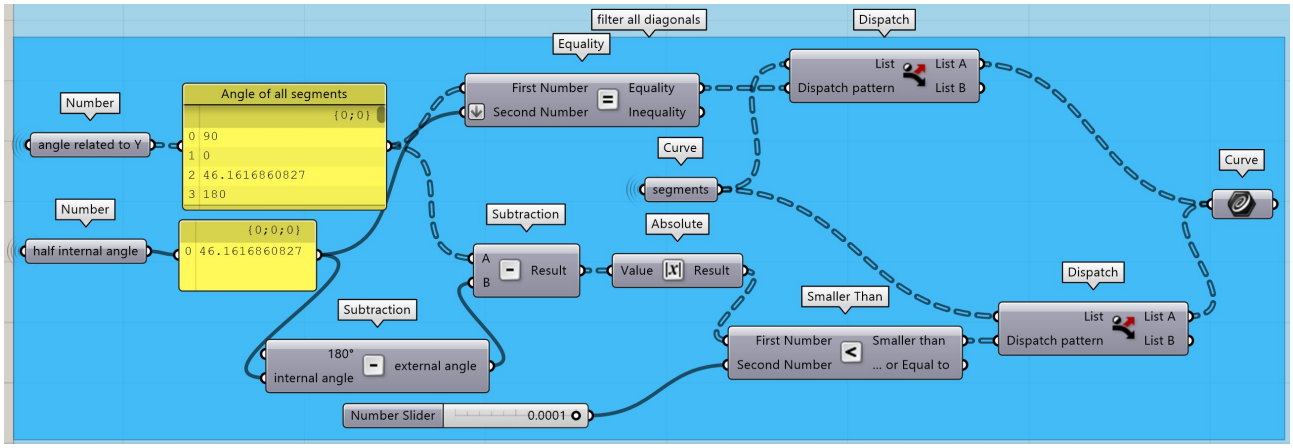


Figure 207: The filtering process to obtain the vertical lines

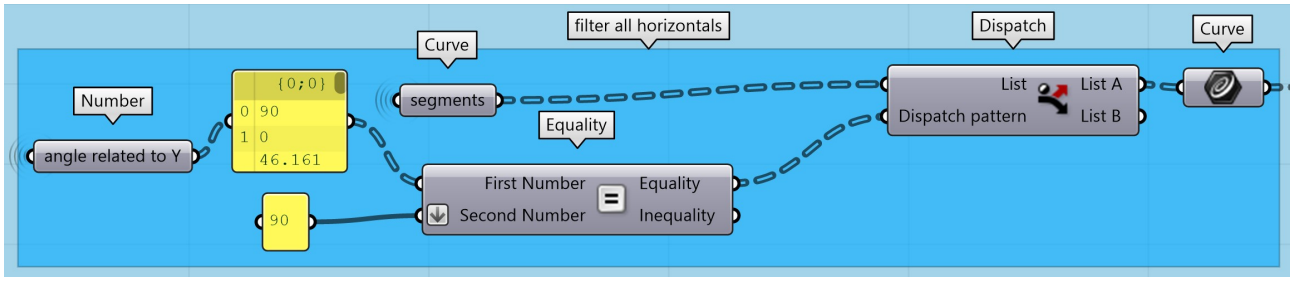


Figure 208: The filtering process to obtain the vertical lines

The next step is to offset the vertical lines in order to represent a double line width. The vertical lines are split into two groups based on the direction of their movement. Additionally, the length of the vertical lines has to change in order to connect with the diagonal parts. The distance between them is calculated, and the vertical lines are extended. This process is shown in Figure 209. However, not all lines need the same alteration in length, which means that some extensions have to be removed.

In order to determine if and which lines are too long, a component called 'Line-Line Intersection' from the plug-in Karamba3D is used. This component finds the intersections between lines and divides the lines into sub-segments that do not intersect. First, the diagonal lines and vertical lines are used to find intersections. Subsequently, the lines smaller than twice the line extension from the last step are removed. This criterion is used because, based on the same reasoning that between some vertical lines and the diagonal lines, a gap exists of one line extension, some vertical lines are one line extension too long because of the different orientation of the diagonal lines. If the angle between the diagonal line and the vertical line is smaller than 90 degrees, the line will be too long. It is the other way around if the angle is greater than 90 degrees.

Next, the horizontal lines are added, and again, the 'Line-Line intersection' component is used. In this case, the lines shorter than the maximum of one line extension or half of the line width are being removed because there are only intersections at the vertical lines at the bottom and top. Finally, a control step is executed in

order to only receive the lines that are connected to each other. The process is split into two steps because the vertical lines at the bottom and top are relatively short compared to the other lines, therefore requiring another minimum length criterion. If the steps are not divided, some lines will be removed incorrectly in some cases, for example, with a large number of hexagons along the width and a very small number of hexagons along the height. Additionally, this process of removing puts constraints on the maximum number of hexagons, both vertically and horizontally, in order to prevent wrongly removed lines. Problems start to occur when the line width and minimum line length obtain similar values. These last steps are displayed in Figure 210, and the shape of the current core is visible in Figure 211.

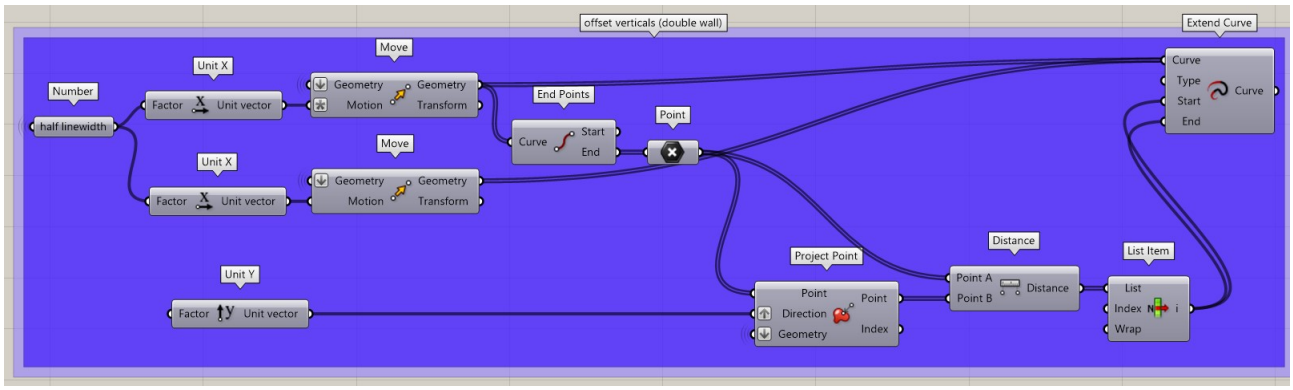


Figure 209: Offsetting the vertical lines

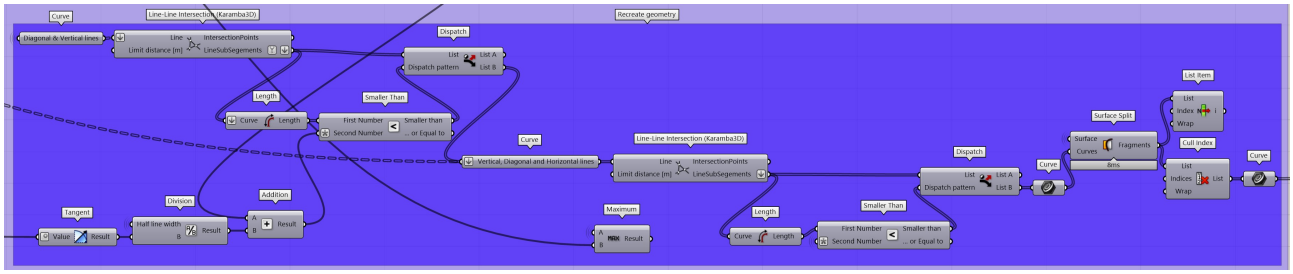


Figure 210: Reconnecting all the lines

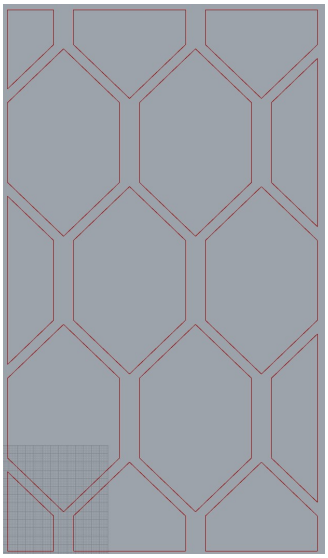


Figure 211: The current shape of the core

Altering hexagon sizes

The next part is to include the variable related to the individual hexagon height, which basically changes the vertical lines of the hexagons in order to influence the height. The first step is to divide the lines again into vertical, horizontal, and diagonal lines. The process is also based on the angles displayed in Figure 212.

Furthermore, the vertical lines on the outside are obtained, and a data tree is created in order to group the vertical lines. The outside lines are obtained by filtering based on the x-coordinate (width). In order to have symmetrical hexagons, all the vertical lines at a certain height (y-axis) have to remain the same size. For example, in the case where four hexagons exist vertically, as displayed in the figures above, five groups of vertical lines exist. The component 'graft tree' creates this data tree. The steps are shown in Figure 213.

Additionally, the other vertical lines are added to their group; see Figure 214. The division of the vertical lines is made through the end points of the vertical lines at the outside, as the center of all vertical lines in the relevant group should be between these two end points. The maximum and minimum y-coordinate (height) are determined. These values are used to construct a domain, and in the subsequent component 'includes' it calculates for each vertical line if the center point is within these domains. The center point is found by using the component 'Point on Curve' and setting the value to 0.500, which is the center of the line. Next, through the component 'dispatch', the relevant vertical lines are stored in the data tree.

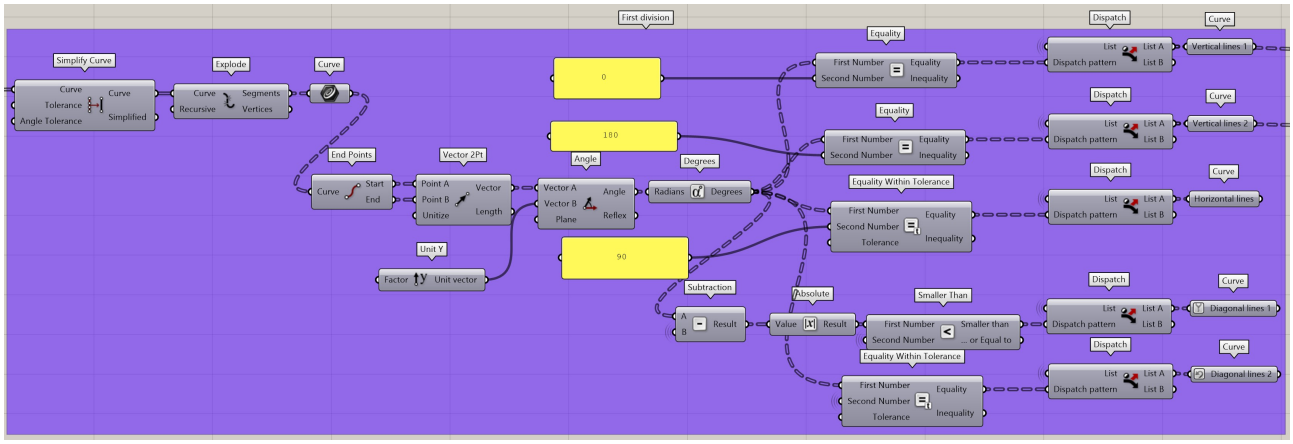


Figure 212: Dividing the core in horizontal, vertical, and diagonal lines

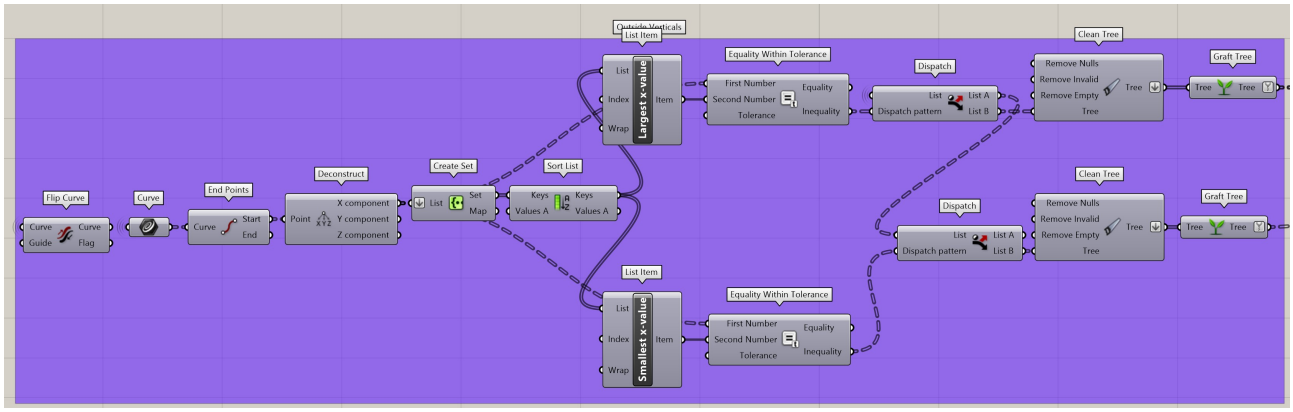


Figure 213: Obtaining the vertical lines on the outside and create groups

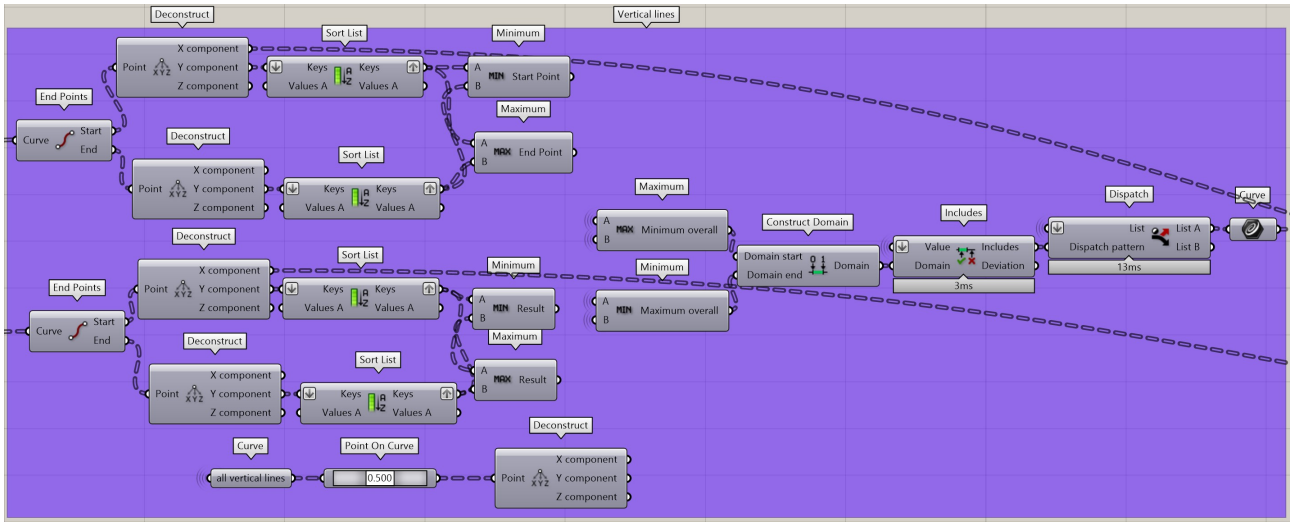


Figure 214: Adding all vertical lines in their group

After creating a data tree that orders the vertical lines, the boundary thickness is corrected. As a consequence of the continuous printing constraint, some sections of the boundary lines of the core structure have twice the line width.

The first step is to separate the boundary lines on either side from the other vertical lines. This is done with the same steps as in the previous component and is displayed in Figure 215. This is followed by selecting the line sections that have an altered line width, as shown in Figure 216. The boundary sections that have a double line width are the sections that have two adjacent diagonal lines; for example, at the right boundary, the green boundary lines have a double line width in Figure 217. In order to select these boundary lines, the other vertical lines are used, because if you are sorting the list of n vertical lines based on their x-coordinate (width), then the third line, or the n-3 line, will have the same y-coordinate as the boundary section, which needs to be adjusted. This division is shown in the right part of Figure 216, and already the midpoints of these lines as well as the boundary lines are calculated and deconstructed into their x, y, and z coordinates. The two line groups in the middle represent the boundary lines.

The third and final step is to group the green lines on either side of the boundary and move them by one line width. Thereafter, all the vertical lines are put together as preparation for the next step. This is visible in Figure 218.

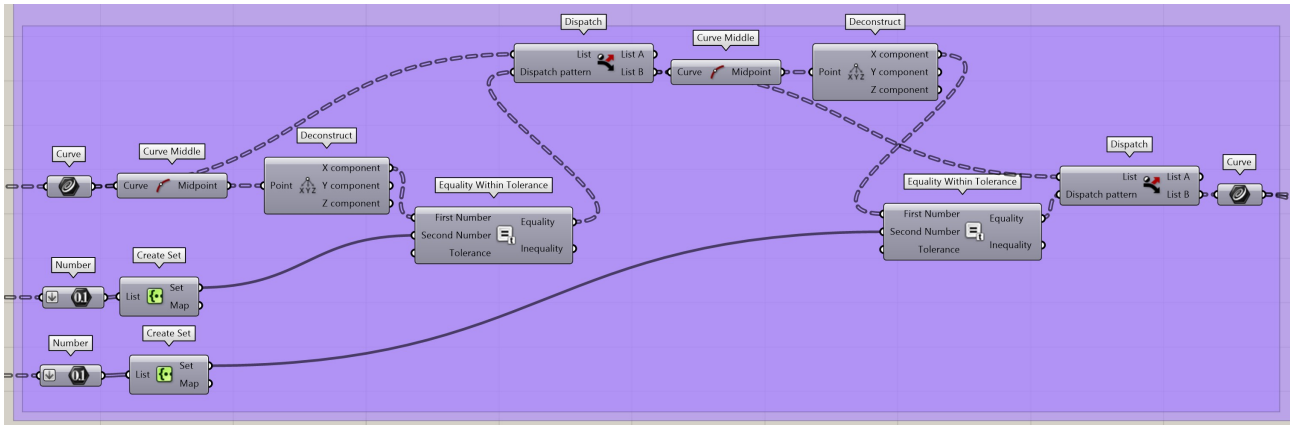


Figure 215: Grouping the vertical lines.

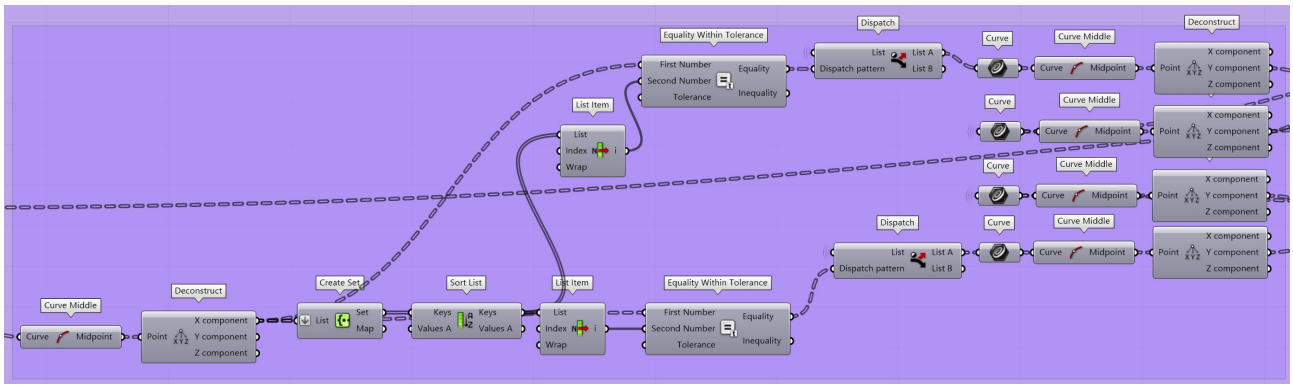


Figure 216: Determining which sections need an altered line width.

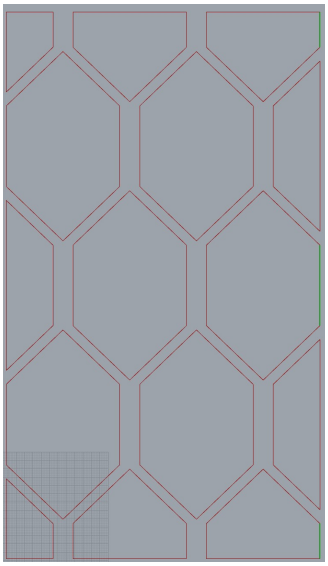


Figure 217: The sections that need an alteration in line width

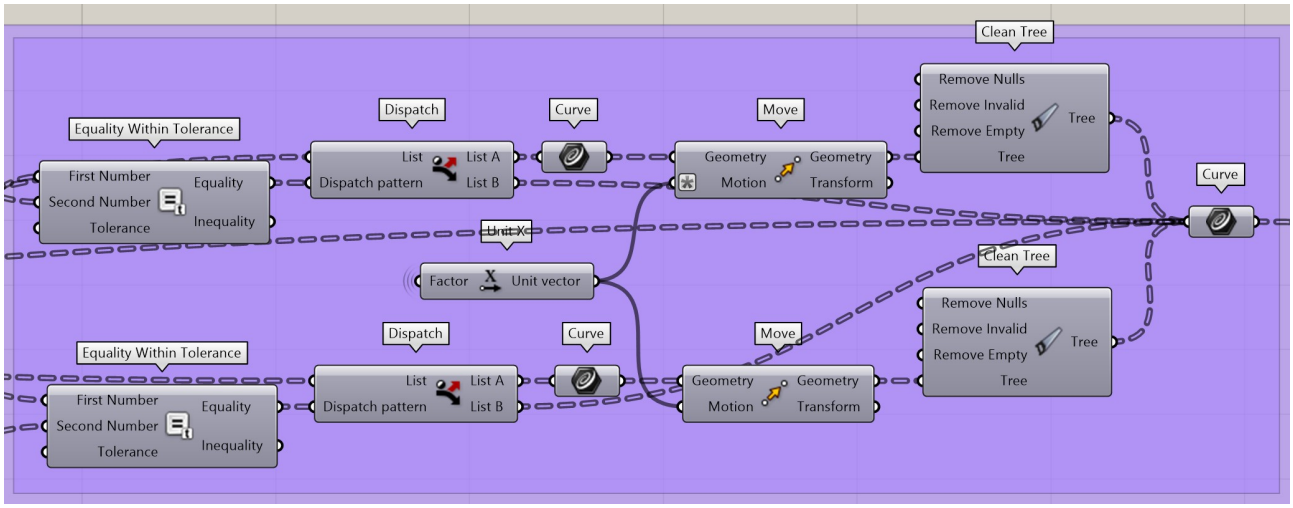


Figure 218: Changing the line width of some boundary sections.

The subsequent step is to define how the vertical line length will be altered with the use of the relevant variable. The method used to change the length is based on a sine function, sin(a), between zero and n*pi. The n value is directly obtained from the variable 'Domain alteration', which is equal to 1, 2, or 3. Additionally, a data

series is created between zero and one, where the number of steps is equal to the number of hexagons vertically. For example, with 5 steps and n=1, the series is represented by 0, 0.25, 0.5, 0.75, and 1. Subsequently, the previously defined sine curve is used to map the data series, where a is equal to the data series multiplied by pi, such that a data series value of 0.5 is mapped to $sin(0.5\pi)$ which is 1.0. This mapping creates a symmetrical result. Finally, the variable is introduced to relate the data series to the individual hexagon height. This step constrains the maximum height to 170 % and the minimum height to 30 % of the original height. The limits are defined by the variable and should be chosen carefully as another line conversion is applied. This process is explained in depth in Chapter 10. Moreover, if the variable is positive, the hexagons in the center of the panel will have an increased height and the hexagons along the sides a decreased height, and vice versa. This is shown in Figure 219.

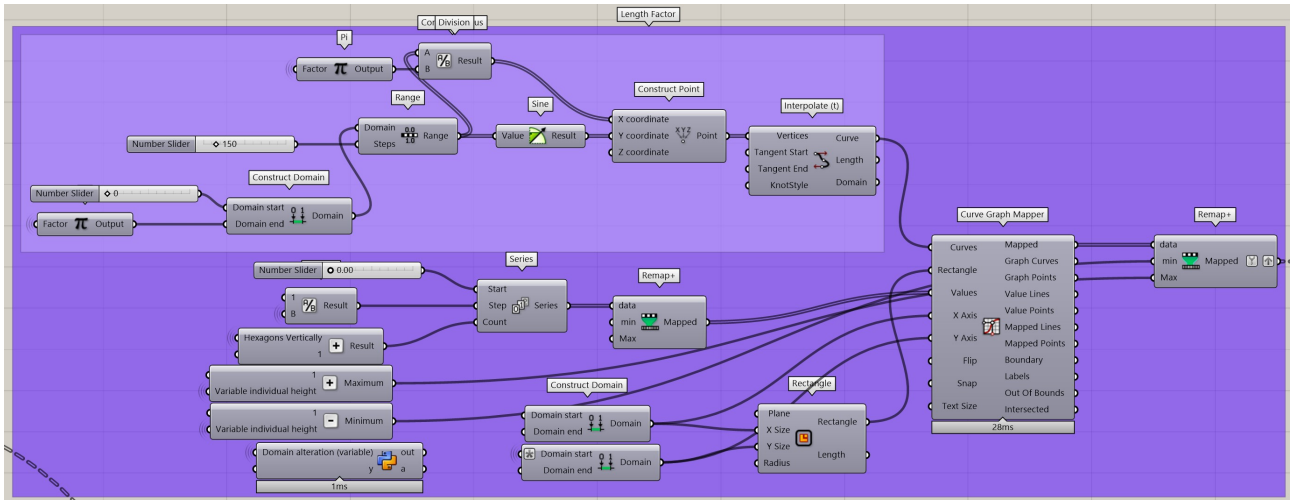


Figure 219: Defining the change in the line length.

Then the lines need to be altered, while the sum of the lengths has to remain the same. This is induced by first calculating the sum of the lengths of the consecutive vertical lines. Additionally, the vertical line length is multiplied by the relative length alteration factor, and again, the sum of these lengths is taken. Next, the sum of the altered lengths is divided by the unaltered lengths. Subsequently, the altered lengths are divided by this number, which implies that the sum of these new lengths will have the same total length as the unaltered lengths while including the alteration factor. Furthermore, the difference between the new lengths and the original lengths is calculated. This difference is used to alter the length of all the vertical lines in the pre-defined data tree, where each vertical line inside one group requires the same absolute length alteration. After scaling all the vertical lines, the vertical and diagonal lines have to be moved in order to connect to each other again. The movement of each line can be determined based on each length alteration. However, the moved boundary lines, which have a double line width, require additional steps to be connected. All these steps are made in Figures 221 and 220.

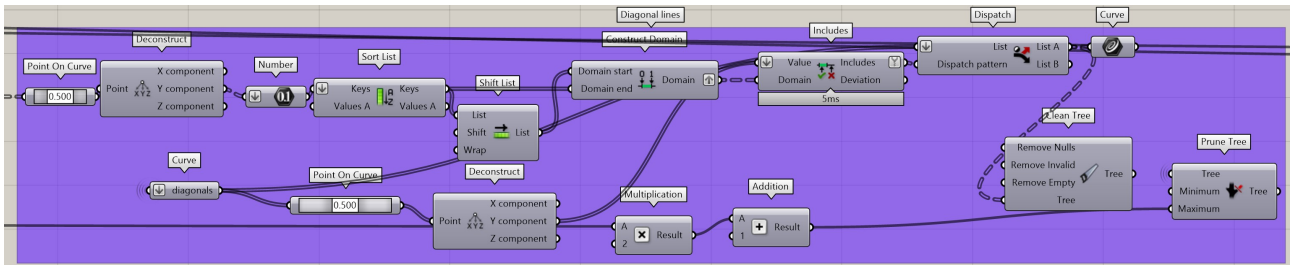


Figure 220: Grouping the Diagonal lines

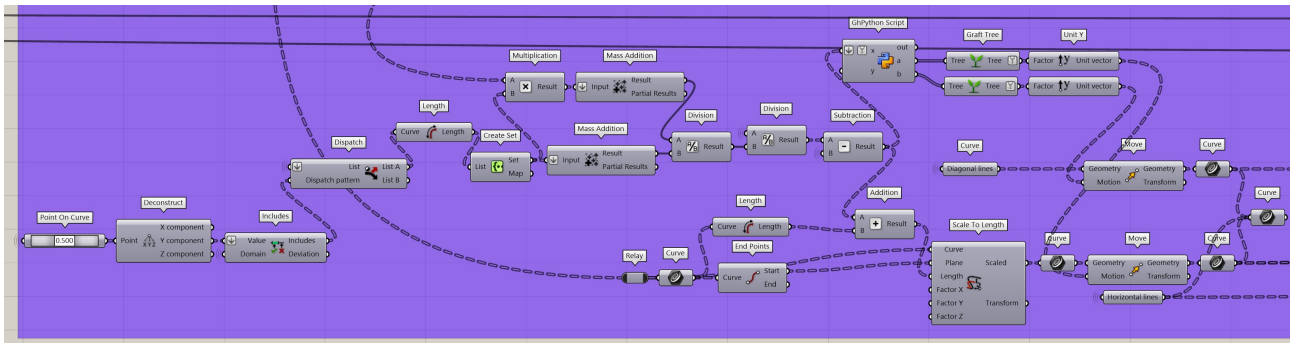


Figure 221: First part of reconnecting geometry

The length increase required by these lines can be determined because the horizontal movement, the line width, and the angle of the diagonal lines are known; thus, by using the tangent of that angle and multiplying it with the line width, the required length increase is known. This number is used to extend all vertical lines at their start and end points. Next, the 'Line-Line Intersection' component is used, and the length of all segments is calculated. Then, the list is dispatched by using an equality filter. If the line segment is equal to the length increase, the line will be removed. Secondly, all lines are removed that are equal to the length of the line segment of the diagonal line, which appeared due to the movement of the boundary lines. Thirdly, all line segments are removed that are equal to the line width, which are horizontal segments that emerged after moving some boundary lines. This is visible in Figure 222, and Figure 223 shows the end result of the geometry. It needs to be noted that if a line section, which should not be removed, is equal to one of the previously mentioned methods of removal, a line will be incorrectly removed, and causing a discrepancy in the geometry that will lead to errors. This is one of the limitations of the model and requires extra caution in the case of a relatively large line width in combination with a small panel.

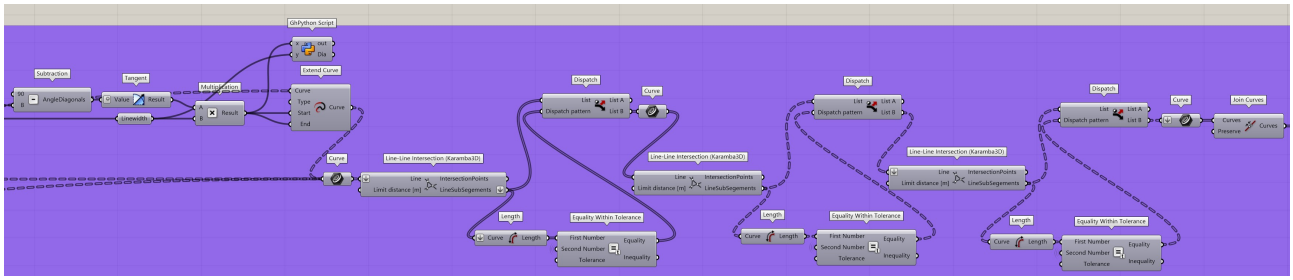


Figure 222: Second part of reconnecting geometry

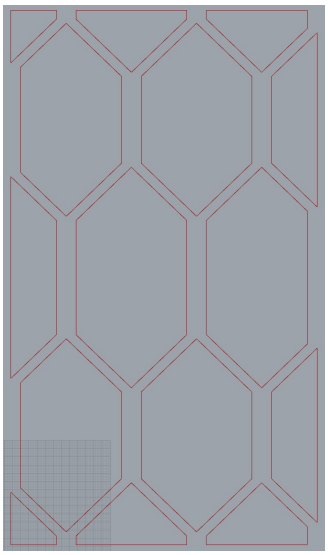


Figure 223: New geometry

3D pattern with inclination

The last alteration to the geometry is to add inclination. The lines are inclined in the y direction, because the inclination is included as a feature that affects direct and indirect sunlight. Thus, the vertical axis will influence this the most. The inclination is bound by 45 degrees because the core structure needs to be printable without additional supports during the printing process. The inclination is added by scaling the geometry differently along the layers and moving them in the z-direction (panel thickness).

First, the scaling factor needs to be found. The scaling factor is related to the desired maximum angle and the line width. The difference in y-coordinate of the layers can be written as:

$$Ma = tan(\alpha) \cdot (n-1) \cdot \frac{lw}{2} \tag{53}$$

Where:

1. Ma is the difference in y-coordinate of layer n with layer n = 1.
2. α is the defined maximum angle of the geometry [degrees].
3. n is the layer number.
4. lw is the line width [mm].

Additionally, the scaling factor depends on the size of the panel. The panel size is determined by the end points of the horizontal lines, which are found in Figure 212, and this process is shown in Figure 224. The scaling is performed around a defined point within the boundaries of the panel, for example, the center point. Then, the inclination of the core depends on the vertical (y) distance between this point and the particular section. Thus, at the maximum possible distance, the defined angle has to be met, and the formulation of the scaling factor is given in Equation (54). The calculation of the scaling factor is presented in Figure 225.

$$S = \frac{Ma}{n \cdot Md} \tag{54}$$

Where:

- S is the scaling factor
- Md is the maximum distance between the reference point and the boundaries.

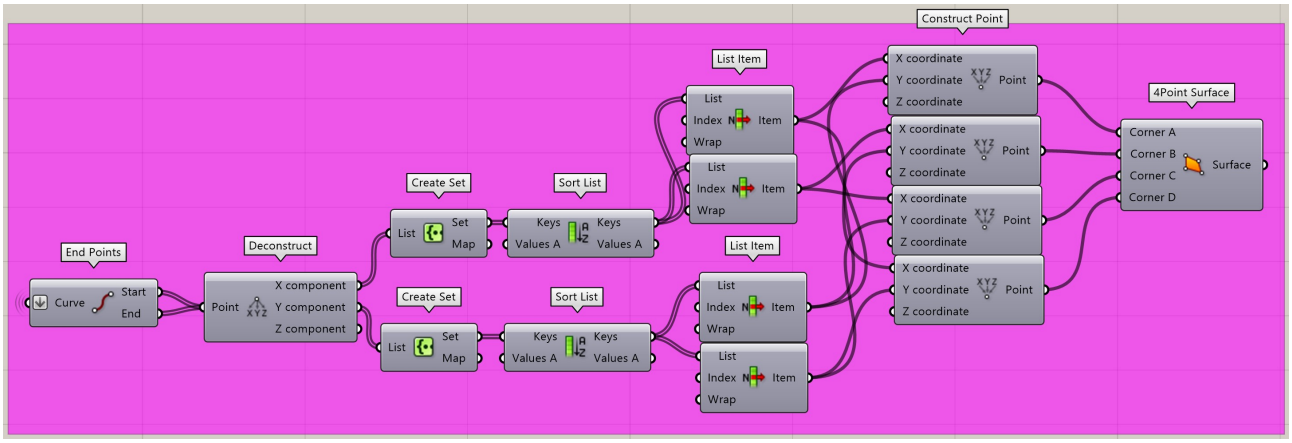


Figure 224: Creation surface panel for scaling factor

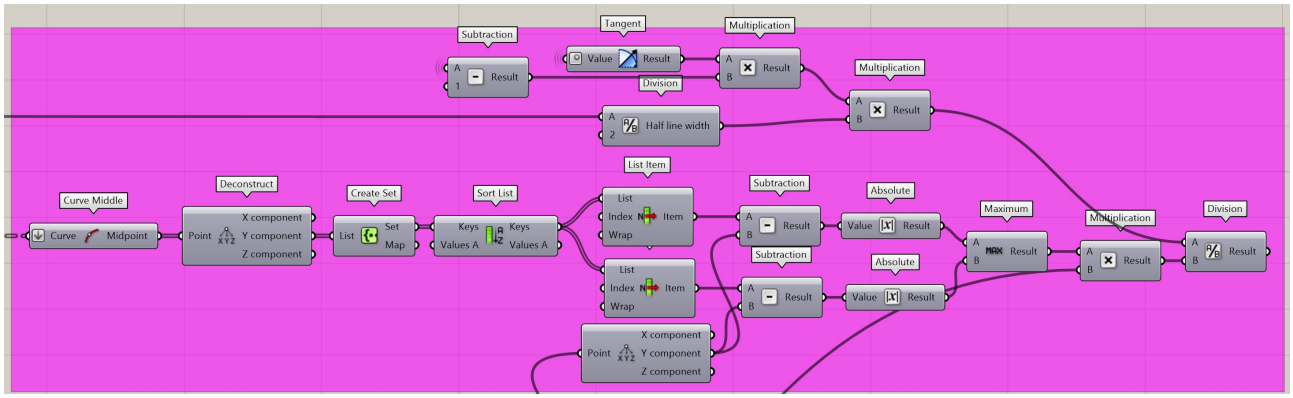


Figure 225: Calculating the step size for inclination

The next step is to create a series containing all the scaling factors of the different core layers, which are determined by the layer thickness, and shown in Figure 226. As the scaling is performed in the y direction, the boundary lines at the top and bottom are also inclined, which is not desired. The boundary lines should remain perpendicular to the glass; therefore, these lines do not scale according to the scaling factor. In order to solve this problem, the boundary lines at the top and bottom have to be separated from the other lines. Additionally, the vertical lines attached to the boundary lines at the top and bottom have to be separated as well, because scaling them in a similar manner as the other lines would detach them from the boundary lines. These before-mentioned lines are dispatched with the use of their midpoint y-coordinate. This process is performed in the components shown in Figure 227.

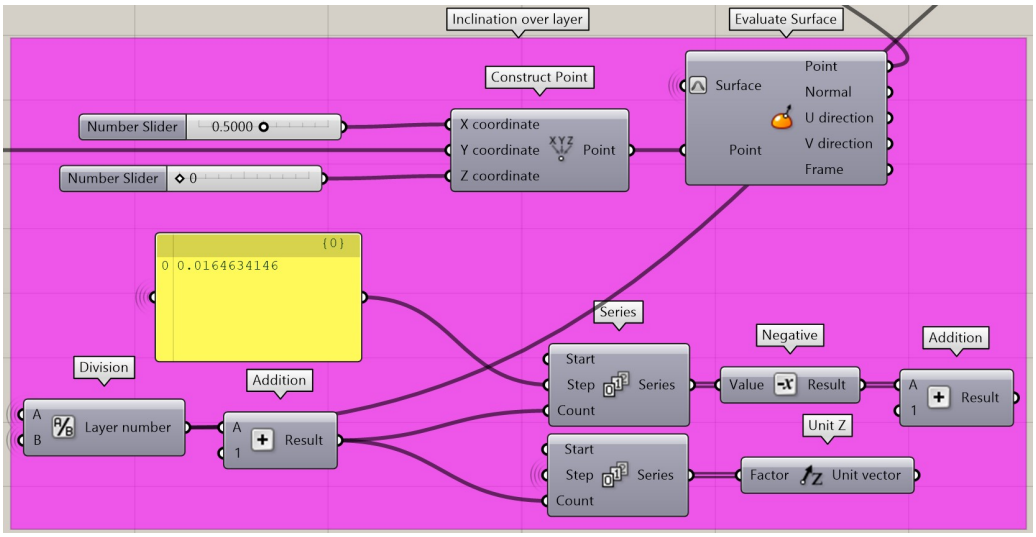


Figure 226: Determining the scaling factors.

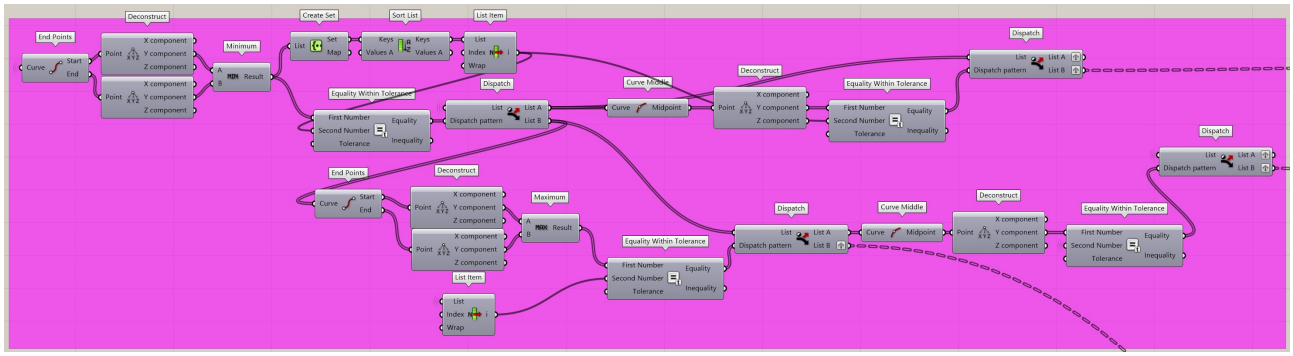


Figure 227: Separation of lines.

The following step is to determine the scaling factor of the vertical lines attached to the boundary lines, as these lines should remain connected on both sides. While the boundary lines have unaltered x and y coordinates, the attached diagonal lines are scaled with the previously determined scaling factor. The length adjustment of the lines at different z-coordinates is calculated in Figure 228.

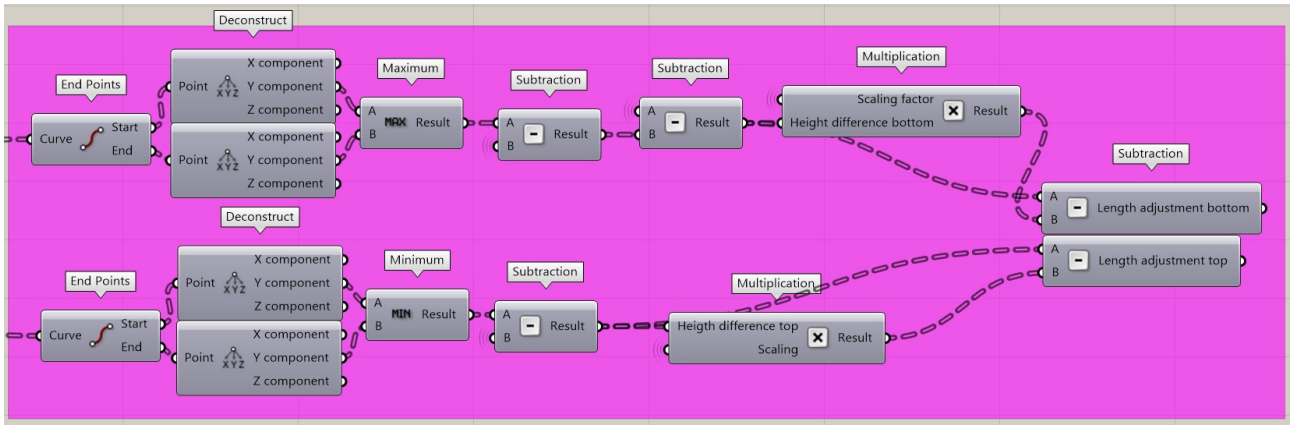


Figure 228: Length adjustment of vertical lines attached to boundary lines at the top and bottom.

The next step is to extend these lines and move them to the right position, as well as move these lines and the boundary lines in the z-direction. Moreover, the lines that can be scaled according to the scaling factor are scaled and moved in the z direction. Furthermore, from all these lines, the lines at a z-coordinate of zero and at core thickness are selected, and a surface is created between them; see Figure 229.

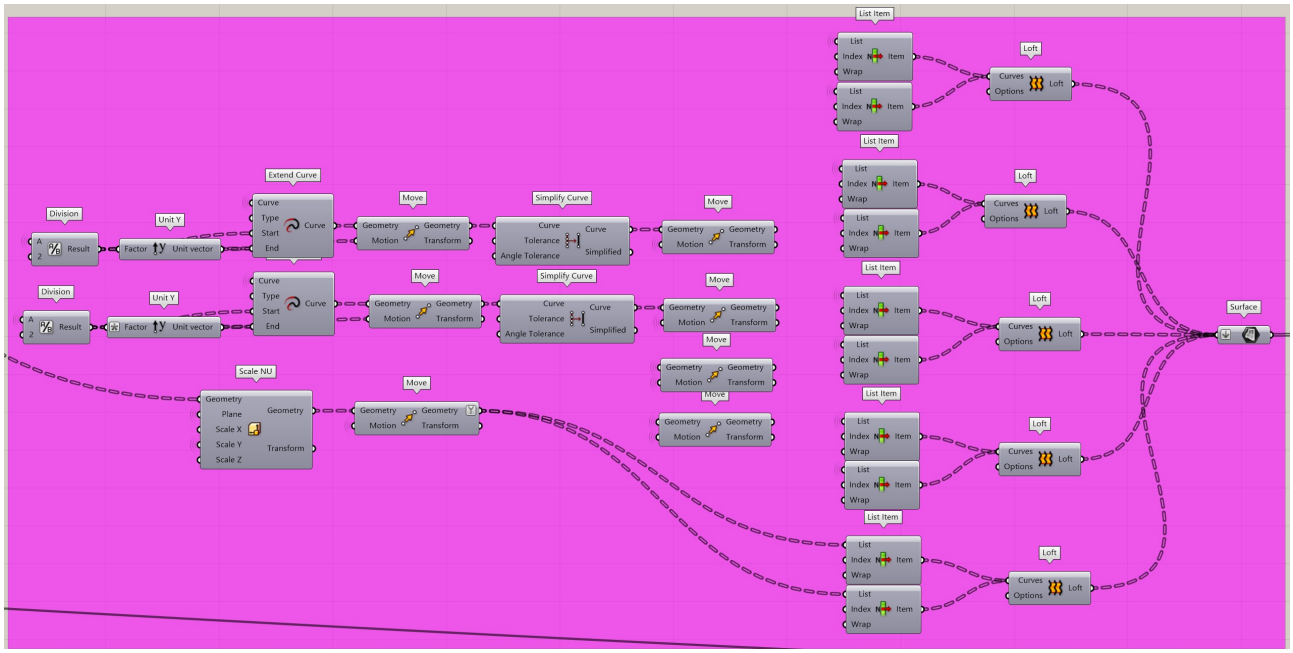


Figure 229: Scaling and moving all the lines and creating a surface.

The final step in creating a 3D geometry of the core is to add line width at the boundary lines, which is done by offsetting the boundary curves. Next, a surface is created at the minimum and maximum z coordinates in order to enclose the surfaces created in the previous step. Currently, all boundary surfaces of the geometry are created, and these surfaces can be joined to create a volumetric model of the core geometry. These final steps are shown in Figure 230 and the core geometry in Figure 231.

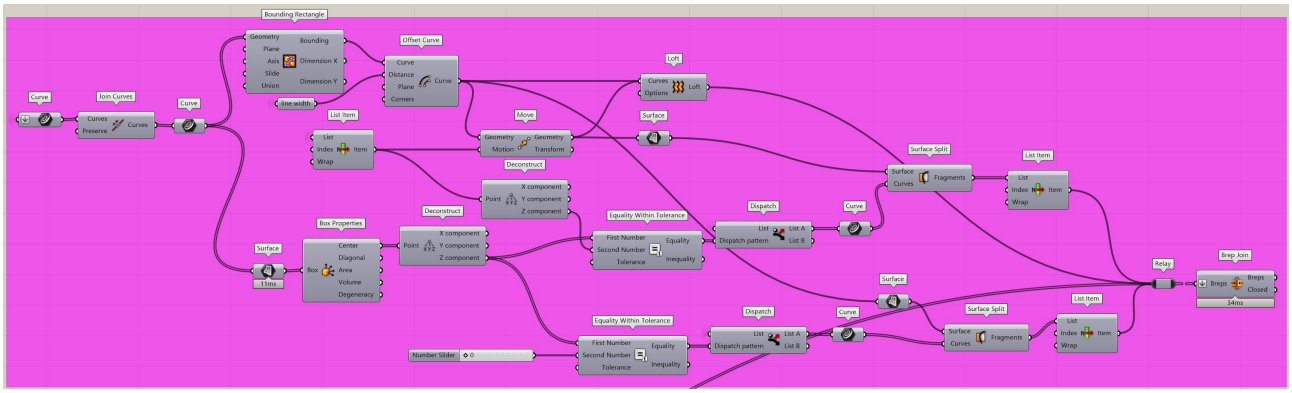


Figure 230: Creation of the 3D model of the core geometry.

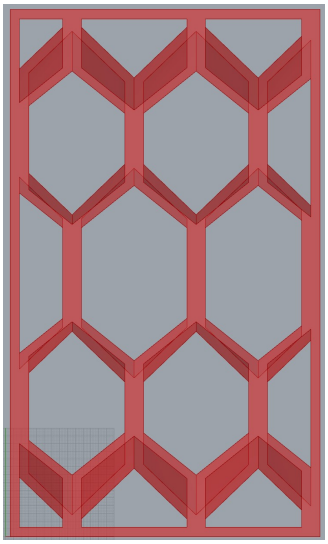


Figure 231: Final core geometry

These steps, shown above, are used in cases where the orientation is set to zero. In the situation with an orientation set to one, the same steps can be applied; however, some small changes are required; see Figures 232, 233, and 234. These differences are the result of the fact that scaling occurs in the other direction, which means that the core lines attached to the bottom and top are diagonal lines instead of vertical lines. Therefore, as the lines are no longer perpendicular, the connection process to attach these lines to the boundaries is a bit more complicated. The start and end points of these lines are gathered first. Then, these points are grouped into those that are on the boundaries and those that are not; see Figure 235. Subsequently, the points on the boundary are only moved in the z-direction (thickness), while the other points are also moved in the direction of the scaling. After this movement, the lines between these points are recreated. These steps are visible in Figure 236. This figure also shows the movement and scaling of all the other lines. The final step is to create the 3D core geometry, which is shown in Figure 237.

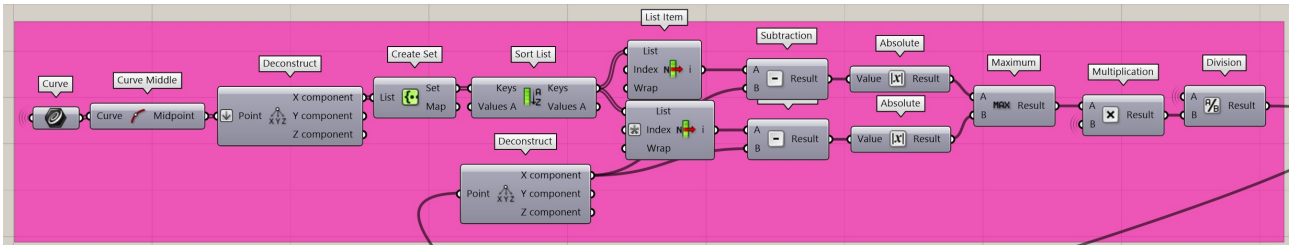


Figure 232: Calculation of step size for inclination if rotated pattern

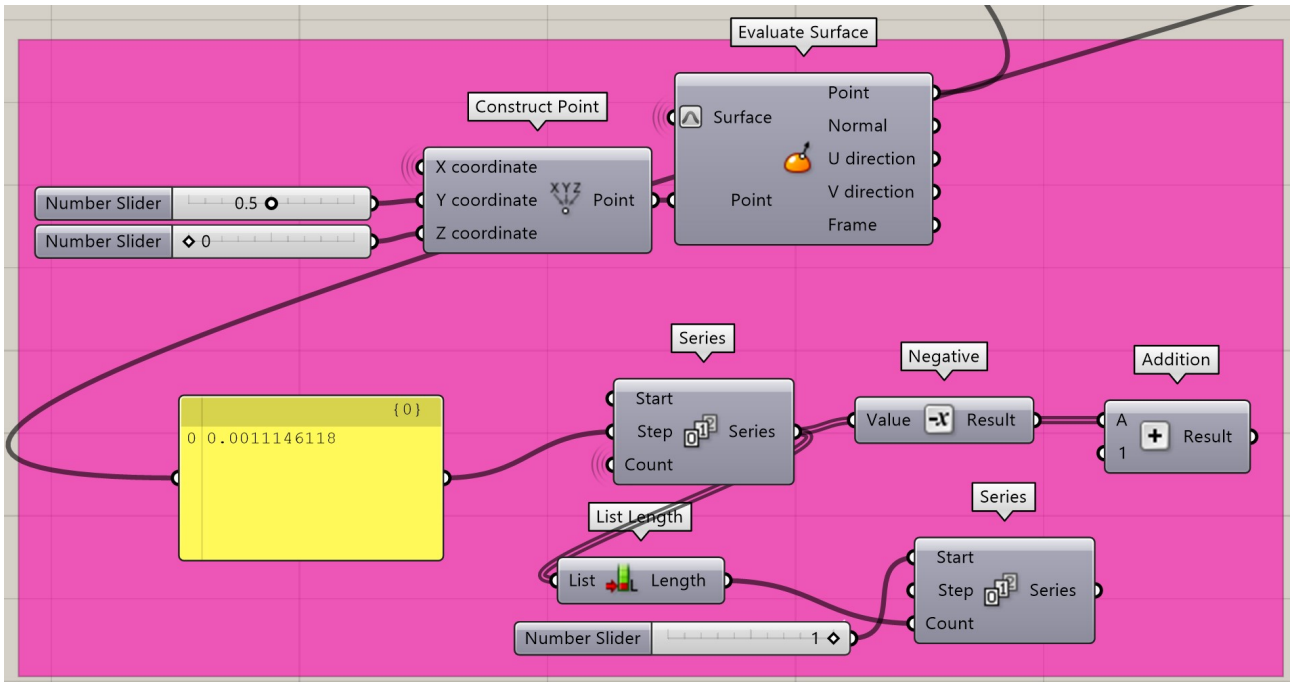


Figure 233: Determination of scaling factors if rotated pattern

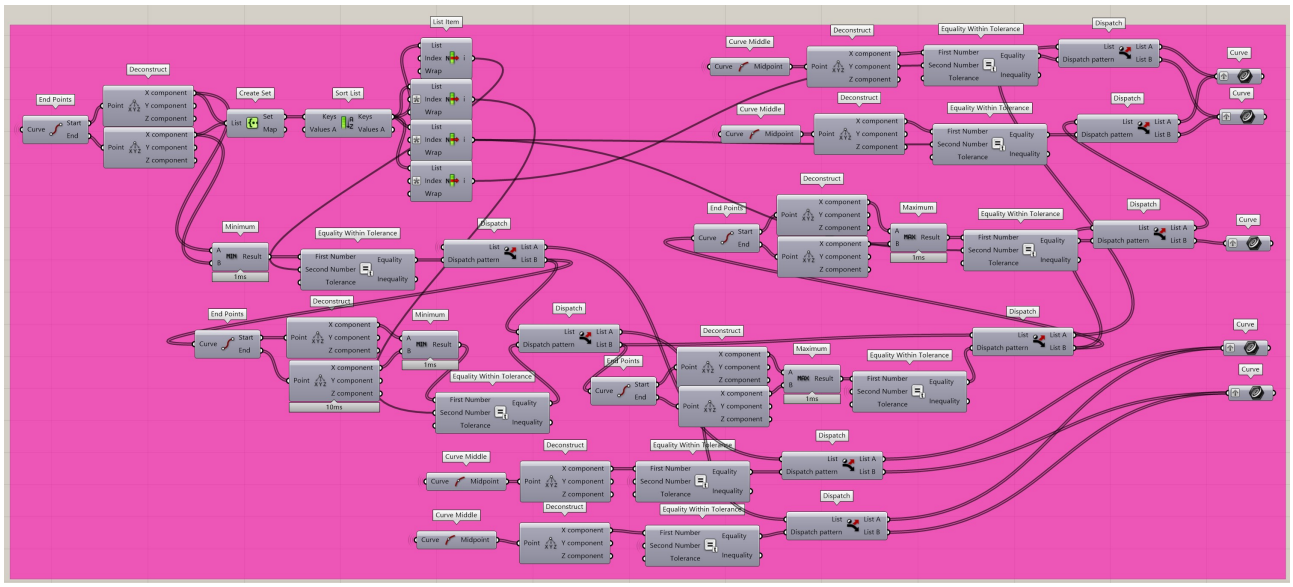


Figure 234: Separation of lines if rotated pattern

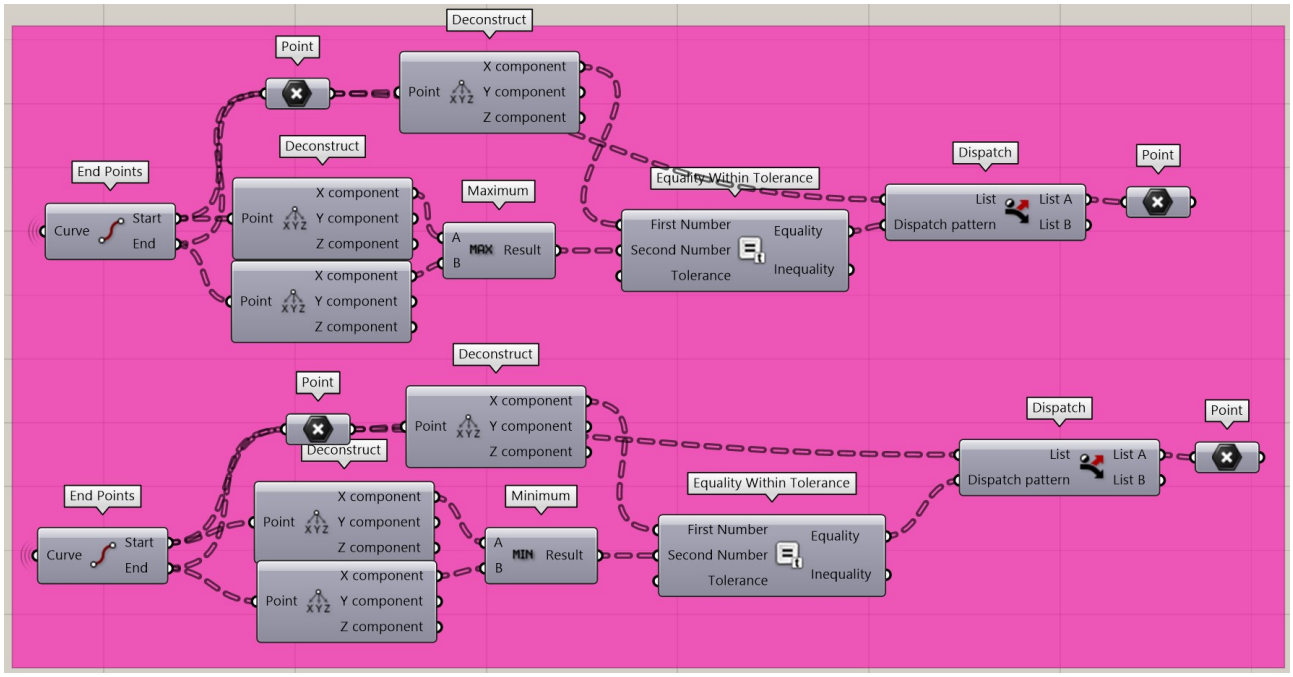


Figure 235: Determination of endpoints lines attached to boundaries if rotated pattern

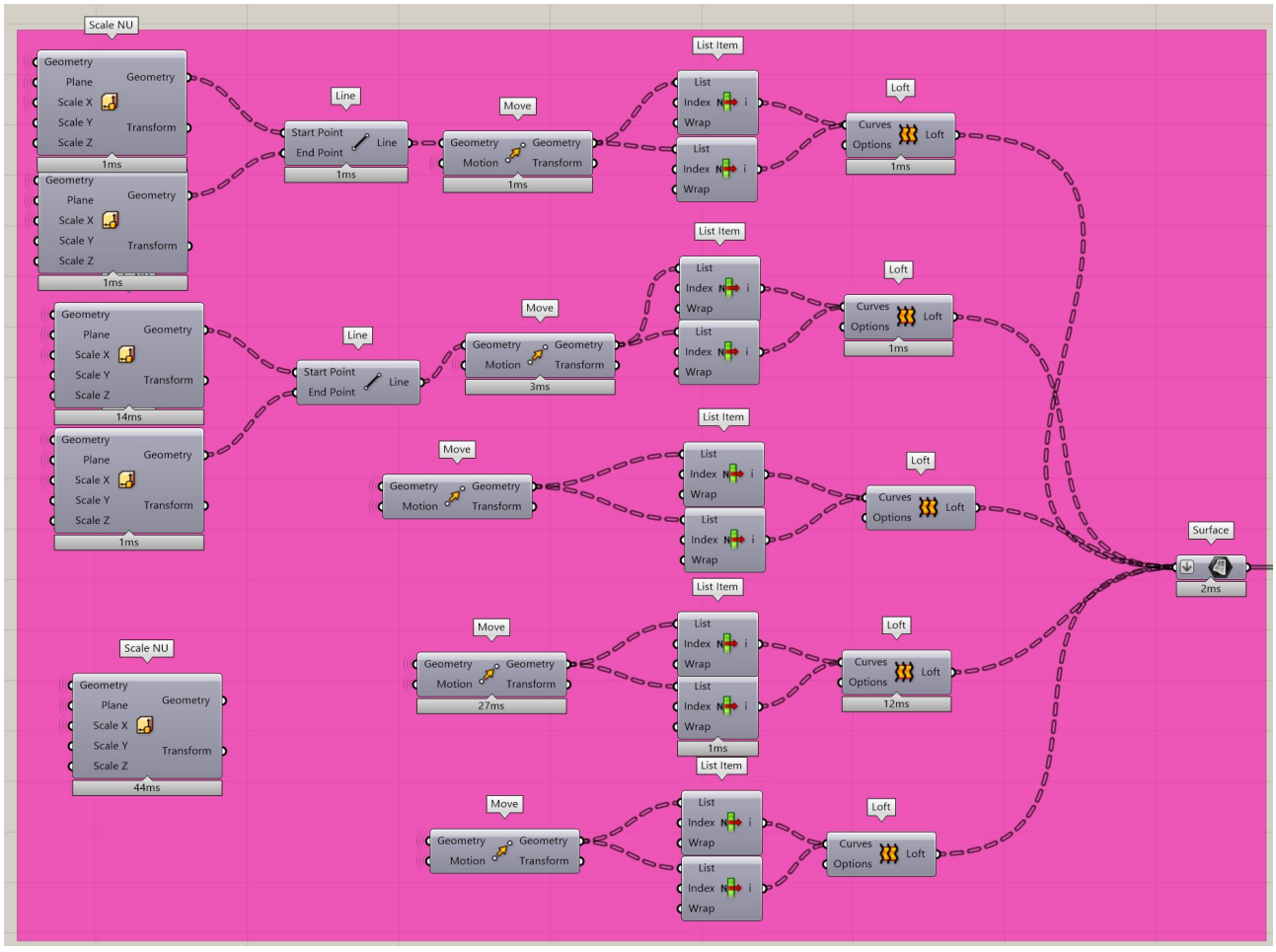


Figure 236: Scaling and moving all the lines and creating a surface if rotated pattern

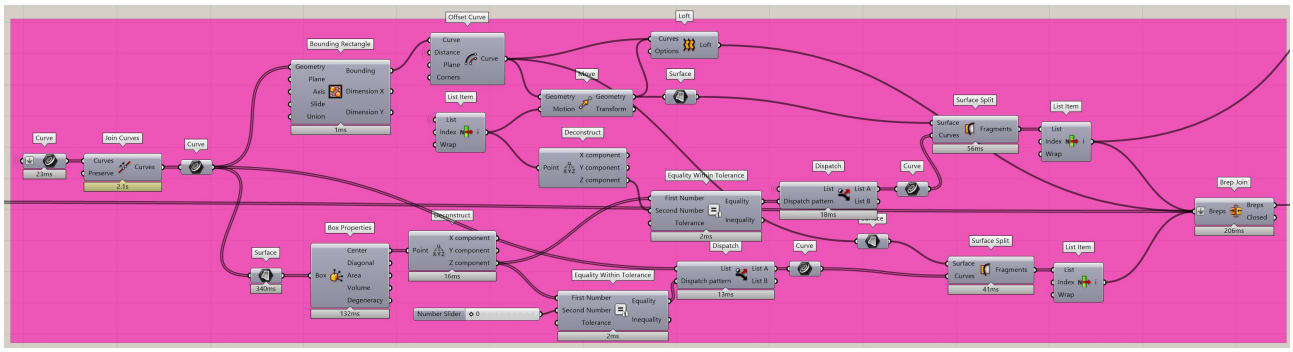


Figure 237: Creation of the 3D model of the core geometry if rotated pattern

Finally, as the core geometry is determined for both orientations of scaling, the associated scaling orientation with the orientation of the core geometry, the geometrical input, has to be chosen. This is shown in Figure 238. The 'z' input refers to the orientation of the panel as the geometrical input. The upper component has the 2D surface as an output, and the lower component has the 3D volumetric model. The output of the upper component is later required in the insulation criterion.

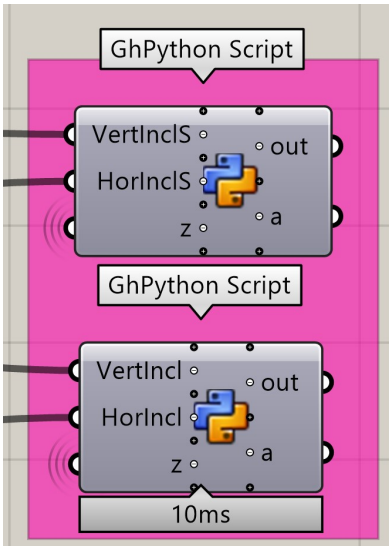


Figure 238: Choosing the correct scaled pattern

Optimisation process

Transparency criterion

The transparency criterion is related to the ability to see through the facade element perpendicularly. This area is calculated with a grasshopper component called 'mesh shadow', which calculates the shadow outlines of a 3D model. Thereafter, a surface is created, and the area is calculated. This process is shown in Figure 239.

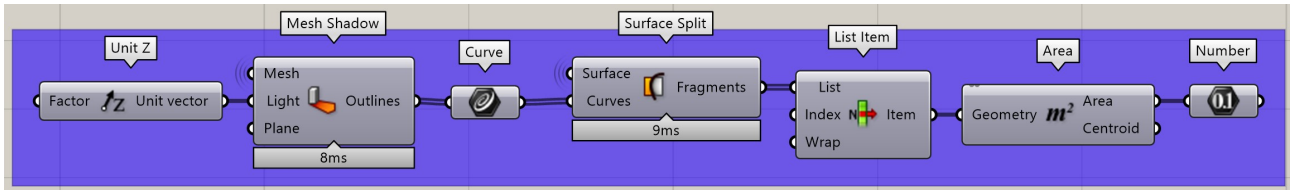


Figure 239: Transparency criterion.

Insulation criterion

The insulation criterion is based on the equation in a prior chapter and is shown in Figure 240. This criterion is based on some geometrical properties, such as the panel thickness, the area of the core attached to the glass panels, the line width, as well as some material properties of the core and glass panels.

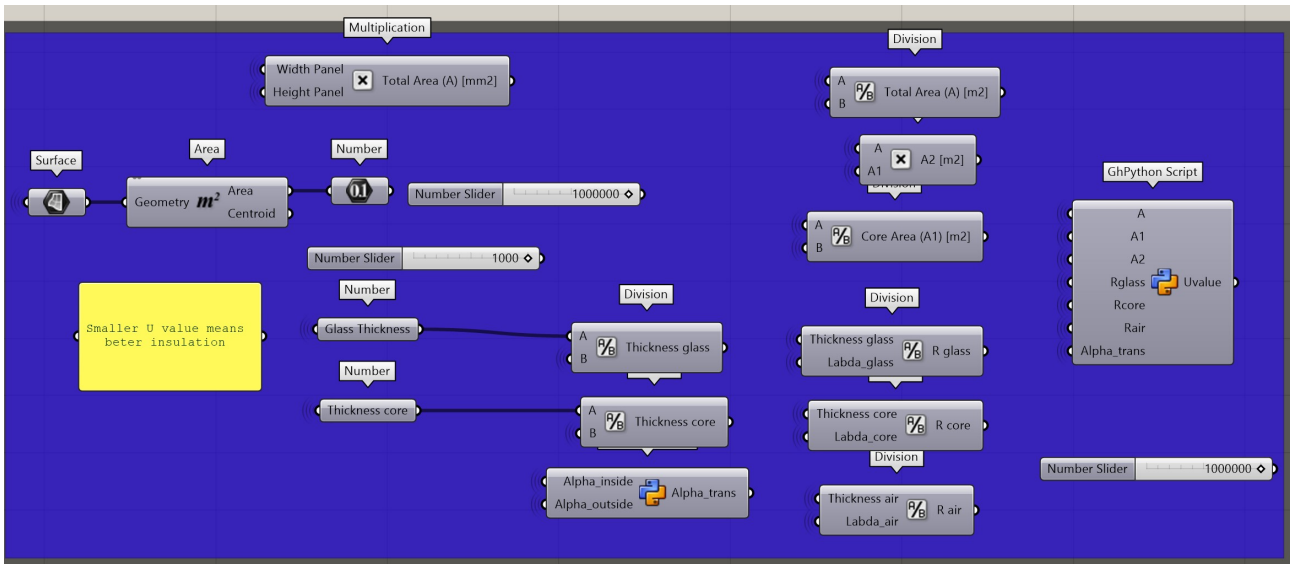


Figure 240: Insulation criterion.

Daylight control criterion

The third and final criterion is daylight control, which is based on the direct and indirect sunlight going through the facade panel. This criterion takes advantage of the Ladybug and Honeybee components in Grasshopper. The first step is to create the daylight scenarios required for both solar radiation calculations. The next steps are divided into two groups, where one group determines the direct sunlight and the other group determines the direct and indirect sunlight. Moreover, the creation of the daylight scenario is slightly different per group; therefore, this creation is discussed per group.

Direct sunlight

The first step in scenario creation is to re-orient the core geometry because of the orientation of the incoming sunlight. This process is shown in Figure 241. Then, a box structure behind the panel is added to simulate a situation where the sunlight can only reach inside the room or the area through the facade panel. This is shown in Figure 242. As an illustration, the situation is shown in Figure 243. Then, a plate-like structure is created within the box structure to catch the incoming sunlight; see Figure 244. This structure is placed one cm behind the panel. The small gap exists because the case that accounts for only direct sunlight and the case that accounts for direct and indirect sunlight should represent the same situation. In cases that include diffused light, a small gap is necessary to allow for diffused sunlight to reach areas that are not reached by direct sunlight.

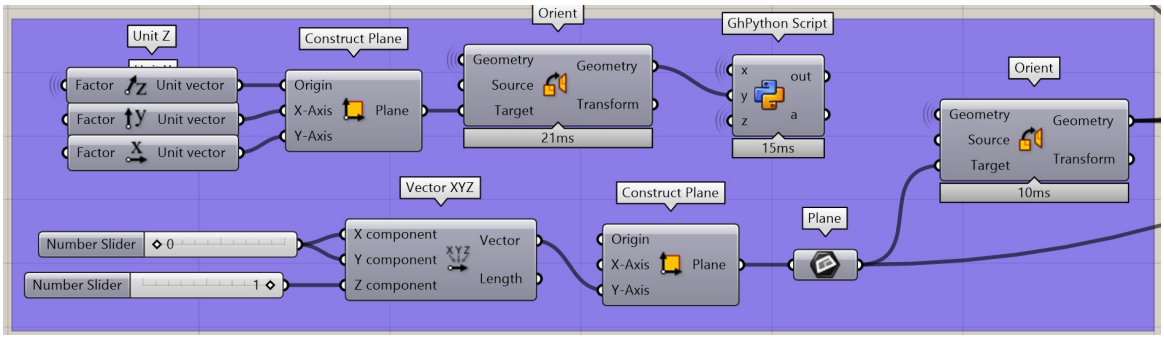


Figure 241: Changing the orientation of the panel.

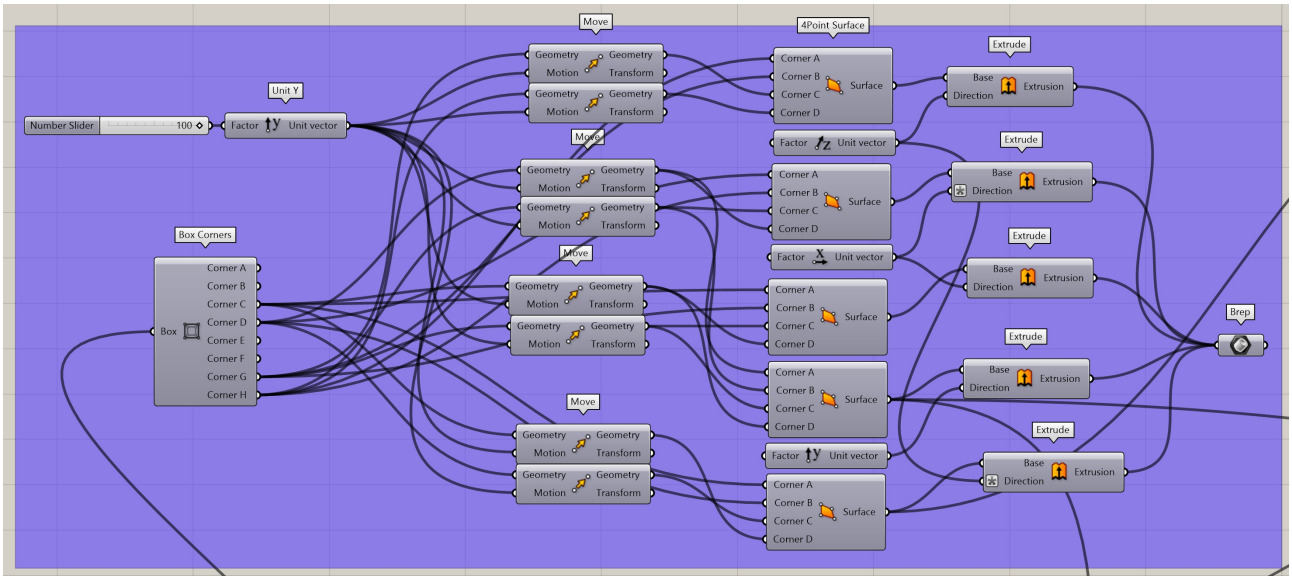


Figure 242: Creation box-like structure behind panel.

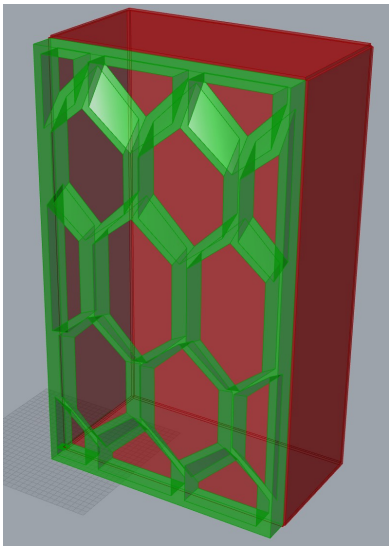


Figure 243: Current situation, where the panel is green and the box-like structure displayed in red.

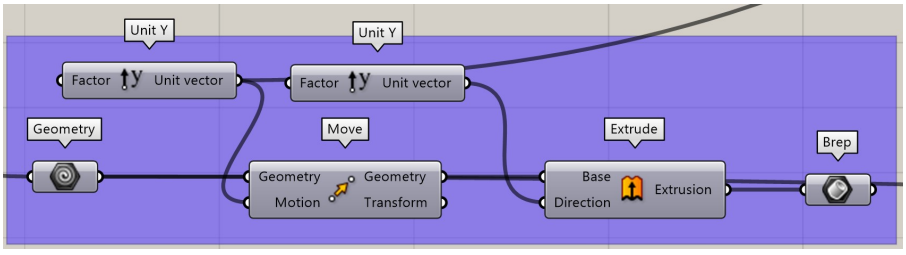


Figure 244: Creation plate-like structure to capture the sunlight.

The next step is to import a file that stores the solar information, such as the position of the sun during different timestamps and the incoming solar radiation during these timestamps. This information is stored in an epw file, and in this case, the solar information of the city of Amsterdam is chosen. The sun position is shown in Figure 245, where the panel is oriented to the south. The orientation of the panel can be changed to reassemble the real situation as close as possible. Moreover, it is possible to add the surrounding area of the location in which the panel will be used, for example, the high-rise buildings in the vicinity.

Additionally, Figure 246, shows the different steps required to calculate the direct sunlight. The sunlight is calculated on both the winter and summer solstices because it represents the extremes in the sun's position. Then, the results of the 'LB Incident Radiation' component are summed and multiplied by a factor, which is the total area of the plate-like structure that collects the incoming sunlight and the number of areas that register the incoming light. This procedure is followed to remove the dependence of the final number on the level of discritisation. This is required to compare this result with the results from direct and indirect sunlight. The grid size is an important parameter that affects the accuracy of the solution. This is studied intensively in the grid size sensitivity study in Chapter 14.

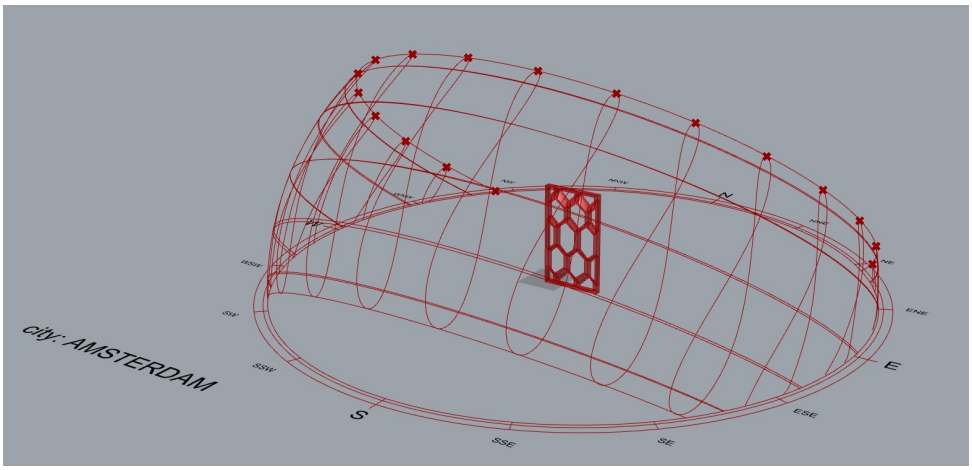


Figure 245: Illustration sun path and facade panel.

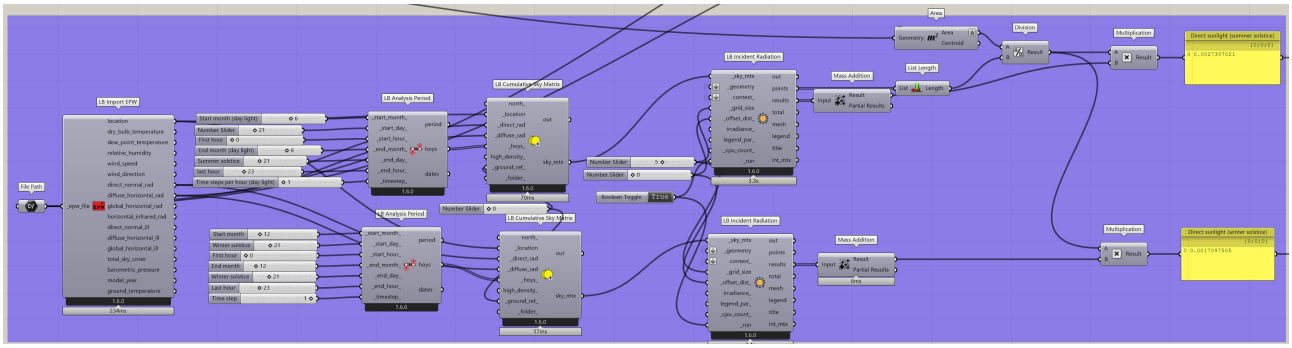


Figure 246: Calculation direct sunlight during winter and summer solstice.

Direct and indirect sunlight

The direct and indirect sunlight radiation is calculated with Honeybee components, which require a slightly different set-up of the model. The general idea remains the same, where the sunlight comes through the facade panel and into an area enclosed by a box-like structure and captured by a surface one cm behind the facade panel. The model is created with the 'HB model' component, and the 'HB Sensor Grid' component creates the surface that registers the incoming sunlight as well as the level of discritisation. This is shown in Figure 247.

The next step is to calculate the total sunlight at both the winter and summer solstices, which also uses the same epw file. This is shown in Figure 248. These components require hours of the year (hoys) to define the period, which is done with the 'GhPython Script' component. The final number in the yellow boxes is again multiplied by the factor equal to the area divided by the number of areas as a result of the discritisation. Moreover, this calculation has a lot of changeable parameters; however, aside from the grid size, all the predefined values for the parameters are used.

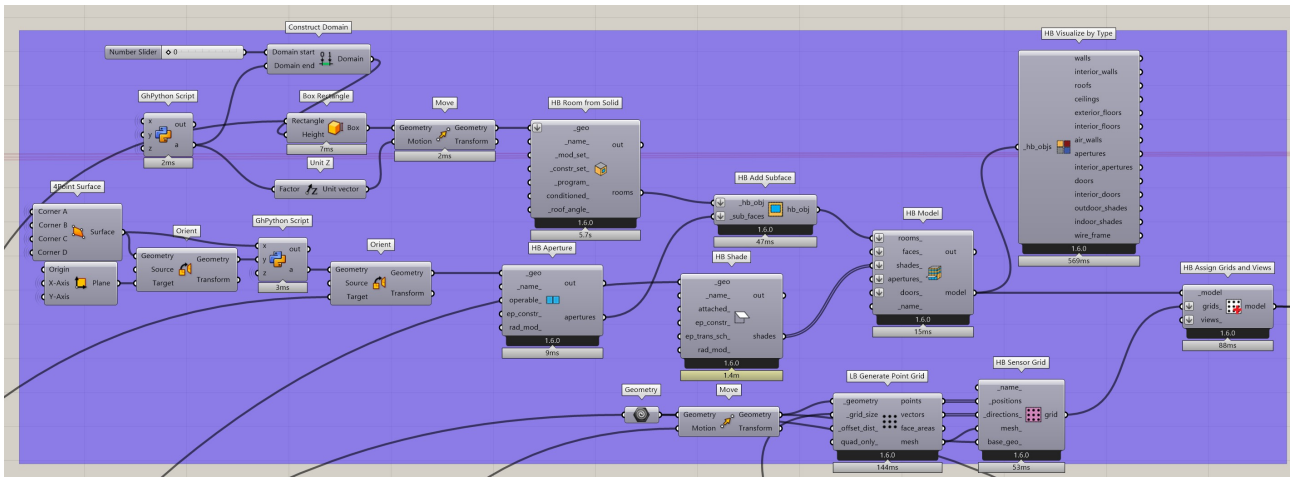


Figure 247: Creation of the model for the direct and indirect sunlight.

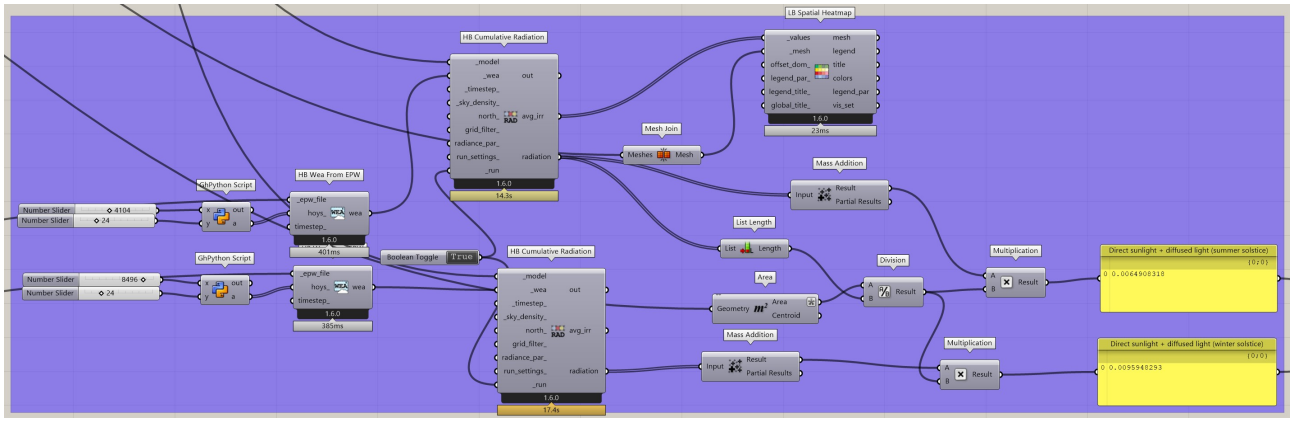


Figure 248: Calculation incoming solar radiation.

Daylight control measure

The final step is to calculate the daylight control measure used in the optimisation process. An optimal facade panel allows for a sufficient level of light to enter the area while limiting direct sunlight. Furthermore, during the winter, in climates similar to the Netherlands, the sun provides additional warmth in the area. Thus, it is desirable to have enough incoming sunlight, direct and indirect, during the winter. Therefore, an optimal panel allows for ample indirect sunlight during the summer and sufficient direct and indirect sunlight in the winter. This is shown in Figure 249.

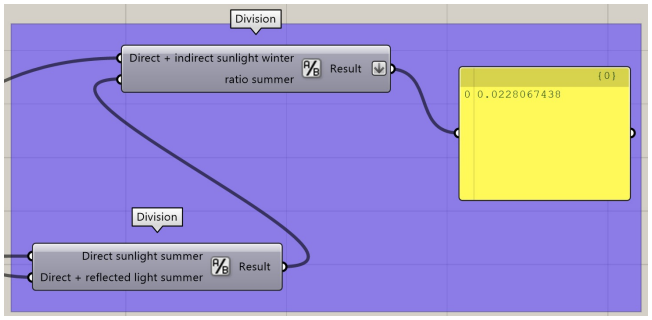


Figure 249: Defining optimisation value for day light control criterion.

Structural constraint

The next thing to be considered is the structural constraint, which first requires a panel reconfiguration necessary for the shell elements. Followed by an explanation of the structural model, the Karamba3D model.

Panel reconfiguration

Before all the criteria can be defined, the center lines of the geometry need to be found, which is primarily important for the structural criterion. The center lines represent the line that is in the center of the line. This center can be equal to the path of the print head, in case there is a single line width, or the middle of a double line-width line. First, the center lines of the vertical lines are generated. This step is facilitated by grouping the vertical lines into the outer-most lines, the second outermost lines, and the other vertical lines. The second-outer most lines are the boundary lines with a double thickness. This division is based on the required movement in the coming steps. This division is performed in the components shown in Figure 250.

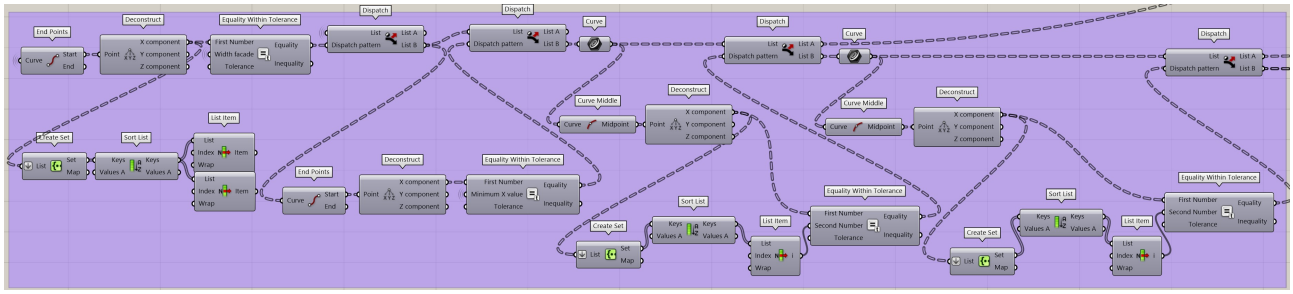


Figure 250: Grouping the lines prior to generating the center lines

Next, the outer-most lines and second outer-most lines are moved to represent the center lines, which are displayed in Figure 251. Beware that the second outermost lines are the parts that have a double line width and therefore have two center lines. Thus, some lines are moved by half a line width, while others are moved by one and a half times the line width. The structural model uses a continuous boundary line, which is partly composed of lines that are moved by one and a half times the line width in cases of double line width. The other vertical lines also have a double line width, and in order to know the direction of the movement, the lines are divided into two groups, representing the left and right sides of the total printed line. Then, the total center line is created by moving the left lines in the right direction with the line width. This is shown in Figure 252.

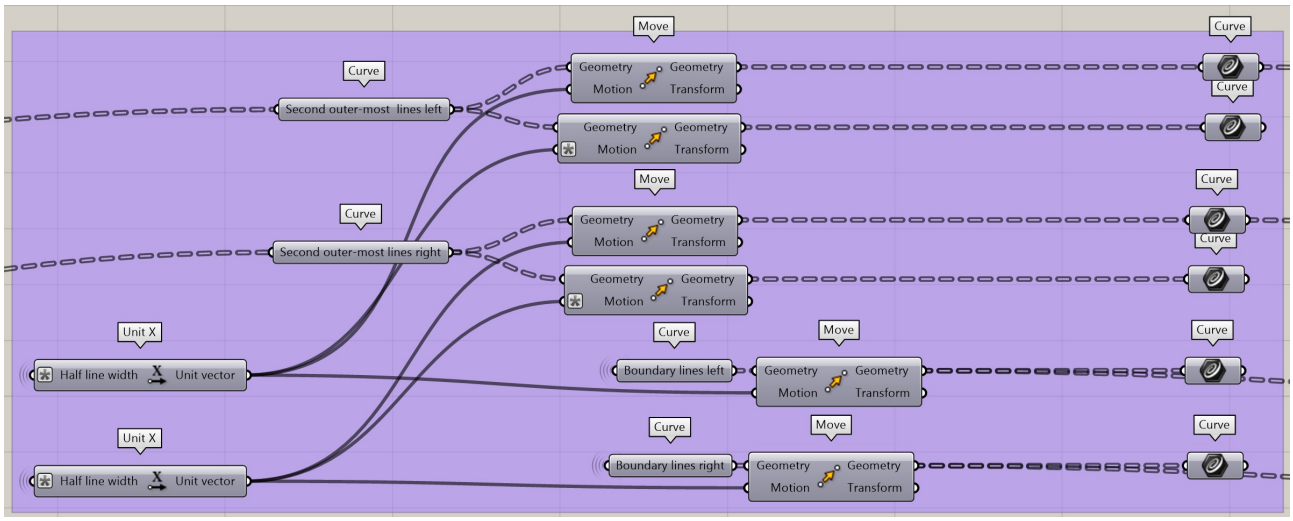


Figure 251: Generating the center lines of the outer-most and second outer-most vertical lines.

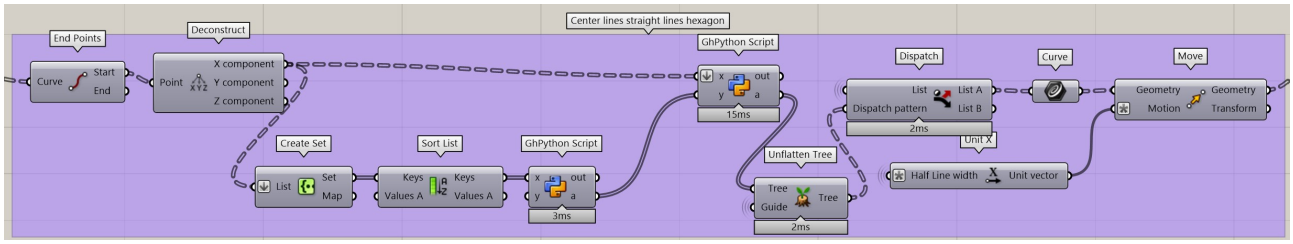


Figure 252: Generating the center lines of the other vertical lines.

After obtaining the center lines of the vertical lines, the diagonal center lines have to be created. All diagonal lines only have a width equal to a single line width. The center lines are created by moving the lines as done before; however, the edge lines must either be moved to the right or to the left. Therefore, the lines are divided into two groups based on this distinction. The lines are sorted by their x-coordinate, and the lines are moved to the left or to the right, alternatively. This is shown in Figure 253.

The next challenge is that the lines should be moved in a perpendicular direction, which requires an additional separation of the diagonal lines into groups with the same orientation. The end point with the largest y-coordinate and its x-coordinate are determined for the lines that have to be moved to the left. Then, the absolute minimum x-coordinate of the two end points per line is calculated. Subsequently, if this minimum x-coordinate does not coincide with the x-coordinate of the end point with the largest y-coordinate, the line will need to be moved to the left (negative x-direction) and upwards (positive y-direction). Additionally, for the other group of diagonal lines that have to be moved to the right, the end point with the smallest y-coordinate and its x-coordinate are determined. Next, if this x-coordinate does not coincide with the absolute minimum x-coordinate of either end point, the line will need to be moved to the right (positive x-direction) and upwards (positive y-direction). This distinctive procedure for both groups is applied to obtain a single line per diagonal, which has to be moved; as shown in Figure 254. Thereafter, as shown in Figure 255, the vector perpendicular to the diagonal line is determined, and the diagonal lines are moved with half a line width along this vector.

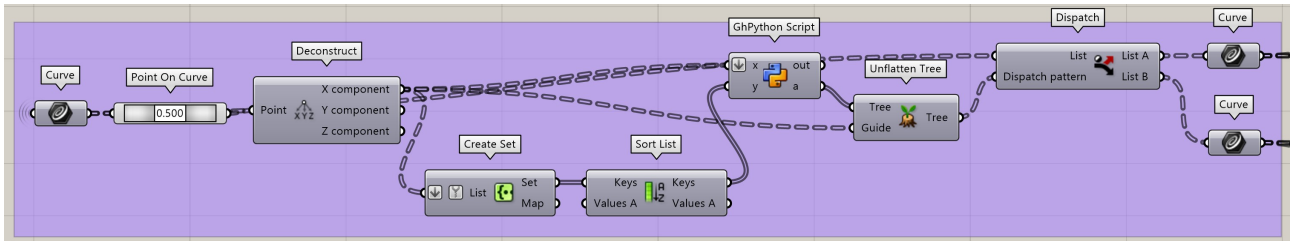


Figure 253: Dividing diagonal lines in two groups based on direction of movement.

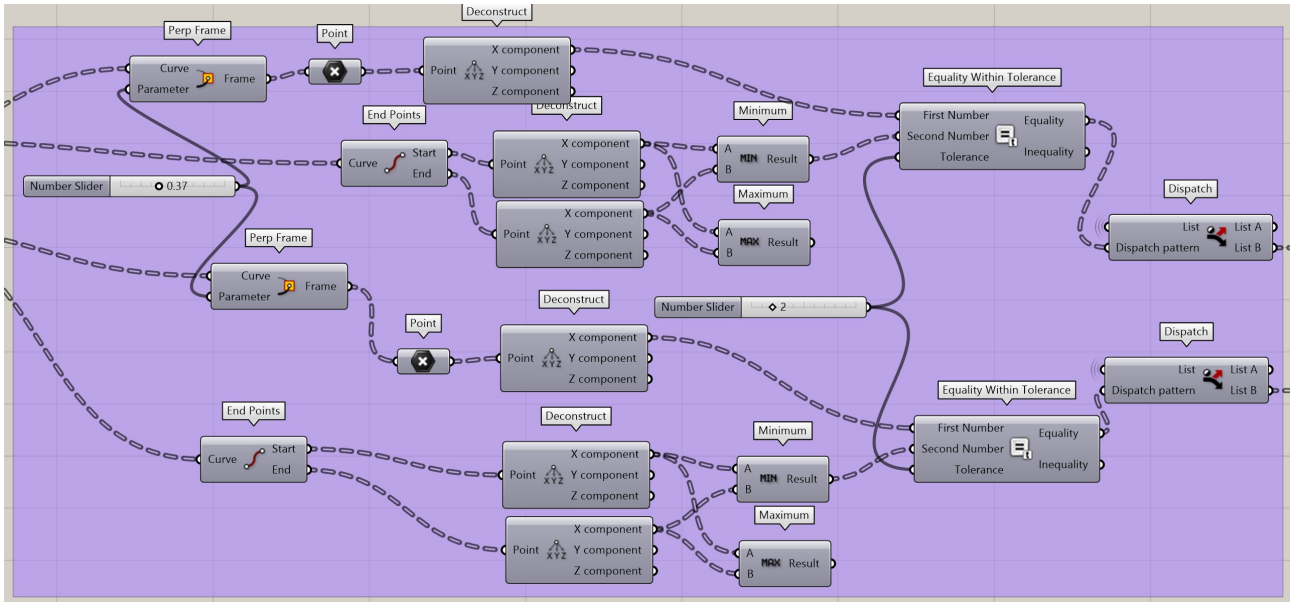


Figure 254: Separating the diagonal lines based on their orientation.

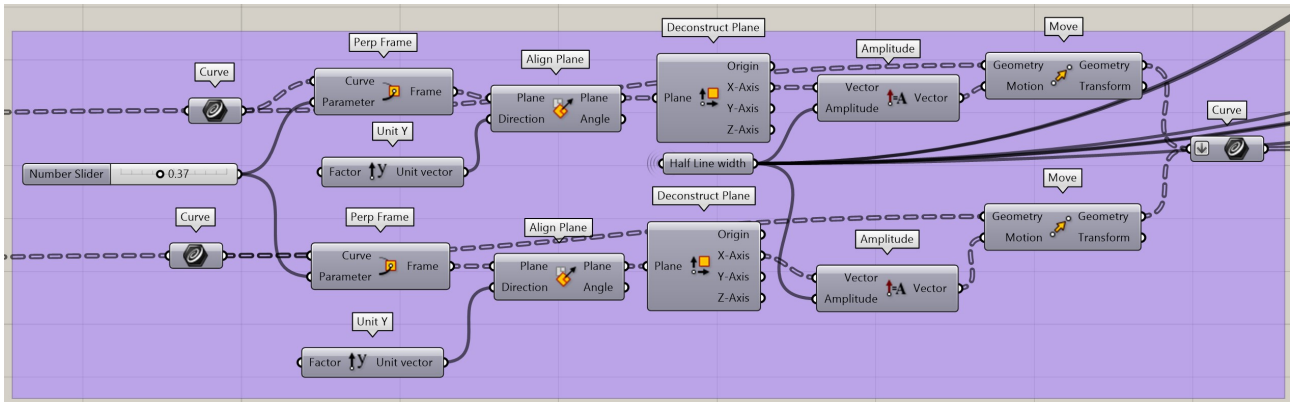


Figure 255: Generating the center lines of the diagonals.

The horizontal center lines are made by dividing the upper and lower horizontal lines and moving them with half a line width. This is displayed in Figure 256.

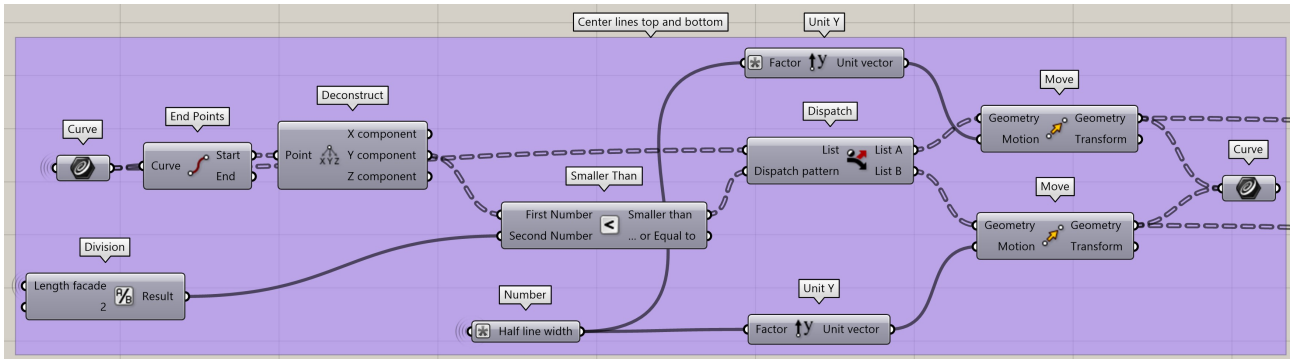


Figure 256: Generating the center lines of the horizontal lines.

Subsequently, the diagonal lines at the left and right boundaries have to be adjusted in order to connect them later on with the boundaries. This process is shown in Figures 257, 258, and 259.

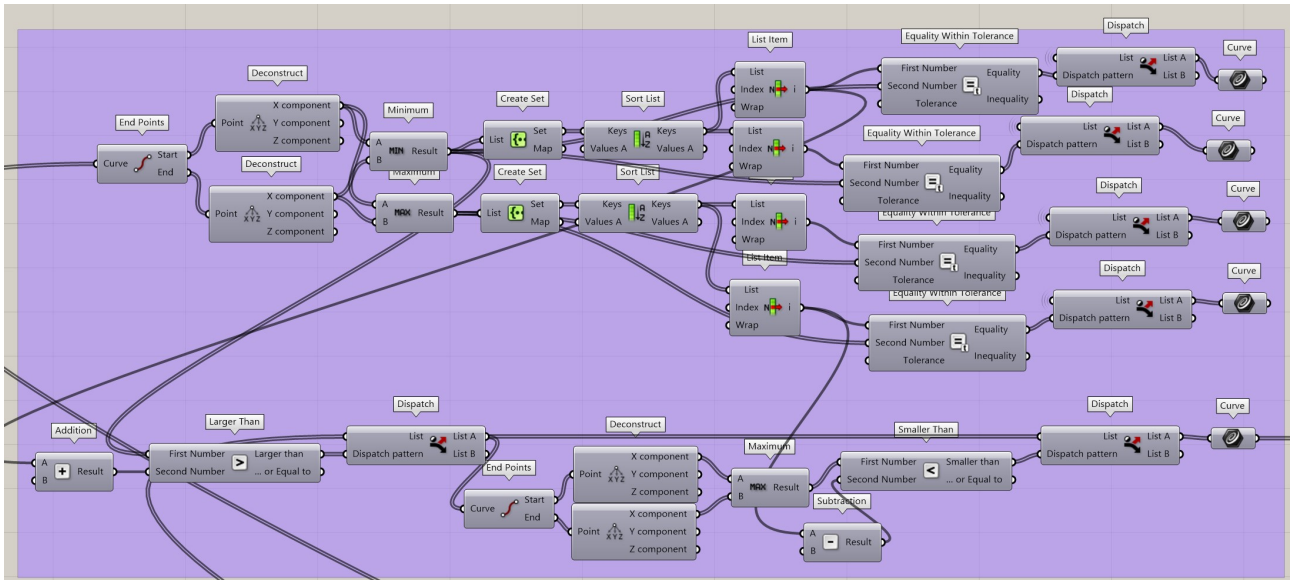


Figure 257: Grouping the diagonal lines at the left and right boundaries.

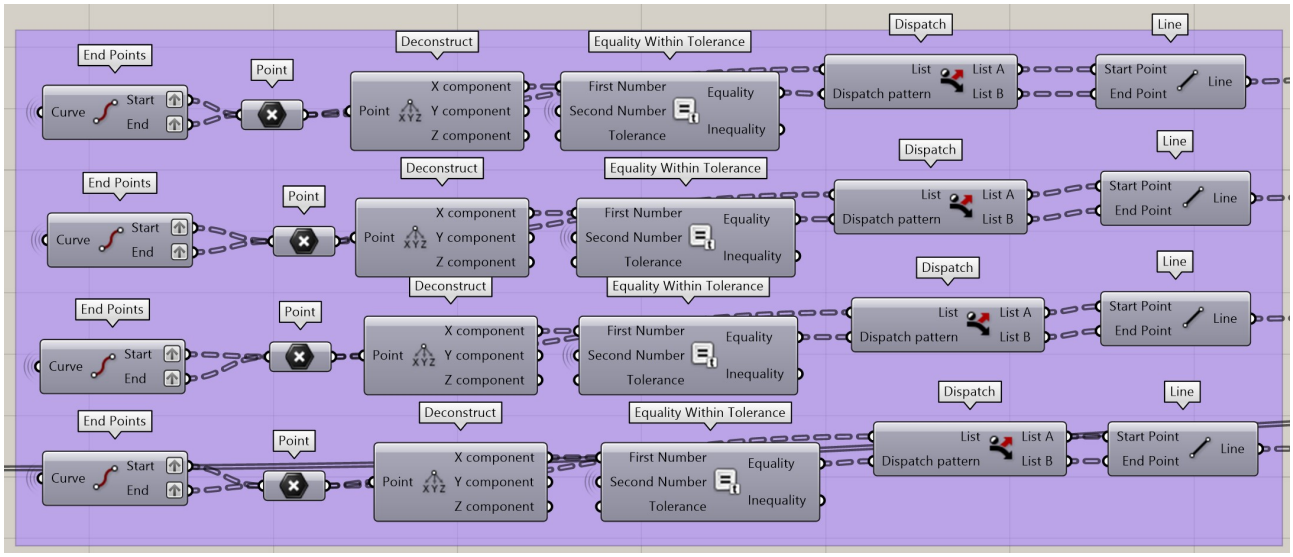


Figure 258: Determining the end points of the diagonal lines situated near the left and right boundaries.

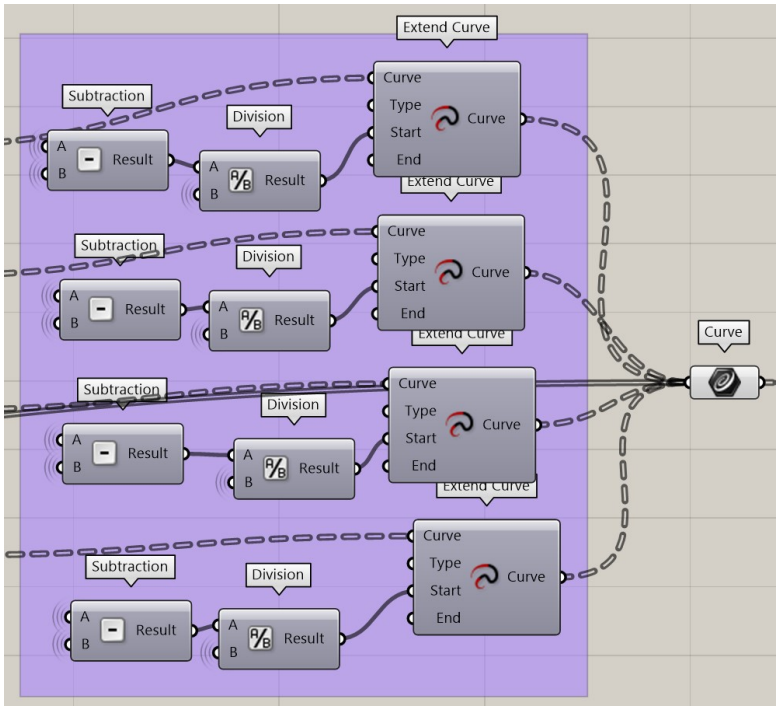


Figure 259: Length adjustments to the diagonal lines at the left and right boundaries.

The next step is to connect the center lines. First, the diagonal and vertical lines, without the boundary lines, are connected. Based on simple geometry, the required length for extending or shortening the diagonal and vertical lines can be assessed. Initially, the second outer-most vertical lines require an extension of Δz_{out}, which is written down in Equation (57). This additional elongation is required because of the earlier movement of the vertical line to represent a double line width in certain boundary sections. Then, all the vertical lines are extended by Δz, as shown in Equation (55), which originated from the movement of the vertical and diagonal lines. These steps are performed in the components displayed in Figure 260.

Additionally, the diagonal and vertical lines are grouped together, and the lines are split based on the intersection points. Subsequently, the lines that are equal to Δd, shown in Equation (56), are removed. These lines represent the extra and unnecessarily section of some of the diagonal lines. Furthermore, some diagonal lines require an additional length of Δz to reach the intersection point with the vertical lines. Also, some of the diagonal lines attached to the second outer-most vertical lines have an extra unnecessarily section with a length of Δd_{out}, written down in Equation (58). This section is removed through the use of the line-line intersection

component and an equality condition. Finally, these lines are reconnected, and this second part of reconnecting the lines is visible in Figure 261.

$$\Delta z = 0.5 \cdot lw(sin(\beta) - tan(\theta) + cos(\beta) \cdot tan(\theta)) \tag{55}$$

$$\Delta d = \frac{lw - 0.5 \cdot lw \cdot cos(\beta)}{cos(\theta)} \tag{56}$$

$$\Delta z_{out} = tan(\theta) \cdot lw \tag{57}$$

$$\Delta d_{out} == \frac{lw - lw \cdot cos(\beta)}{cos(\theta)} \tag{58}$$

Where:

- Δz is the vertical distance to the intersection point from the vertical center line to the diagonal center line.
- Δd is the required addition or shorting of the diagonal lines relative to the intersection point.
- Δz_{out} is the required additional extension of the outer vertical lines.
- Δd_{out} is the required additional shortage of the outer diagonal lines.
- lw is the line width.
- β is the angle between the diagonal and the horizontal axis [degrees].
- θ is the angle between the diagonal and the vertical axis [degrees].

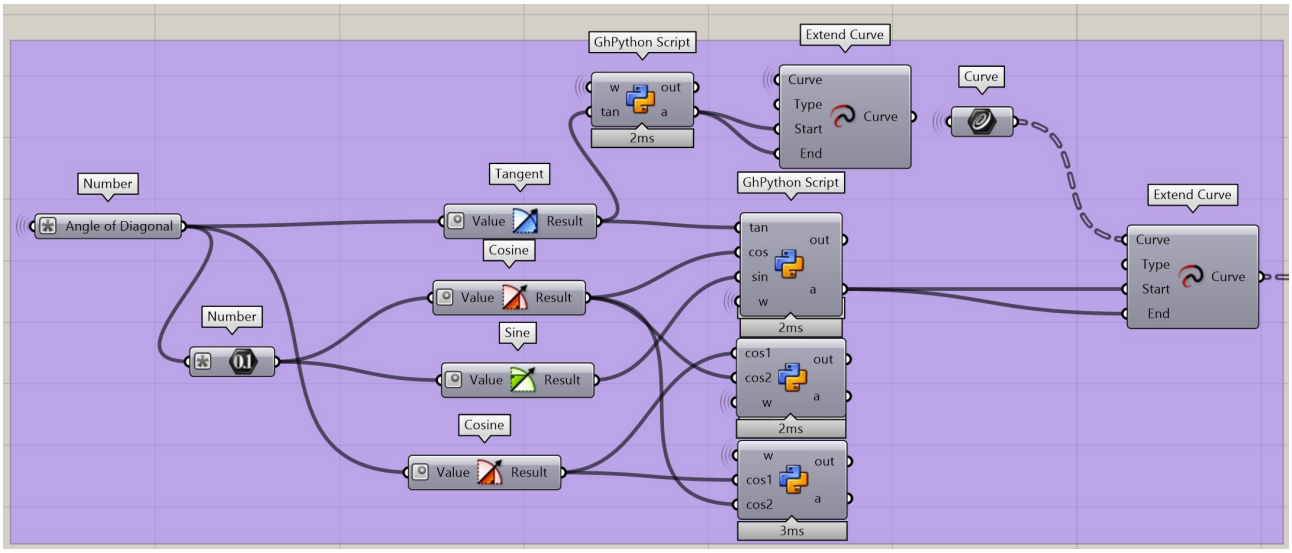


Figure 260: Extending the vertical lines prior to reconnecting them with the diagonal lines.

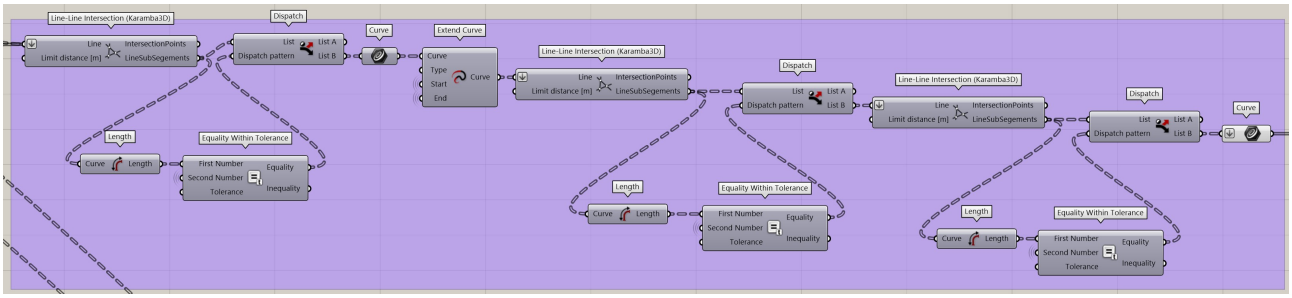


Figure 261: Dispatching, and shortening and extending vertical lines.

Furthermore, the boundary lines have to be connected as well. The boundary lines have to form a continuous line along the boundary. However, the boundary lines at the sides of the panel have a thickness of once or twice the line width. Therefore, the boundary lines are also grouped based on the line thickness. This process is shown in Figures 262 and 263. The next step is to adjust the line segments of the boundary to connect them to the inner hexagon center lines, shown in Figure 264. Subsequently, the left and right boundary lines connected to the upper and lower boundary are grouped in distinct sets, as shown in Figure 265.

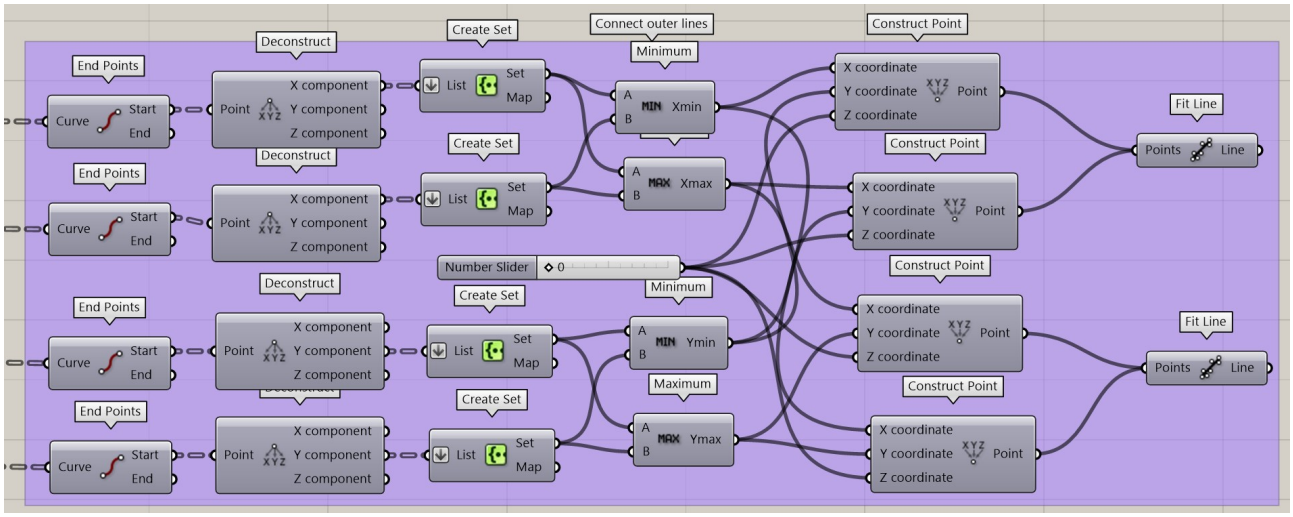


Figure 262: Determining the corner points and top and bottom boundary line.

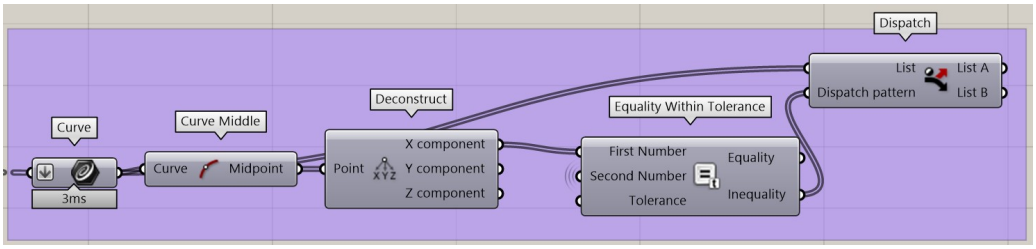


Figure 263: Grouping the lines with double line thickness.

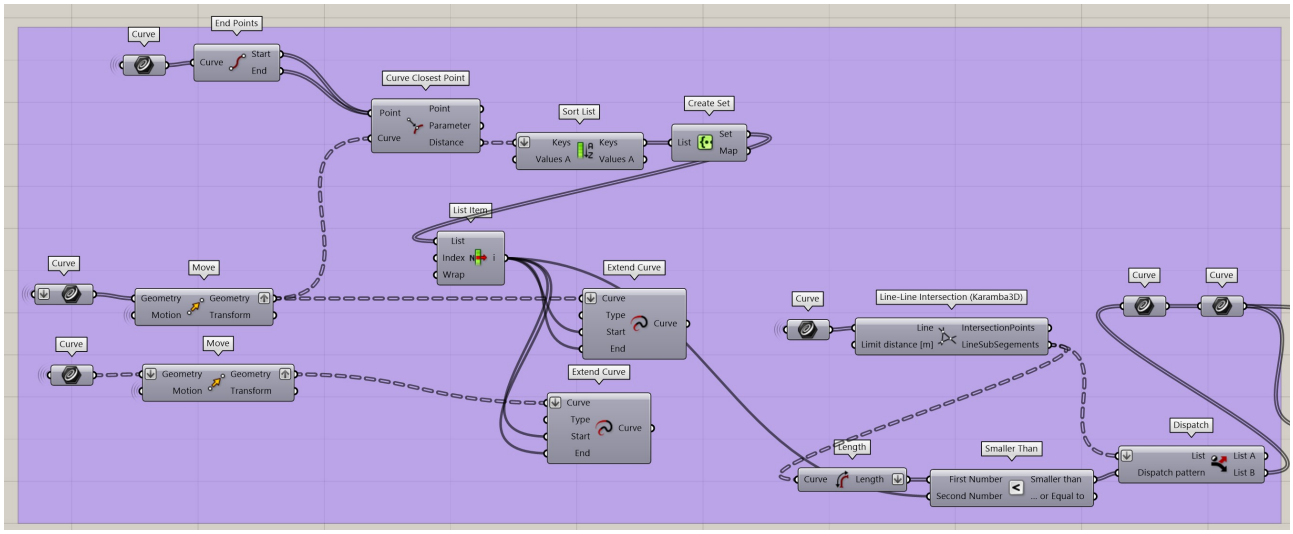


Figure 264: Line length adjustments boundary lines right and left side

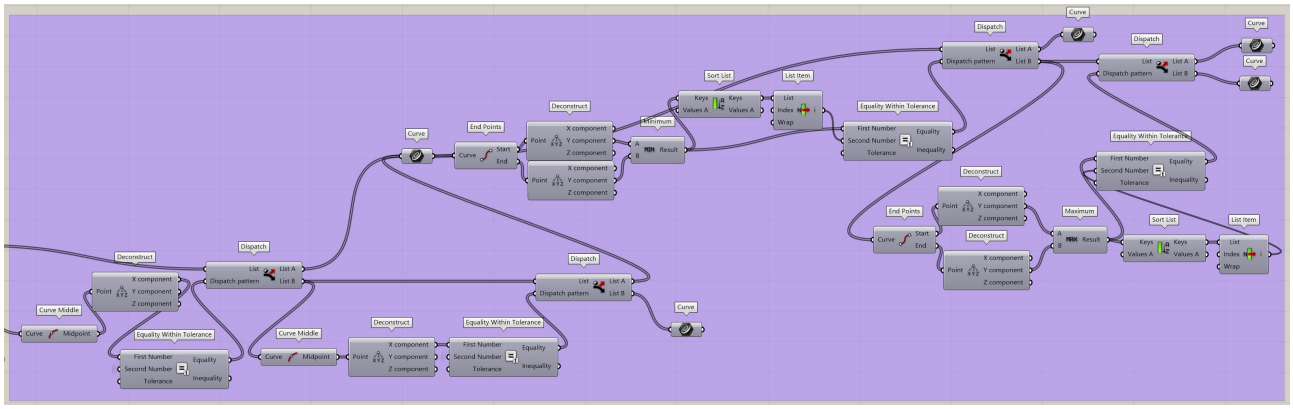


Figure 265: Grouping boundary lines based on inclination adjustments.

Finally, the vertical lines have to be attached to the boundary lines at the top and bottom. This is done by creating two groups that only consist of vertical lines that need to be either attached to the bottom boundary line or the top boundary line. This is already done in Figure 265. This separation is also done as a preparation step for the addition of the inclination to the center lines. The extension is calculated by determining the distance between the vertical lines and the boundary line. Then, half the extension is added to each end of the vertical line, followed by a movement, equal to half the extension, of the line in the direction of the boundary line. These steps are displayed in Figure 266. Additionally, in preparation for the inclination procedure, the diagonal lines and vertical lines of the hexagons are grouped separately because of the difference in line thickness. This is visible in Figure 267.

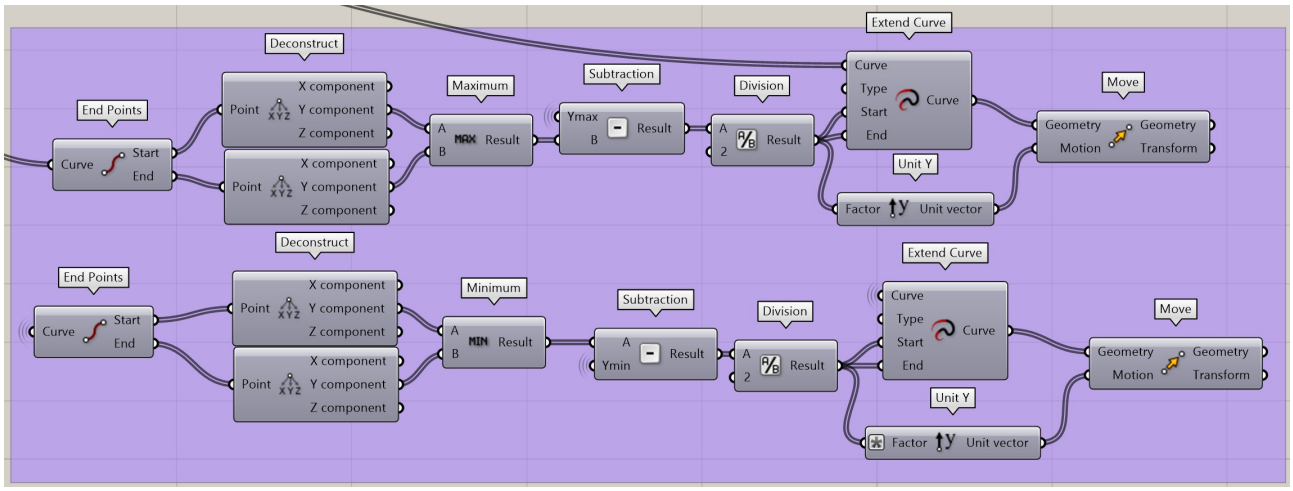


Figure 266: Extending the vertical lines and attaching to the boundary line.

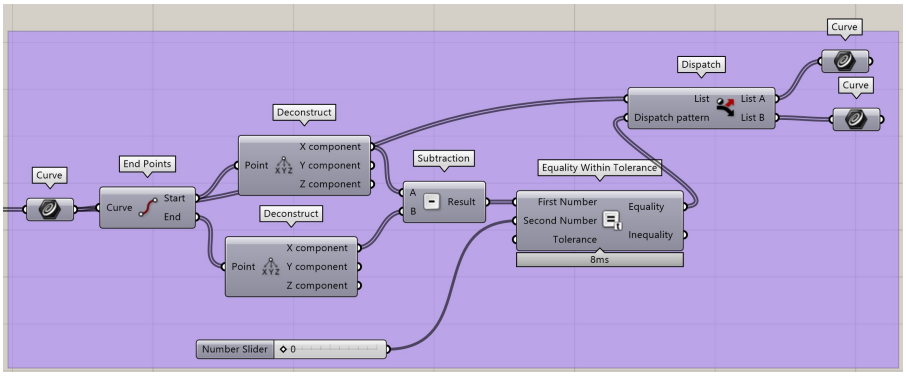


Figure 267: Grouping vertical and diagonal lines of hexagons.

However, the separation in groups has to be slightly different if the hexagon pattern is rotated, which is similar to the previously discussed procedure of adding inclination. The grouping is based on the same distinction as in the previously described addition of inclination, although extra groups are added based on line thickness, which is important in the structural model. The steps are shown in Figures 268, 269, and 270.

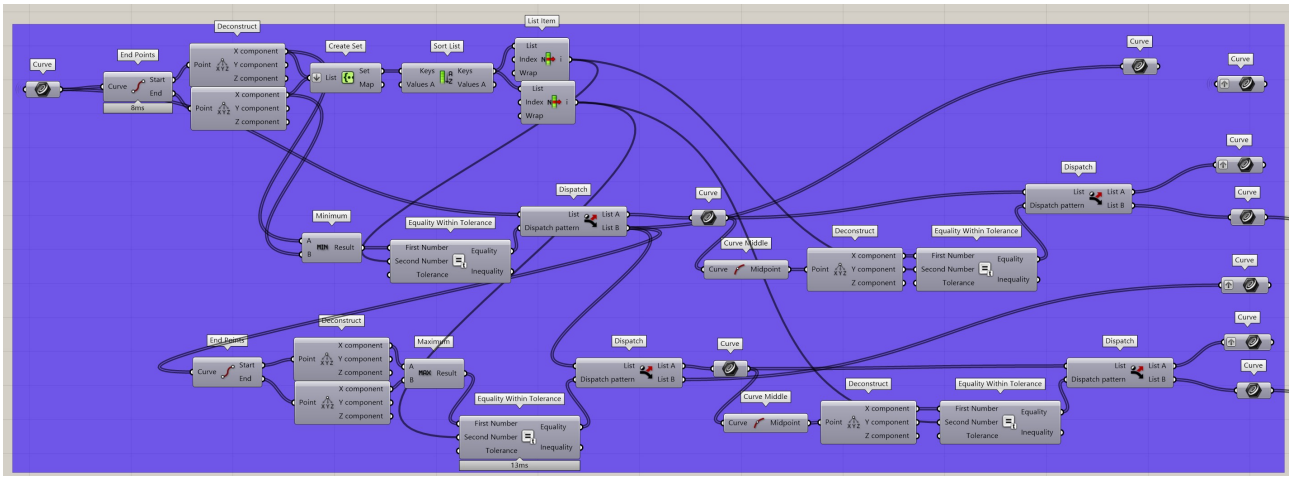


Figure 268: First part of grouping the lines in case the hexagon pattern is rotated.

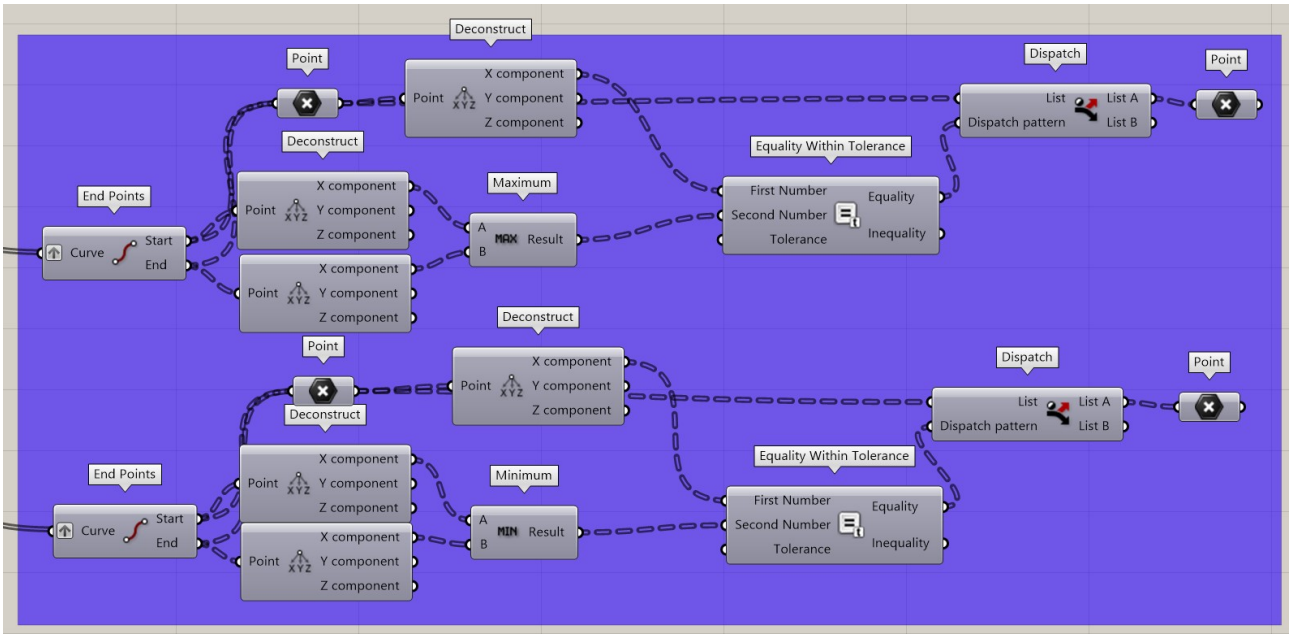


Figure 269: Second part of grouping the lines in case the hexagon pattern is rotated.

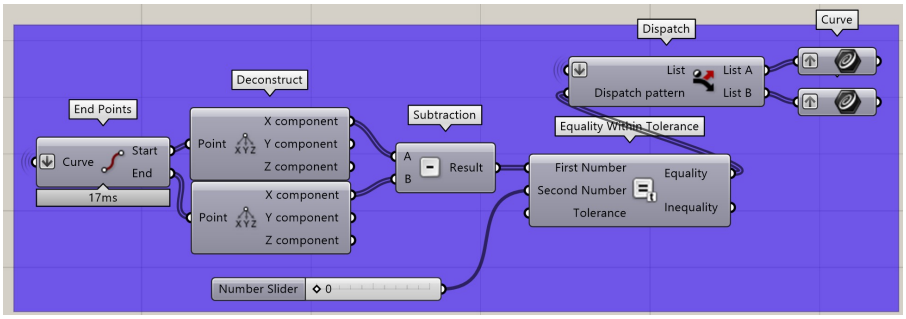


Figure 270: Third part of grouping the lines in case the hexagon pattern is rotated.

The last preparation for adding the inclination is to determine the factor that accounts for the inclination angle, as shown in Figures 271 and 272. Also, the vectors in the z-direction are determined, as shown in Figure 273. Furthermore, if the vertical lines are attached to the upper and lower boundary lines, a special vector in the y-direction is formulated as the boundary lines will not be inclined. This is depicted in Figures 274 and 275.

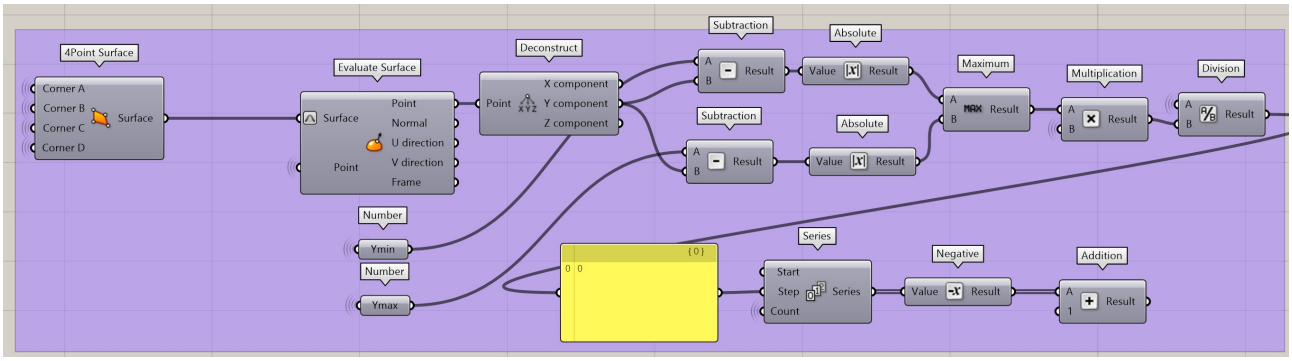


Figure 271: Determination factor with normally orientated hexagon pattern.

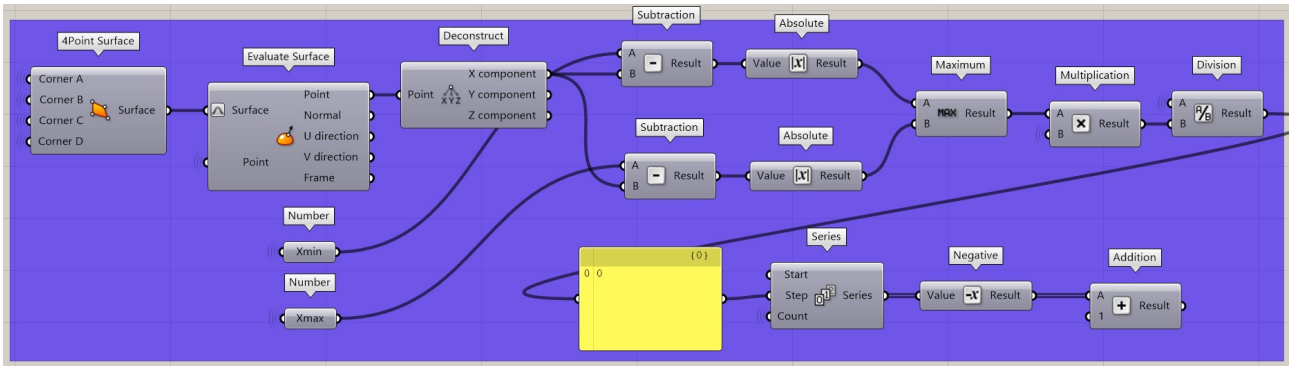


Figure 272: Determination factor with rotated hexagon pattern.

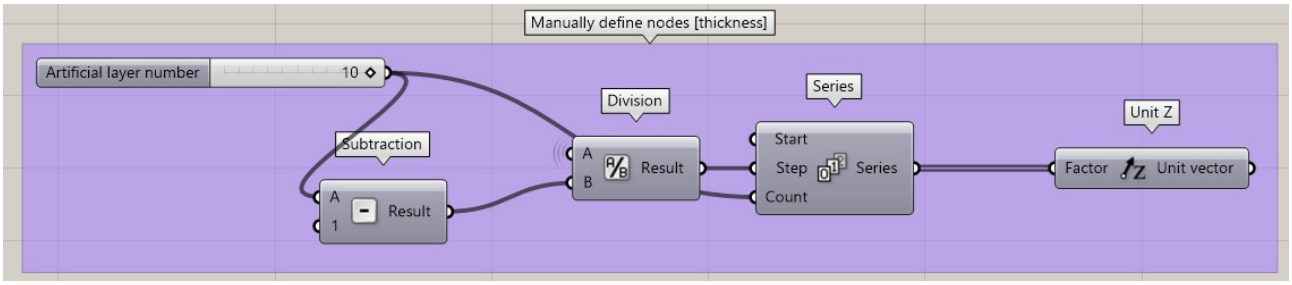


Figure 273: Determination vectors z-direction.

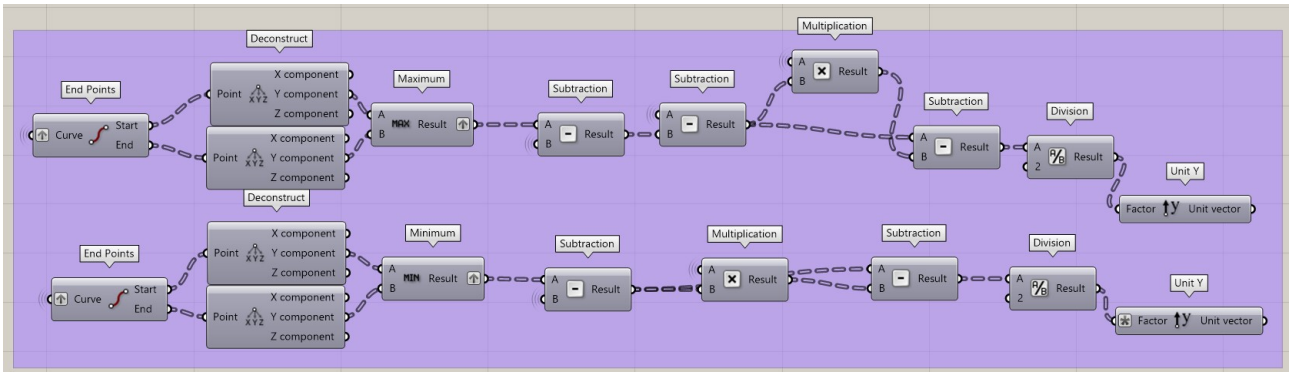


Figure 274: Determination vectors y-direction of vertical lines in the center of the panel.

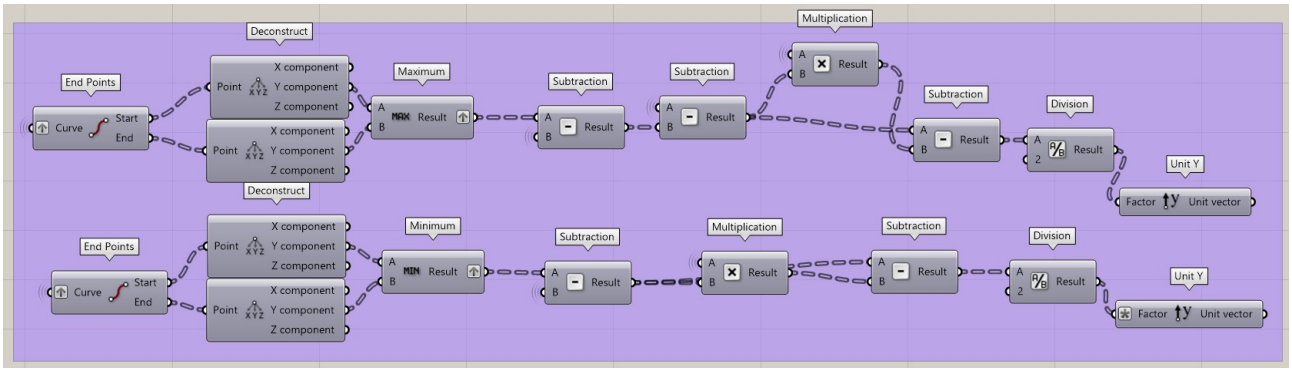


Figure 275: Determination vectors y-direction of vertical boundary lines.

After preparing the lines of the core in groups based on the inclination procedure, the inclination can be added to them. This process is also divided into a normal-oriented hexagon core and a rotated one. The normal-oriented hexagon uses eight different groups of lines to add the inclination, as shown in Figure 276. The red colour represent lines with a double line width, and the green colour represents lines with a single line width. The following line groups are used: vertical lines attached to the lower boundary; vertical lines attached to the upper boundary; upper and lower boundary; left and right boundary segments attached to the bottom boundary; left and right boundary segments attached to the upper boundary; left and right boundary segments with a single line width not attached to either the upper or lower boundary; all other vertical line segments with a double line width; and the diagonal lines. The distinction is thus either based on the line width or the procedure to include the inclination. Subsequently, as shown in Figure 277, for each of these groups, layers are created over the thickness of the core. Each column represents a single group, and the colours have the same meaning. This process is performed as a preparation step to induce a certain mesh density along the thickness. In this case, there will be nine elements along the thickness.

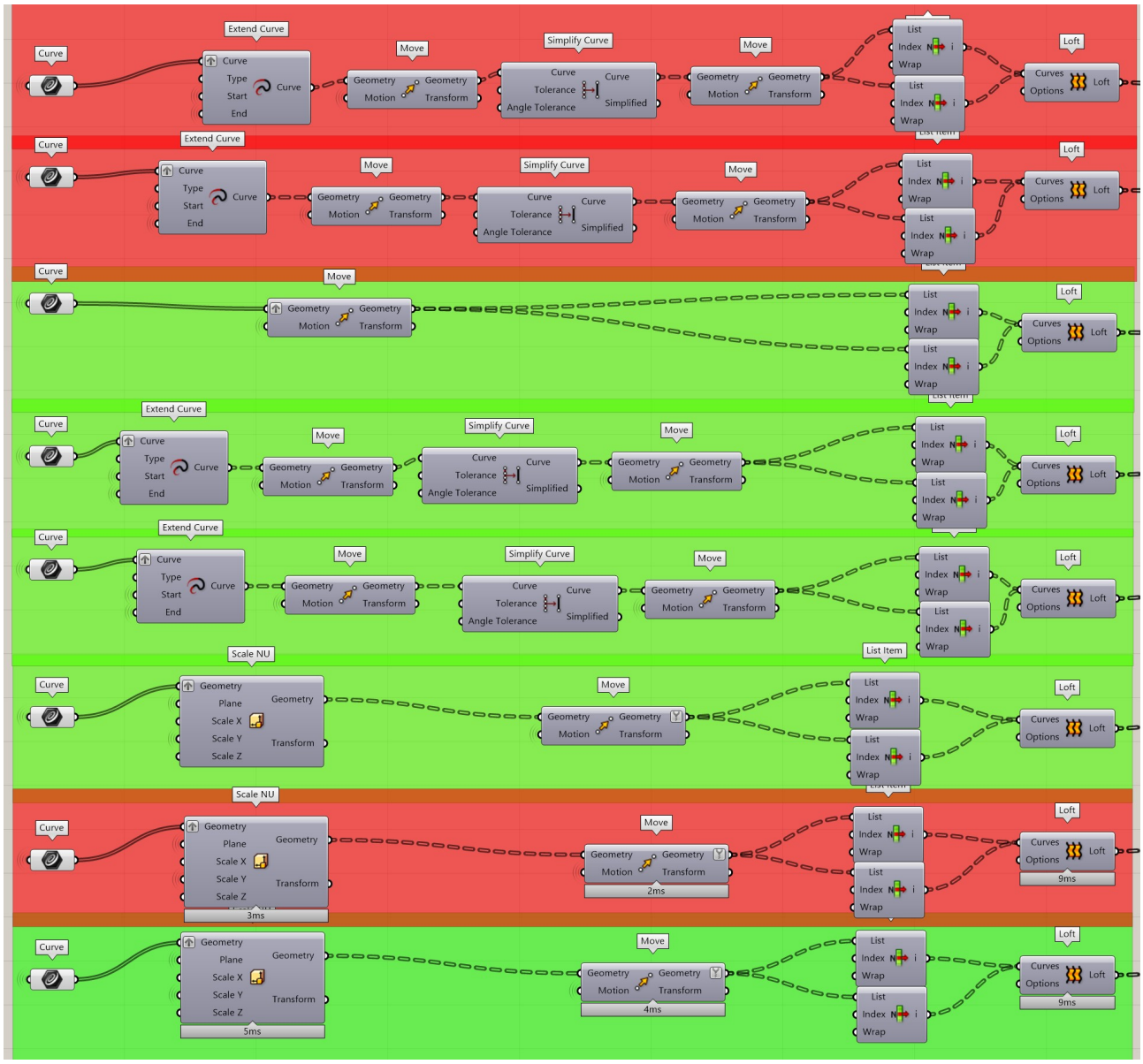


Figure 276: Adding inclination to the center lines with normal rotation.

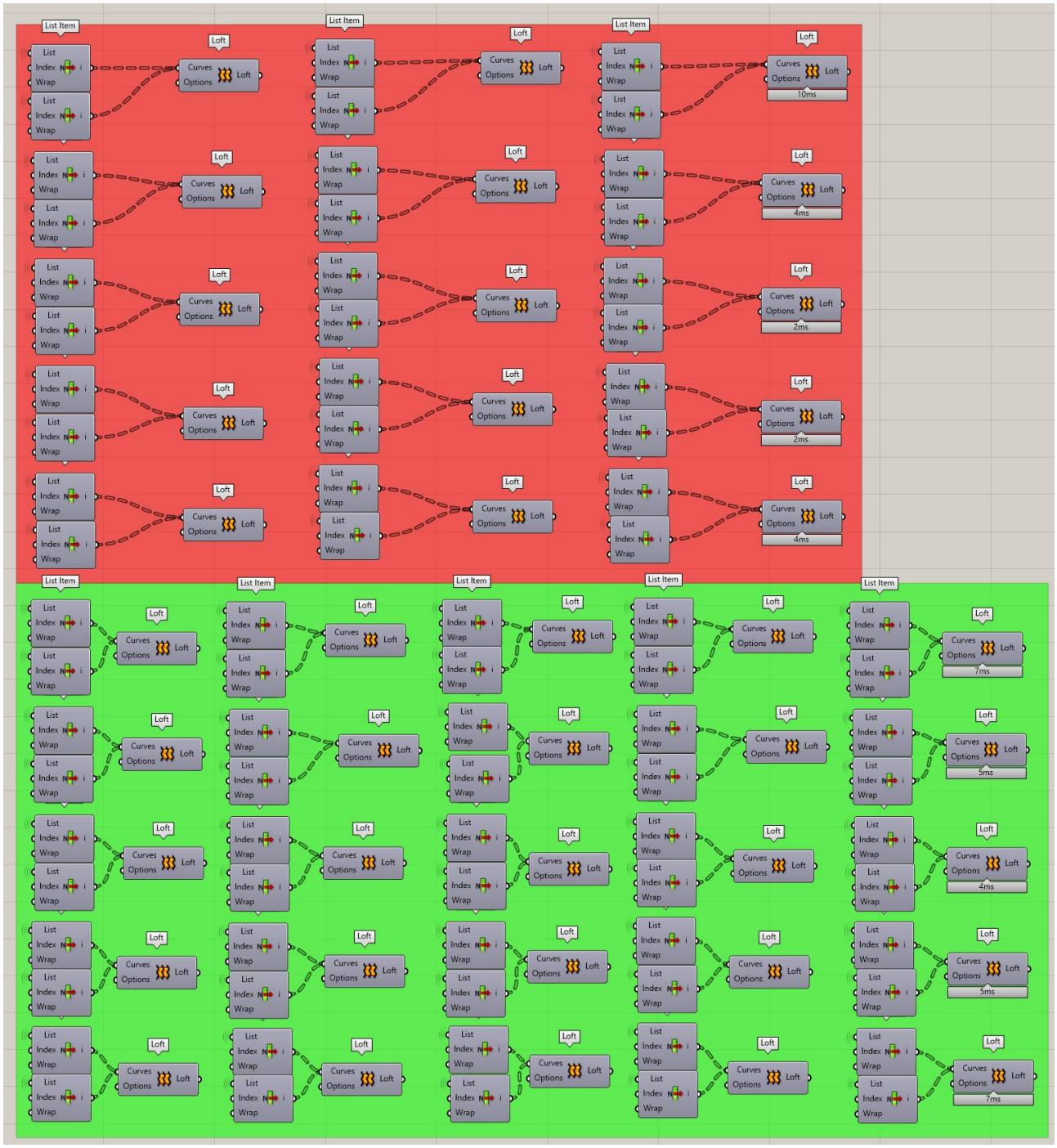


Figure 277: Dividing core pattern into layers as a preparation for meshing in the structural model.

Additionally, the inclination is added to the rotated hexagon pattern based on line groups that have different line widths or a different procedure to add inclination. This is shown in Figure 278. The following groups are used: boundary line segments with a single line width, lower boundary line segments with a double line width, internal horizontal line segments with a double line width, diagonal lines not attached to boundary lines, upper boundary line segments with a double line width, diagonal lines attached to the lower boundary line, and diagonal lines attached to the upper boundary line. Furthermore, layers over the thickness are created, similar to the normally rotated hexagon pattern shown in Figure 279.

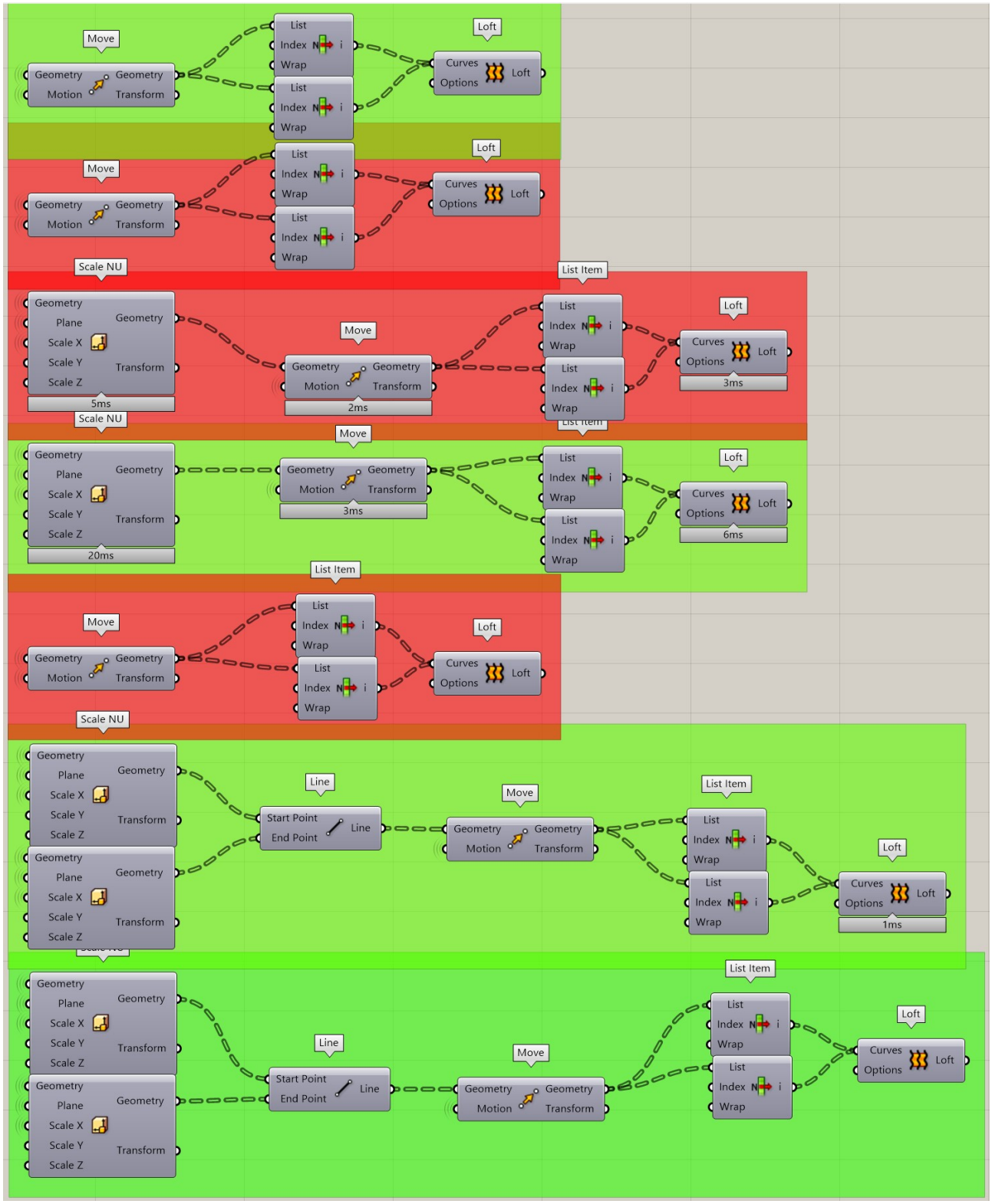


Figure 278: Adding inclination to the center lines with a rotated hexagon pattern.

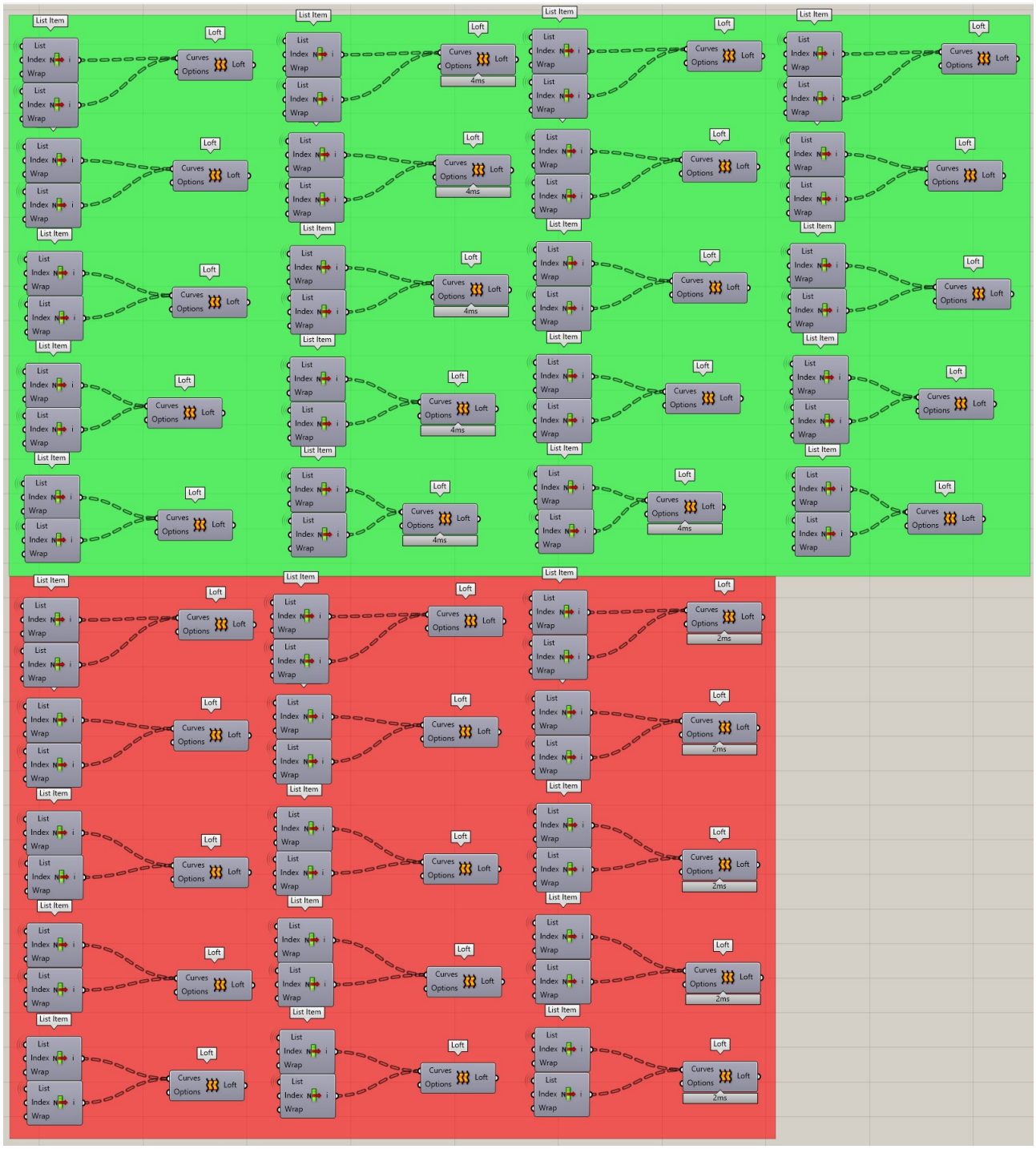


Figure 279: Dividing core pattern into layers as a preparation for meshing in the structural model with a rotated hexagon pattern.

Finally, as the geometry is calculated for both orientations, the orientation that is in line with the input parameter concerning the orientation has to be applied. This process uses Python components and is simply based on if-statements, as depicted in Figure 280.

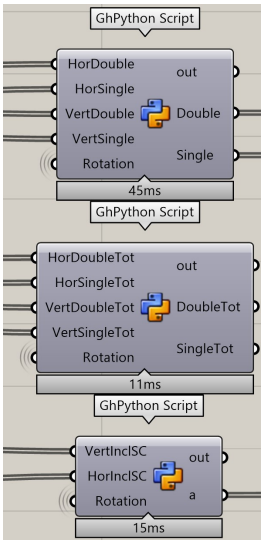


Figure 280: Determining the relevant geometry needed for the structural model.

Structural Model (Karamba3D)

The structural model requires a geometry, a chosen mesh density, material properties, loading conditions, boundary conditions, and finally the calculation method. These steps are discussed in this order. The geometry has already been created in the previous sub-chapter and the glass layers are generated; this is shown in Figure 281. The second step is to determine the mesh density. The mesh density is already partly determined because, on the previous pages, the number of layers along the thickness describes the number of elements along the thickness. In this case, there are 9 layers, which means there are 9 elements along the thickness of the core. Furthermore, the number of elements in the area can be determined per optimisation problem and is based on the required accuracy and computation time. This is depicted in Figure 282.

After defining the geometries and mesh density, the material properties are specified, and the mesh is applied to the geometries. This is shown in Figure 283. Four groups of shell elements can be seen. The top two components represent the two glass layers. The other two represent the core geometry with either a single or double line width.

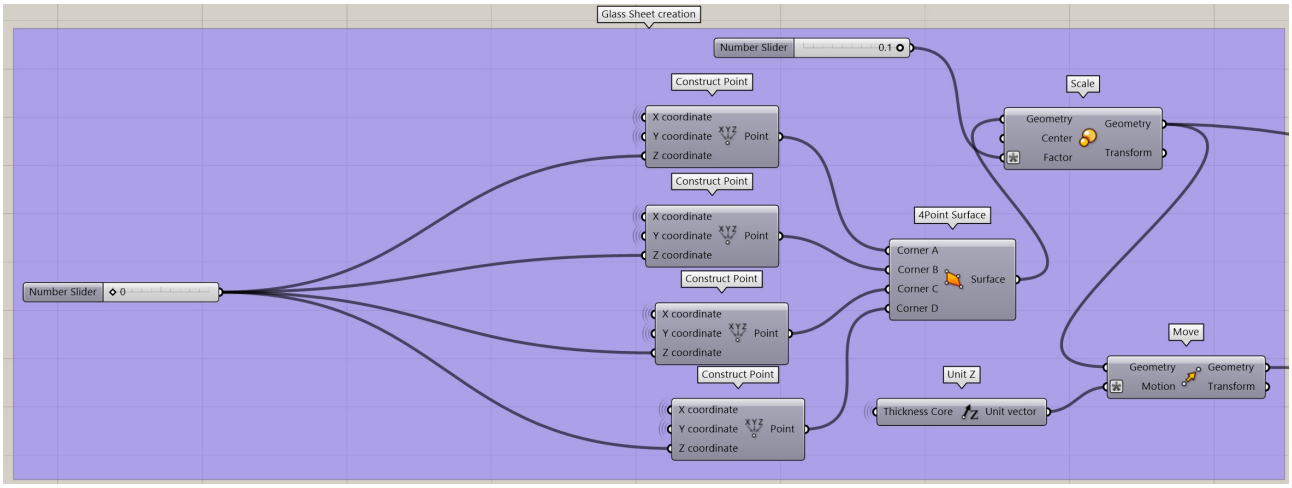


Figure 281: Generating the glass layers.

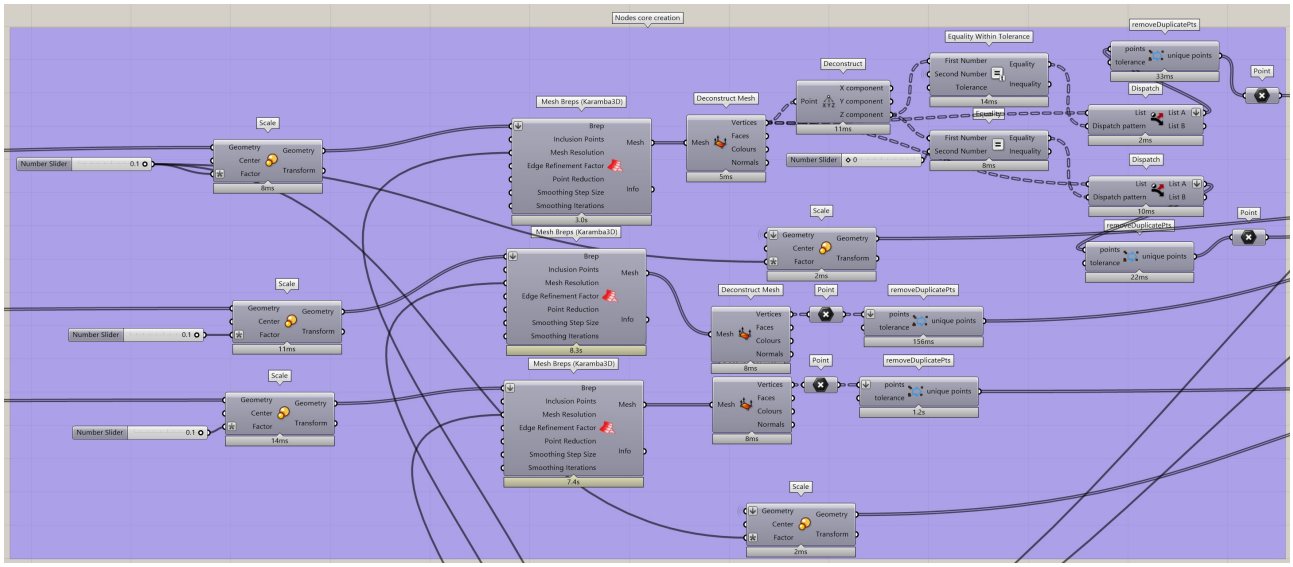


Figure 282: Determining the position of the nodes and elements.

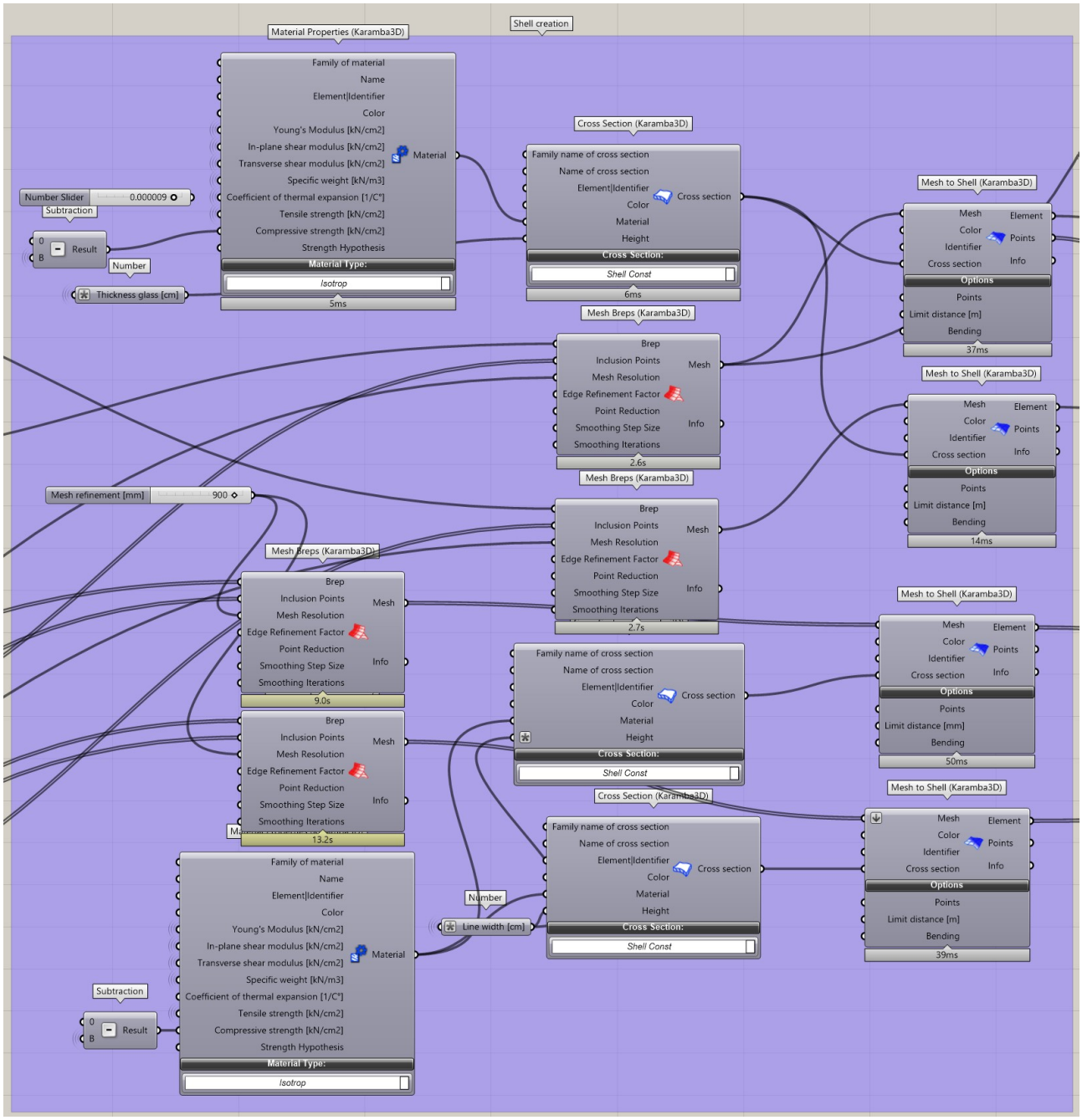


Figure 283: Adding the material properties and applying the mesh to the geometries.

Next, the loading conditions are defined in Figure 284. It has to be noted that the structural model is based on center lines; therefore, the area of the facade has been reduced slightly. This change in area has to be included in the magnitude of the load. Thus, the area load is increased slightly as the input parameter concerning the load is based on the volumetric representation. Also, the boundary conditions have to be defined. All possible support conditions are defined in Figure 285, where the support conditions are limited to hinge supports. Then, the actual boundary conditions are determined in Figure 286, which uses the input parameters. The combination of supports is designed such that there is no horizontal restriction in the movement of the panel. For example, if two supports on either side are used, one is a hinge support and the other is a rolling hinge support.

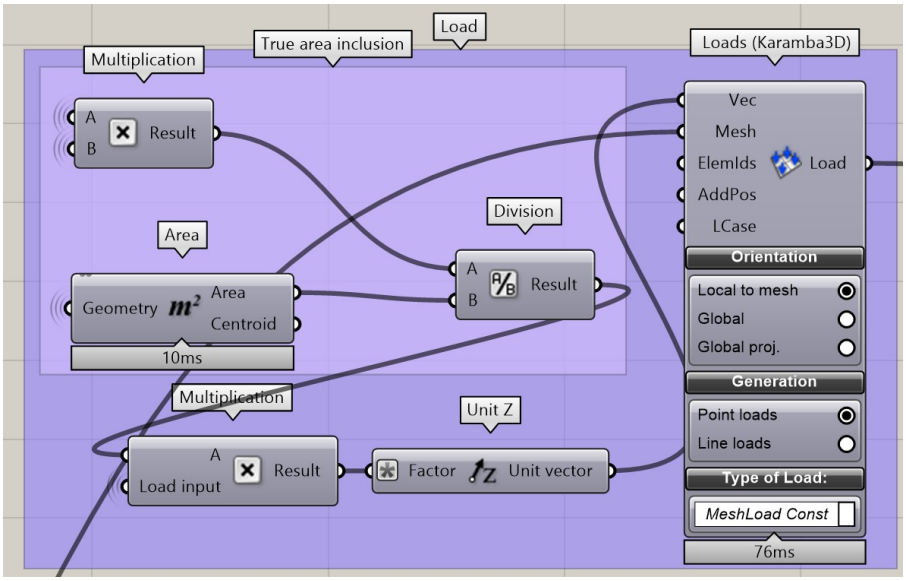


Figure 284: Defining the loading conditions.

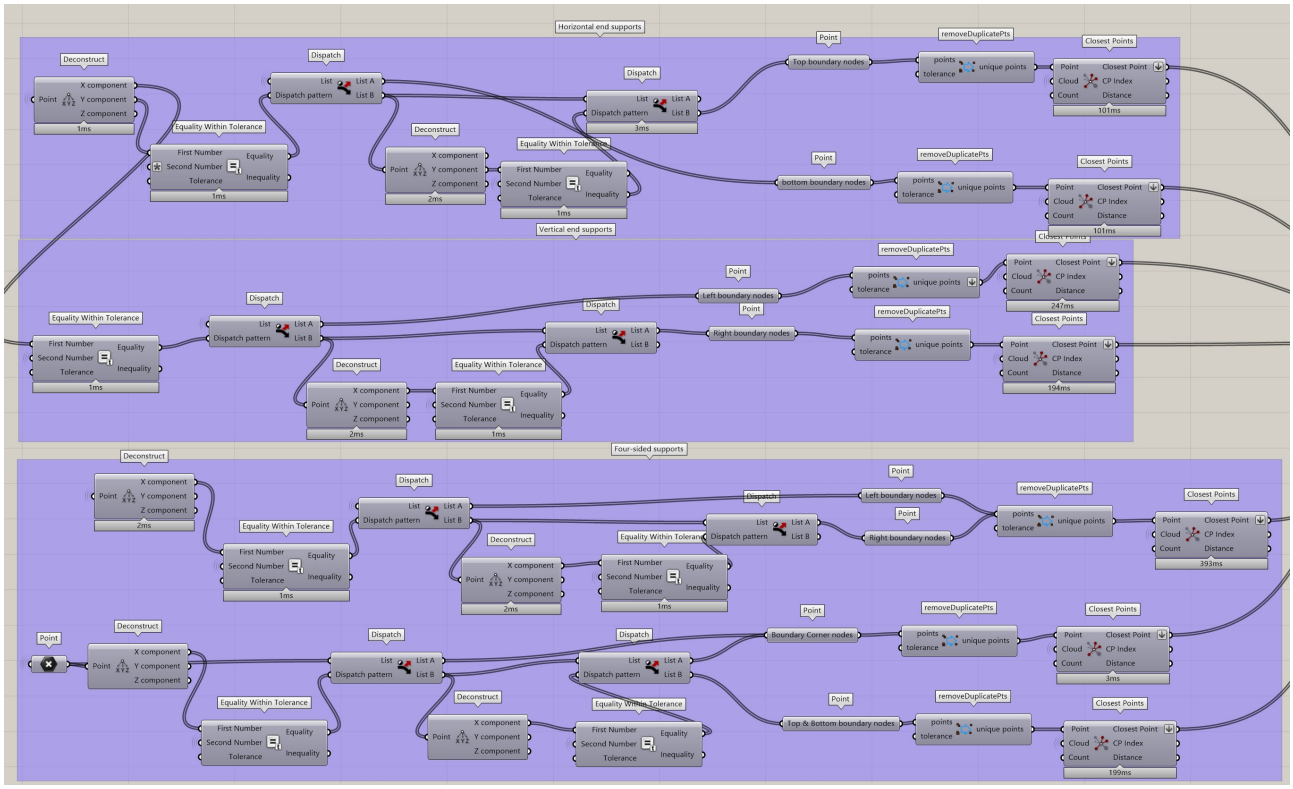


Figure 285: Defining the possible support conditions.

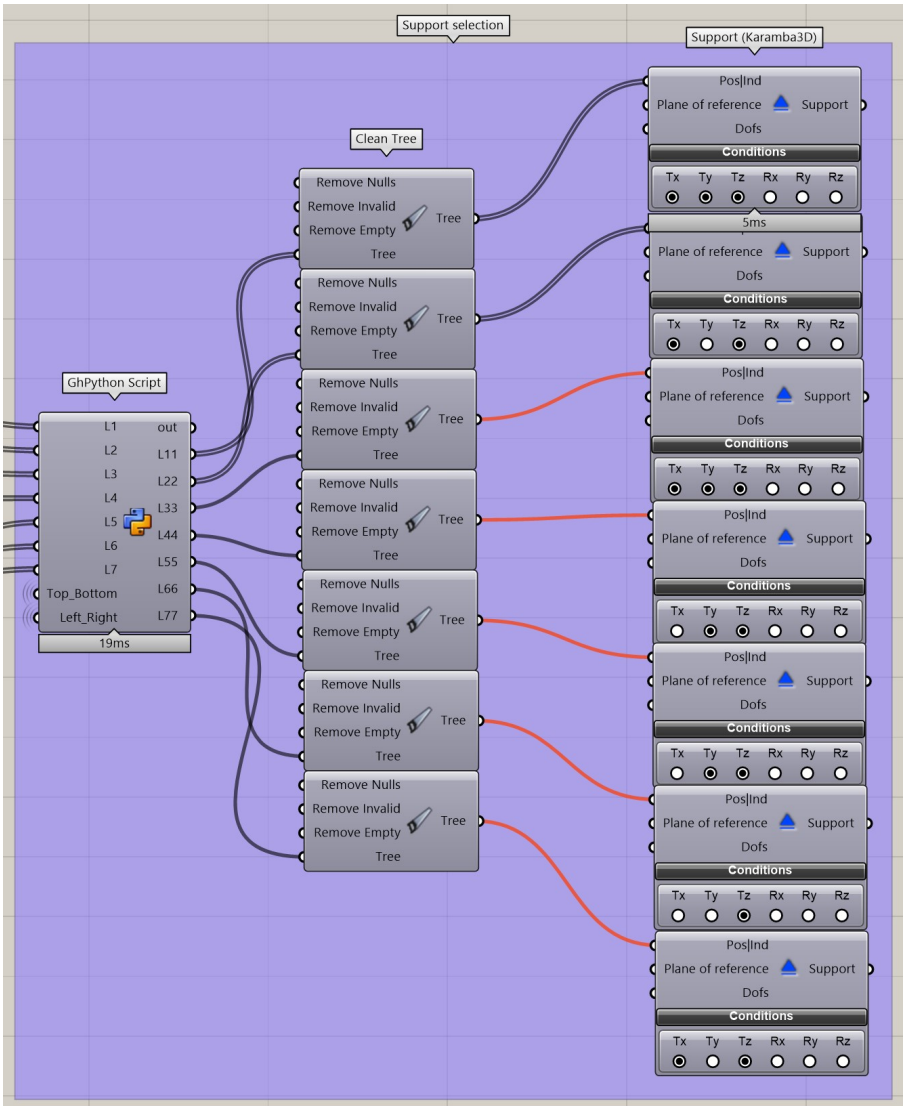


Figure 286: Determining the actual loading conditions based on the input parameters.

After defining the complete structural model, a calculation method has to be chosen. It is chosen to use a linear model for small deflections. In the end, the maximum deflection and Unity Check are determined, which will be used in the structural constraint. This is shown in Figure 287.

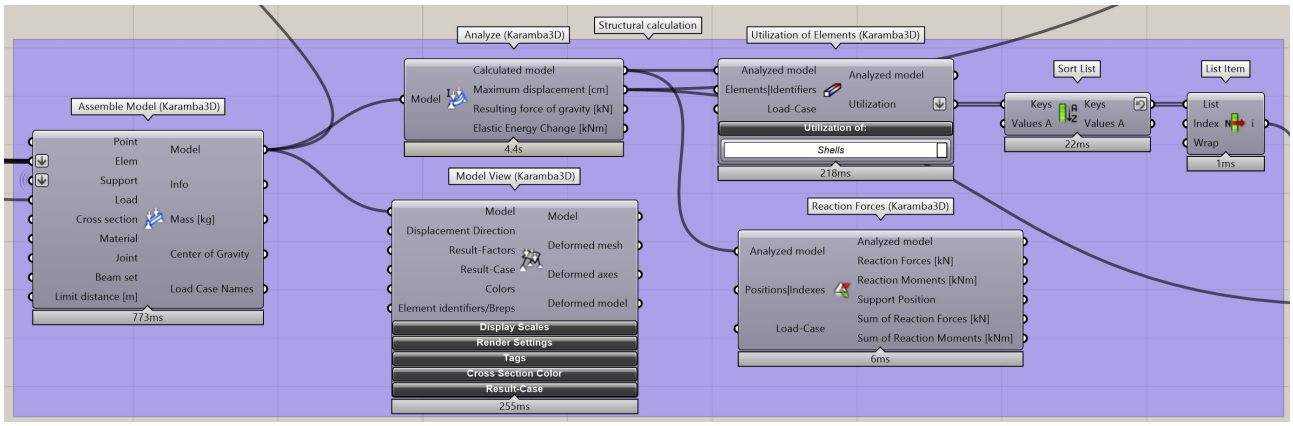


Figure 287: Calculation method of the structural model.

Optimisation procedure

The optimisation procedure first requires defining the criteria and the constraints. The three criteria are displayed in Figures 288, 289, and 290, where each of them represents the same formula as described in Chapter 10. The transparency criterion is simply the transparent area, and the insulation criterion uses the formula defined in Chapter 4. The daylight criterion is equal to a ratio shown in Equation (59). The structural constraint is based on the deflection, which cannot be larger than the diagonal divided by 65 or 50 mm. This is defined in Chapter 3. Subsequently, the Unity Check cannot exceed 1.0. This is shown in Figure 291.

Finally, all the criteria and constraints are used to formulate the fitness function. Additionally, the optimisation algorithm is the Genetic Algorithm, which uses the 'Galapagos' component. This component requires multiple settings, such as maximum stagnant generations, population size, the initial boost of population in the first generation, and the population percentage to maintain and inbreed. This is visible in Figure 292.

$$Daylight\,criterion = \frac{Total\,solar\,radiation\,winter}{\frac{Direct\,solar\,radiation\,summer}{Total\,solar\,radiation\,summer}} \tag{59}$$

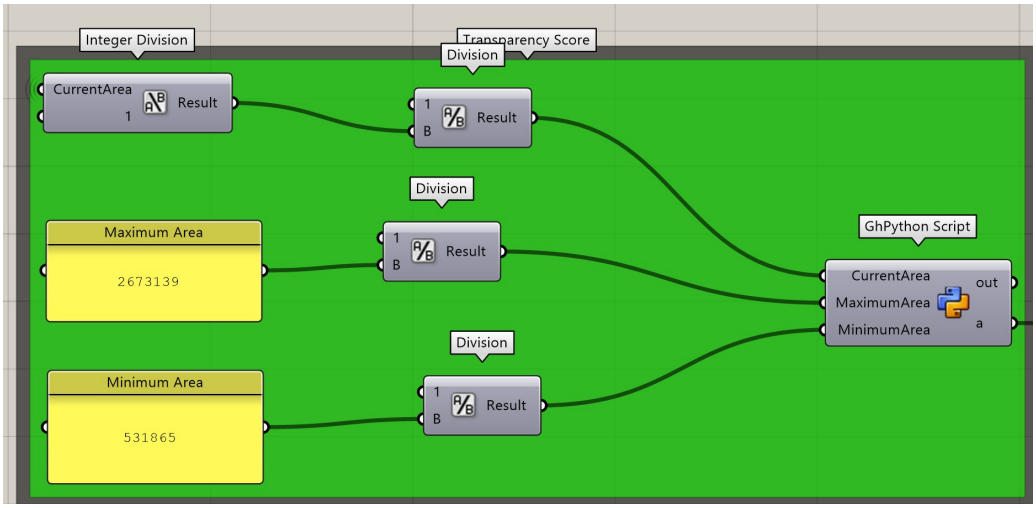


Figure 288: The transparency criterion.

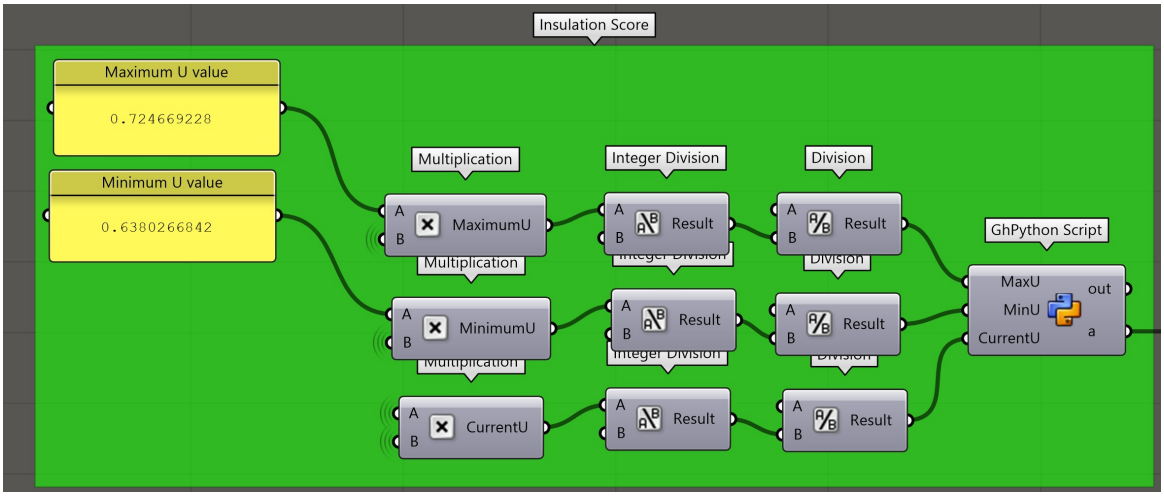


Figure 289: The insulation criterion.

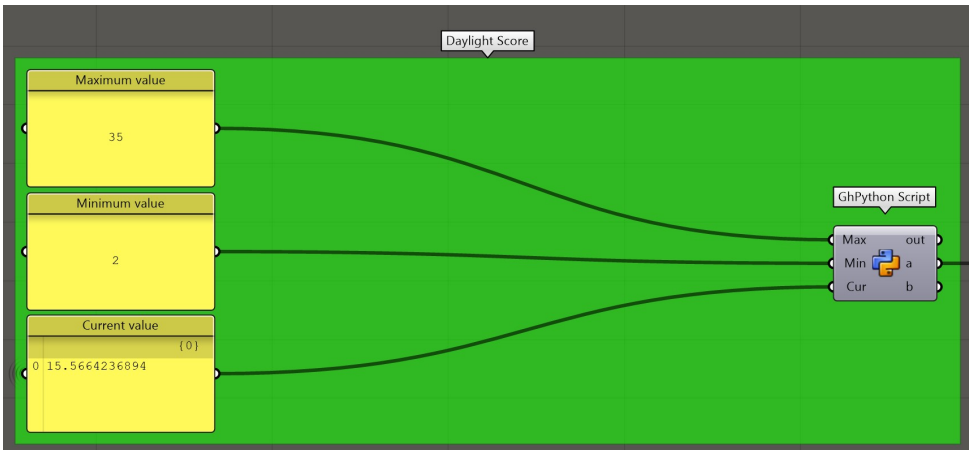


Figure 290: The daylight criterion.

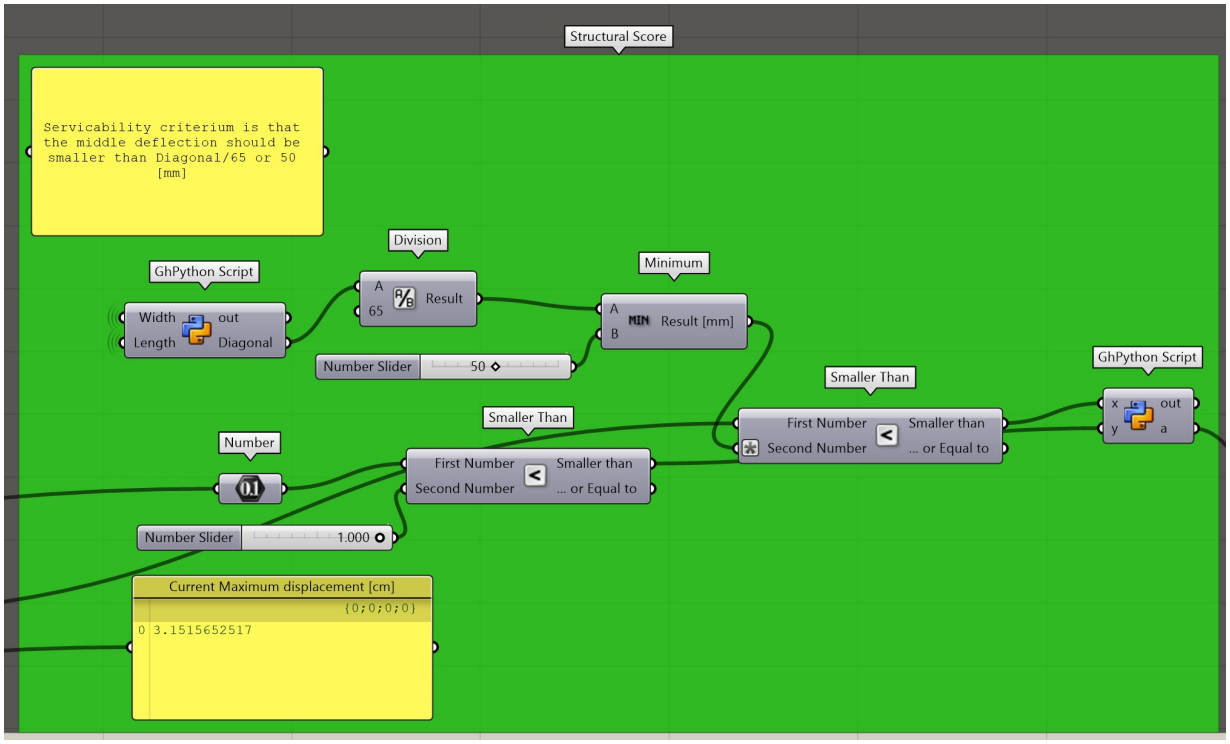


Figure 291: The structural Constraint.

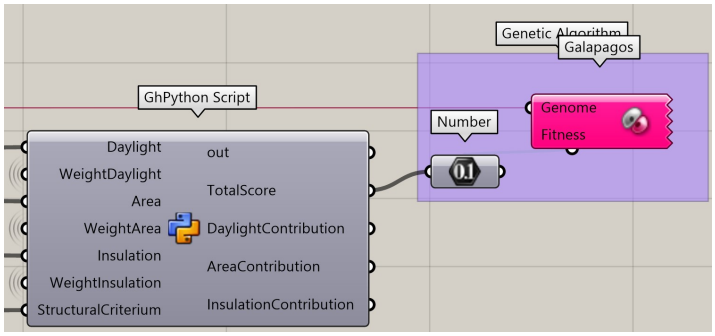


Figure 292: The fitness function and optimisation algorithm.

Appendix I: Strength experiments

Results conducted experiments

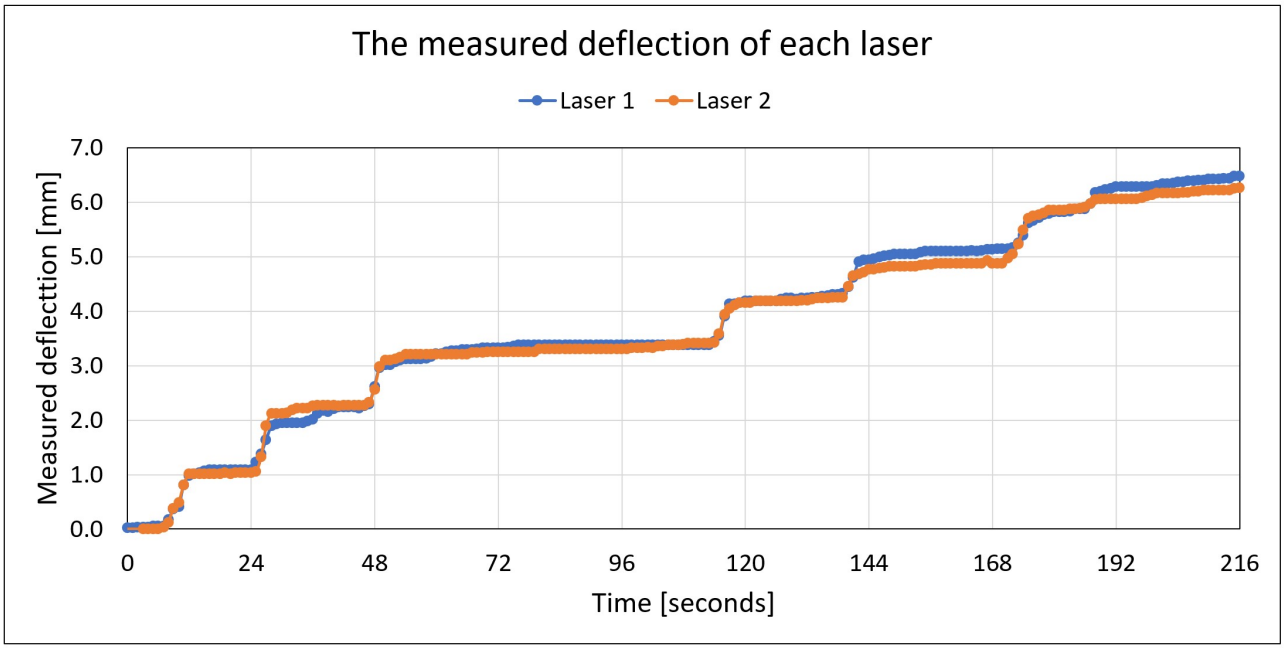


Figure 293: The deflection of the rotated pattern during the experiment.

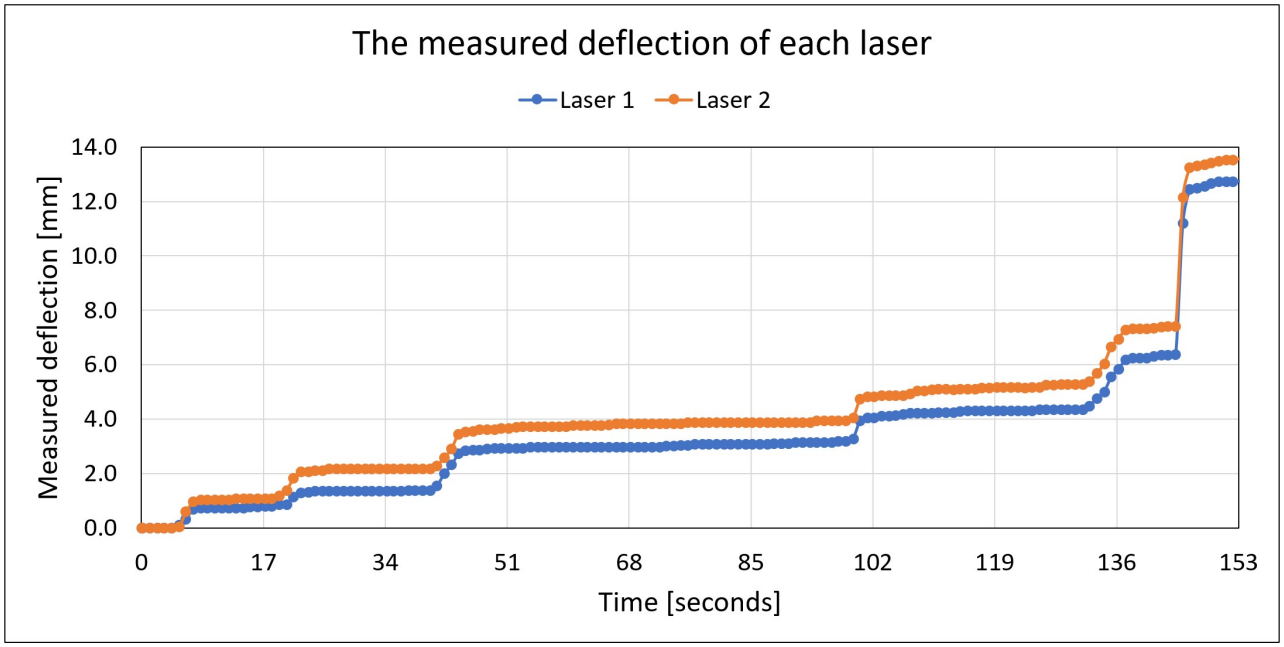


Figure 294: The deflection of the non-rotated pattern during the experiment.

Diana models sandwich panel

Non-rotated pattern, 2 mm element size

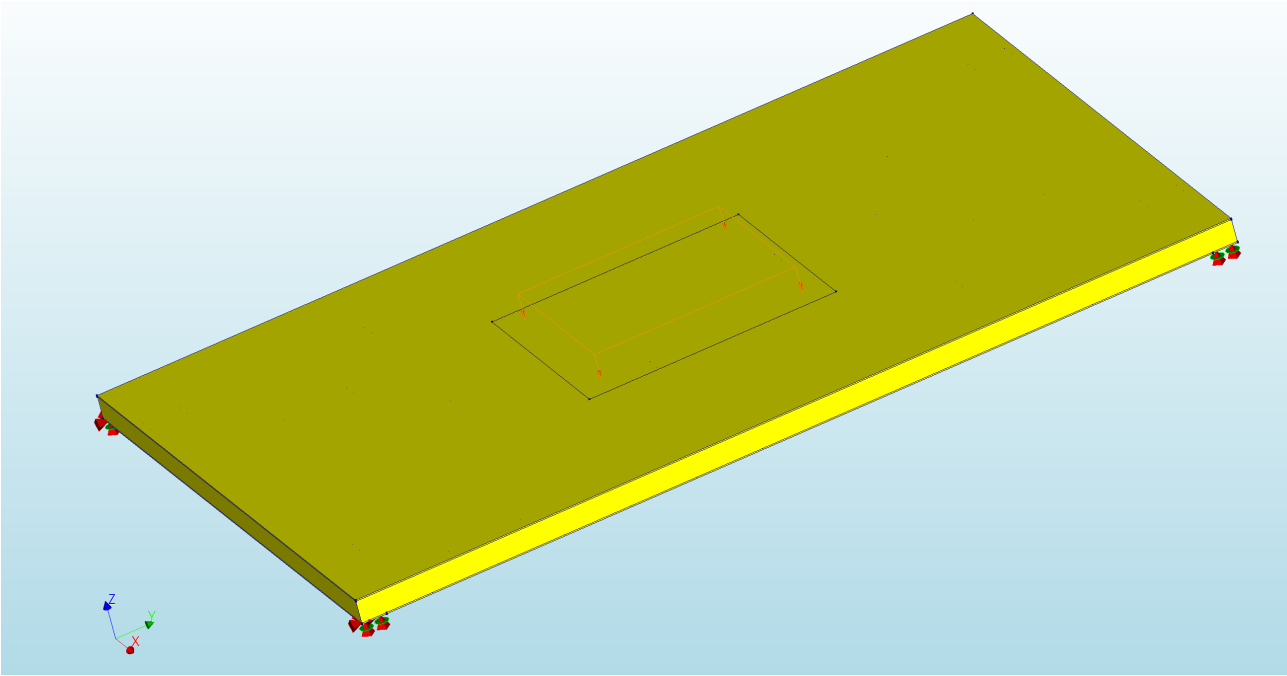


Figure 295: Top view of the geometry of the model with the load, supports, and boundary interface elements.

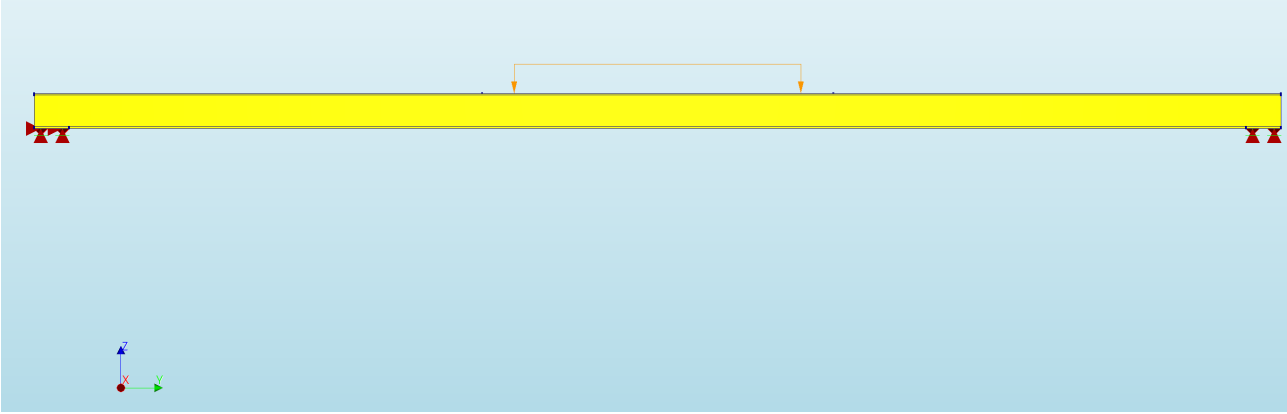


Figure 296: Side view of the geometry of the model with the load, supports, and boundary interface elements.

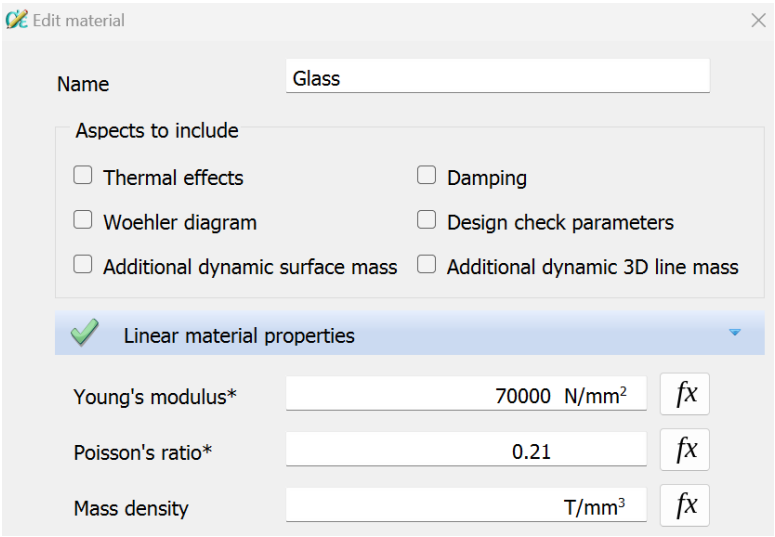


Figure 297: Material properties glass.

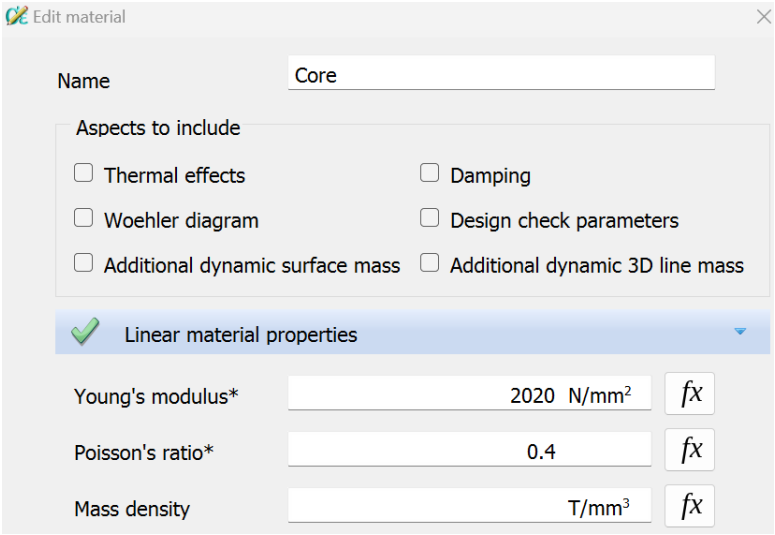


Figure 298: Material properties core.

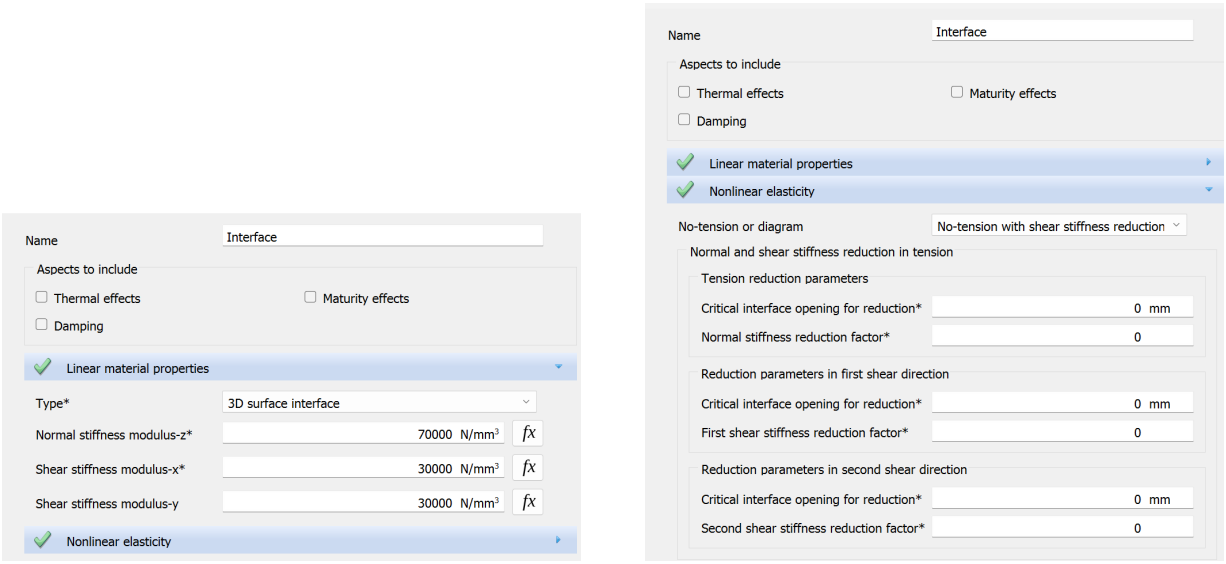


Figure 299: Material properties boundary interface elements.

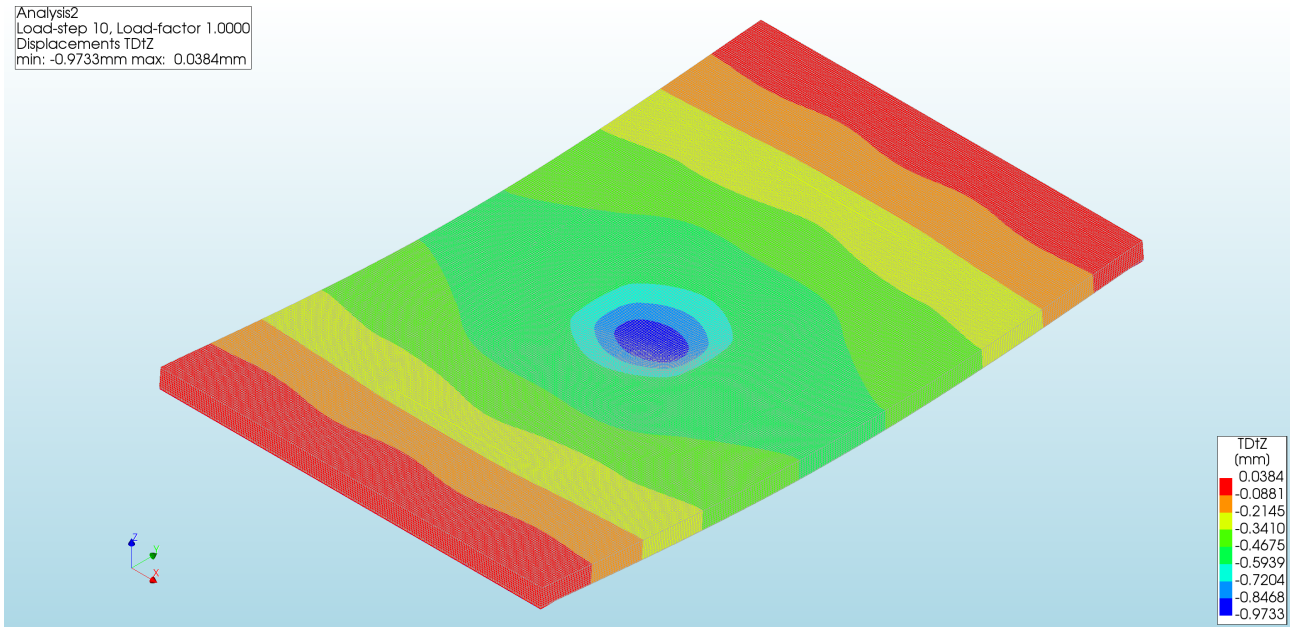


Figure 300: Displacements at last load step seen with top view.

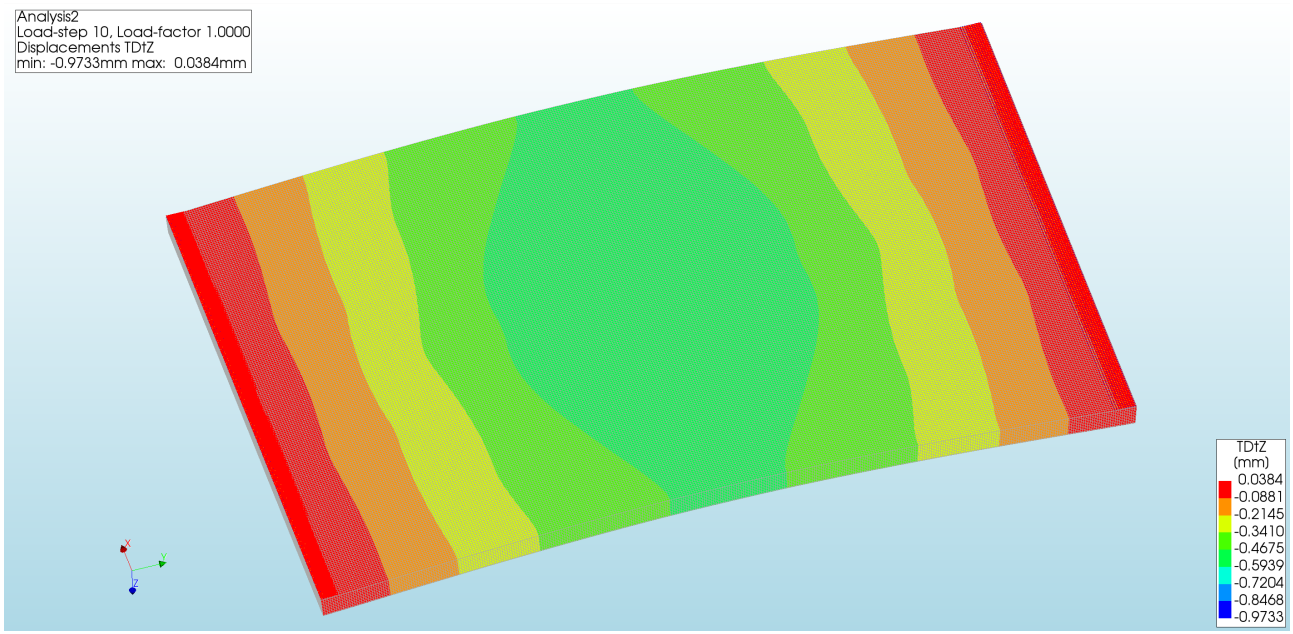


Figure 301: Displacements at last load step seen with bottom view.

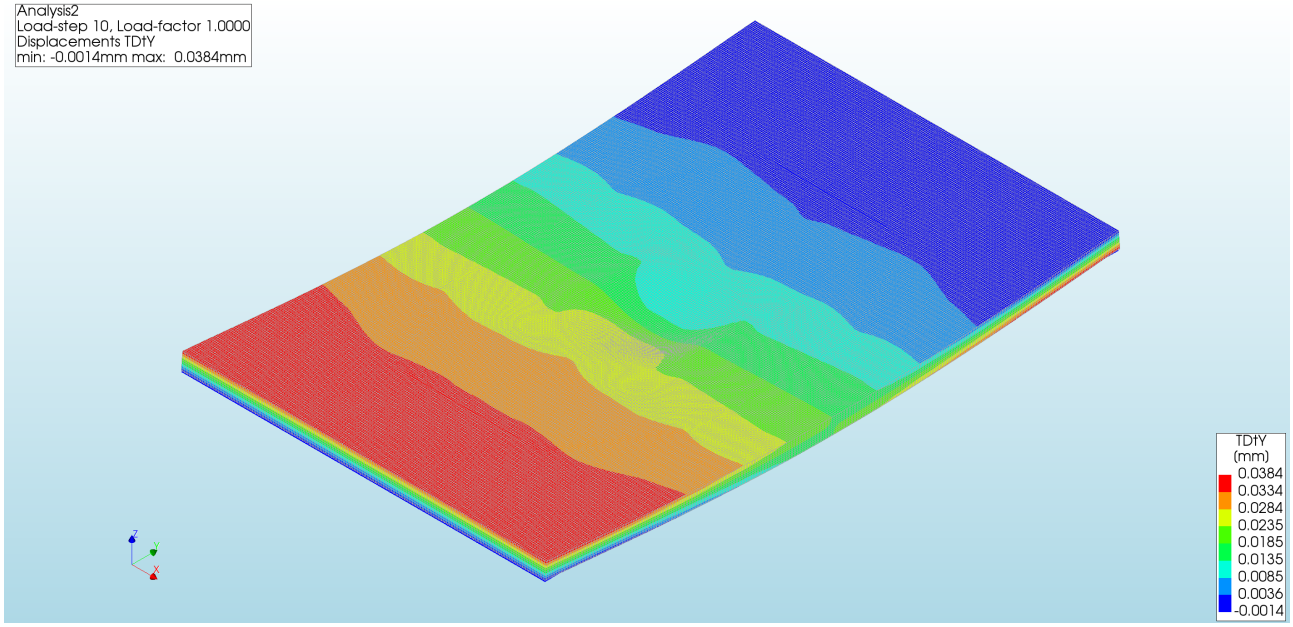


Figure 302: Displacements in y-direction.

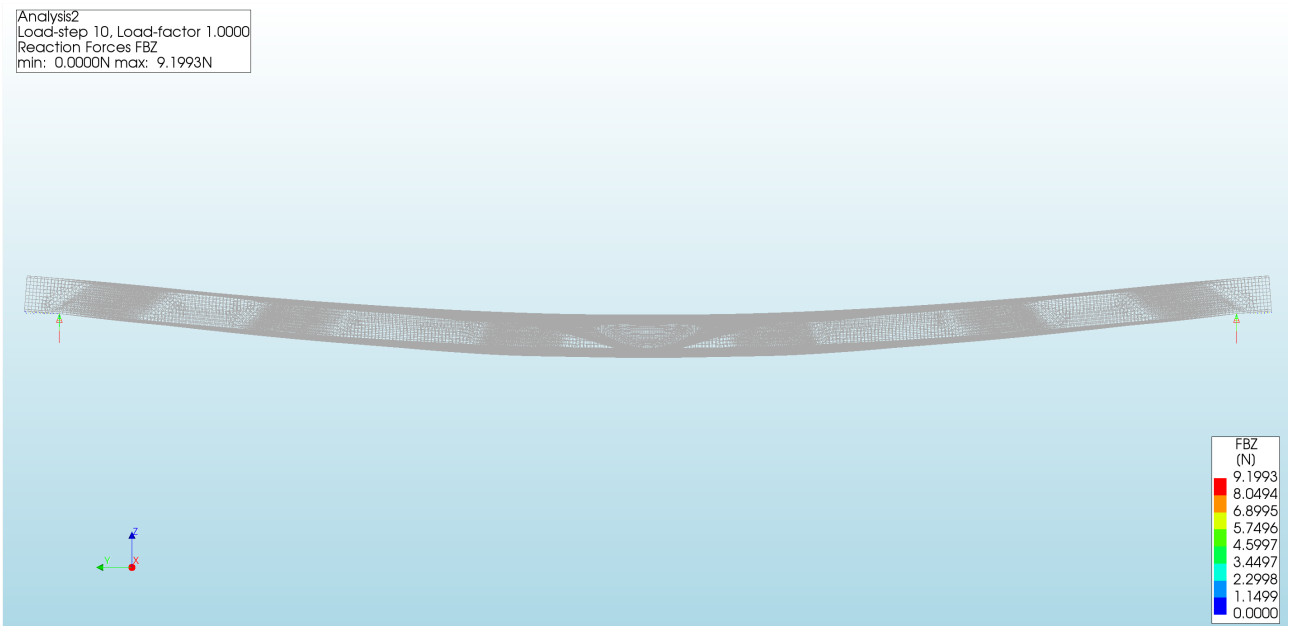


Figure 303: Reaction forces at last load step.

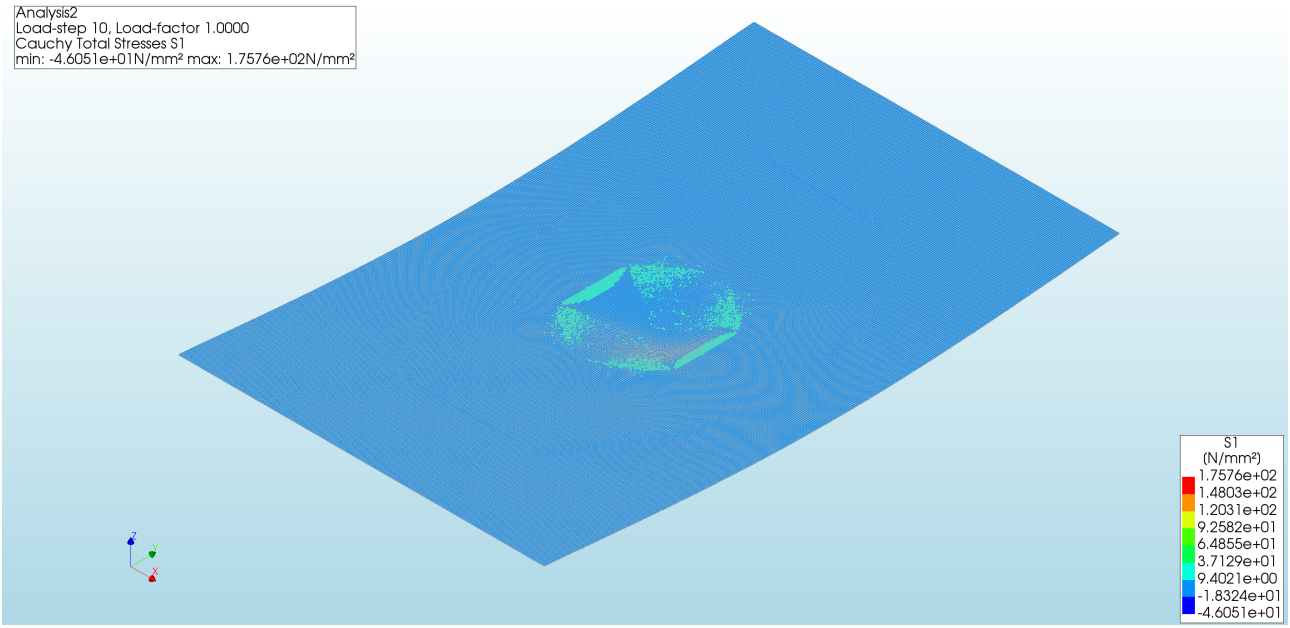


Figure 304: Cauchy principal stress S1 of the top glass layer.

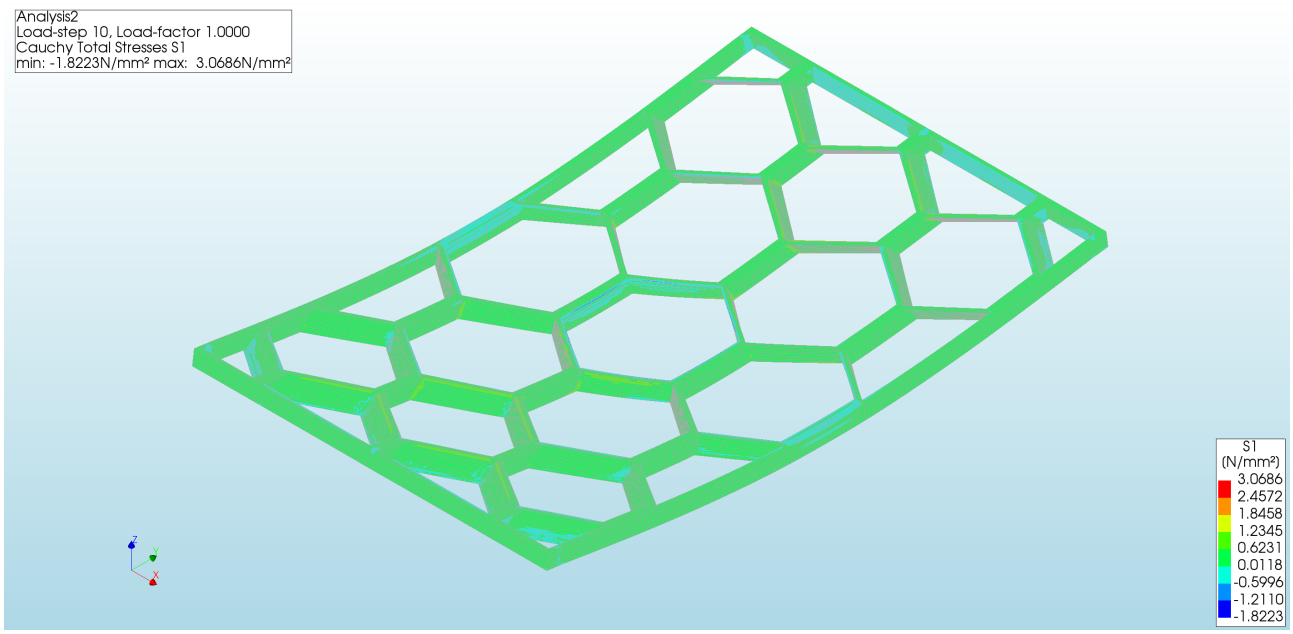


Figure 305: Cauchy principal stress S1 of the core layer.

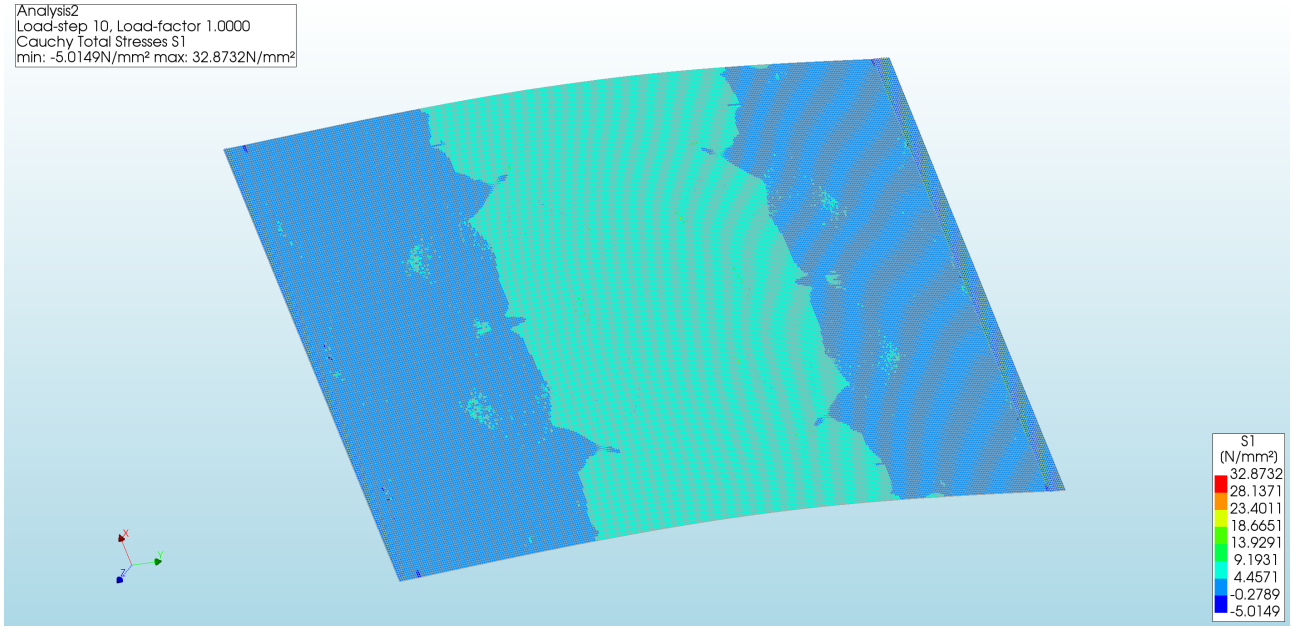


Figure 306: Cauchy principal stress S1 of the bottom glass layer.

Non-rotated pattern, 3 mm element size

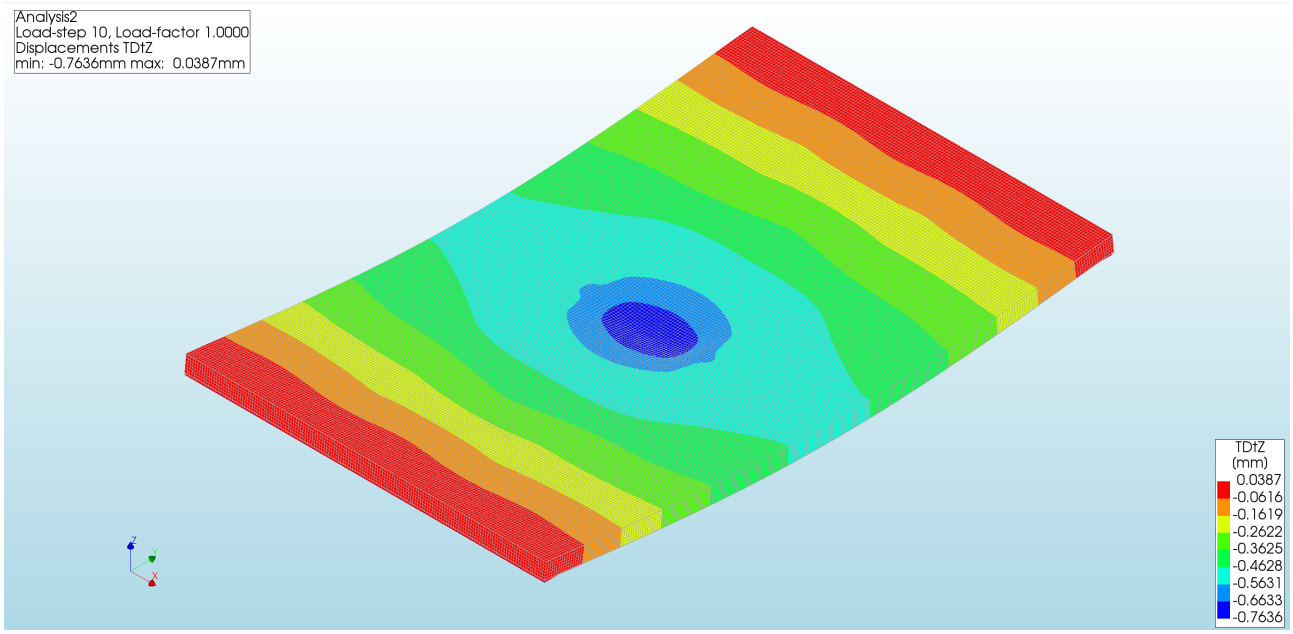


Figure 307: Displacements at last load step seen with top view.

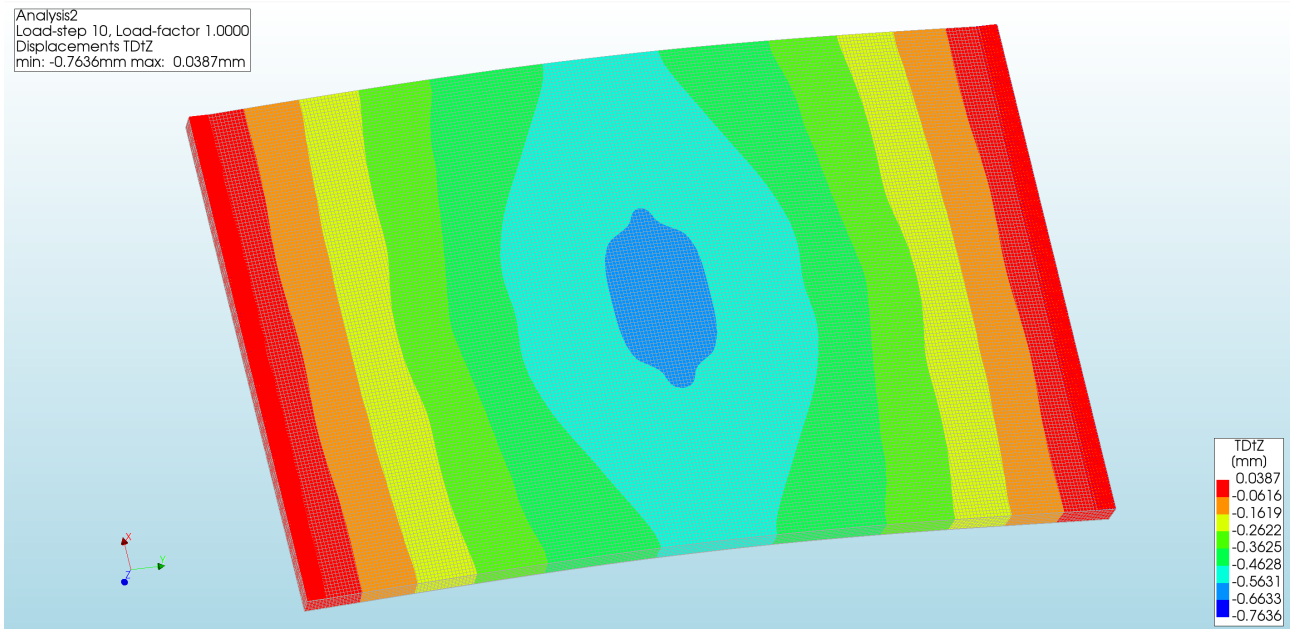


Figure 308: Displacements at last load step seen with bottom view.

Rotated pattern, 1.5 mm element size

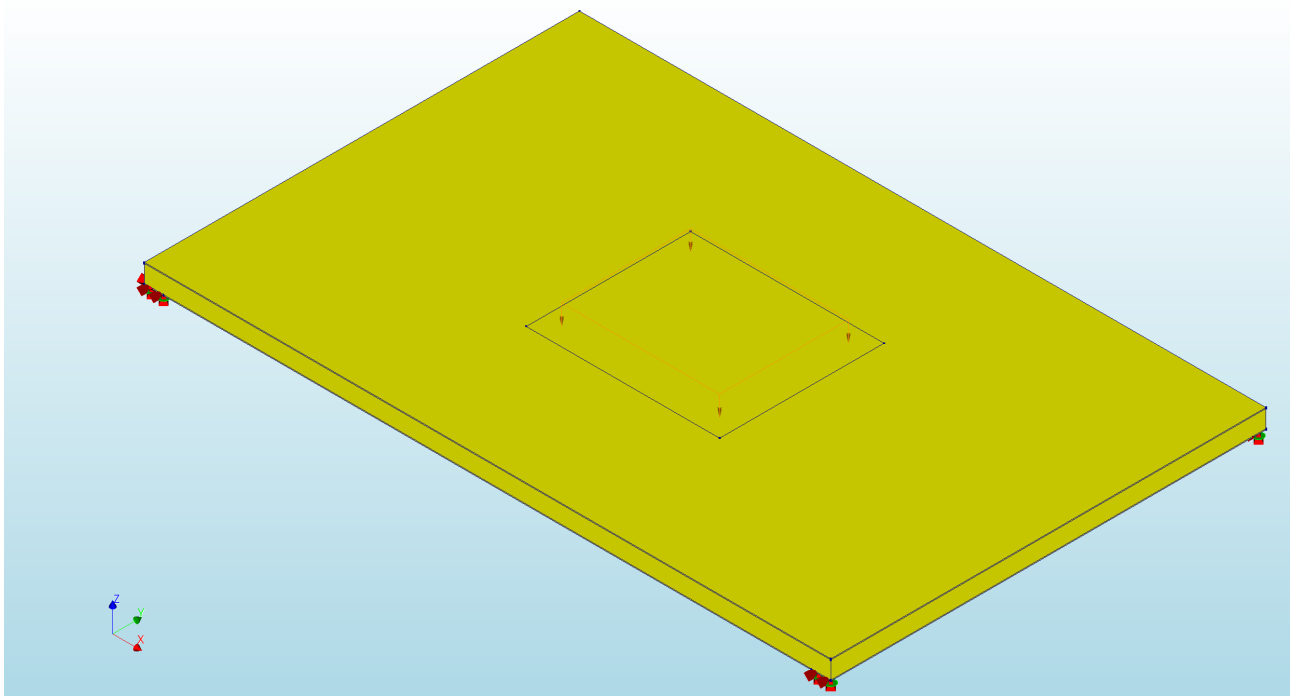


Figure 309: Top view of the geometry of the model with the load, supports, and boundary interface elements.

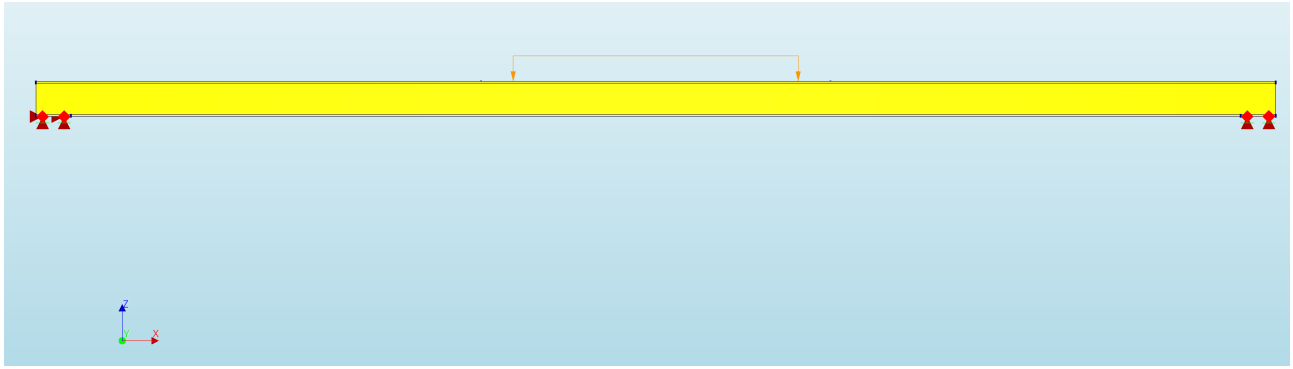


Figure 310: Side view of the geometry of the model with the load, supports, and boundary interface elements.

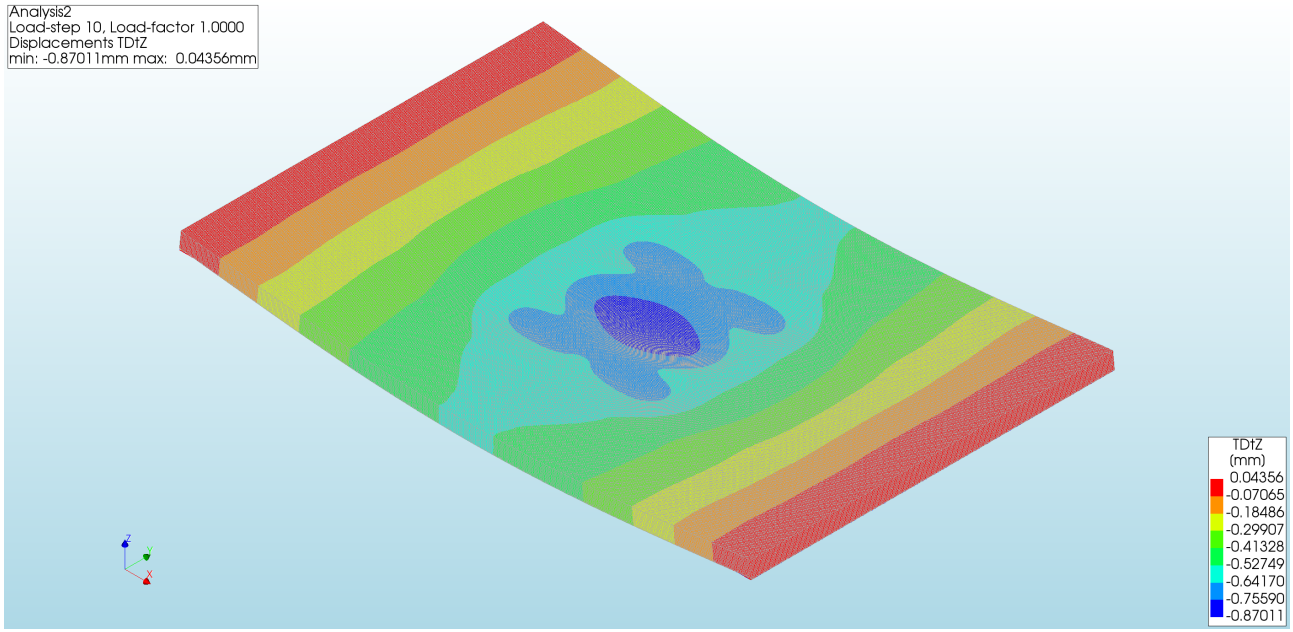


Figure 311: Displacements at last load step seen with top view.

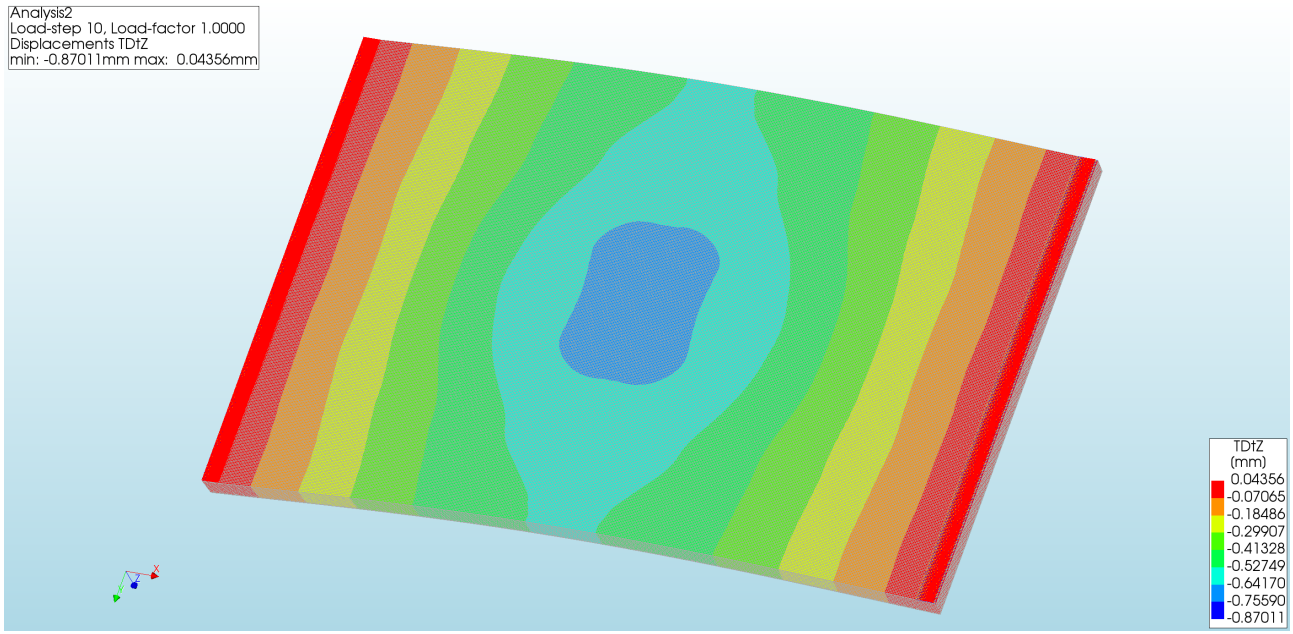


Figure 312: Displacements at last load step seen with bottom view.

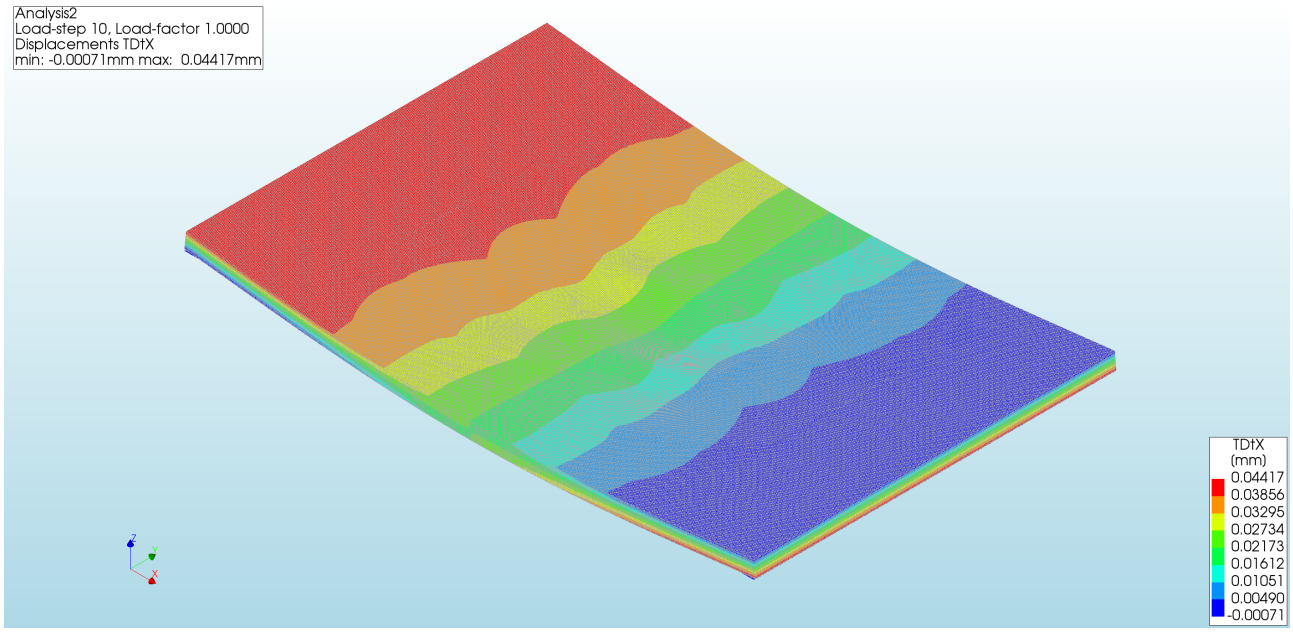


Figure 313: Displacements in x-direction.

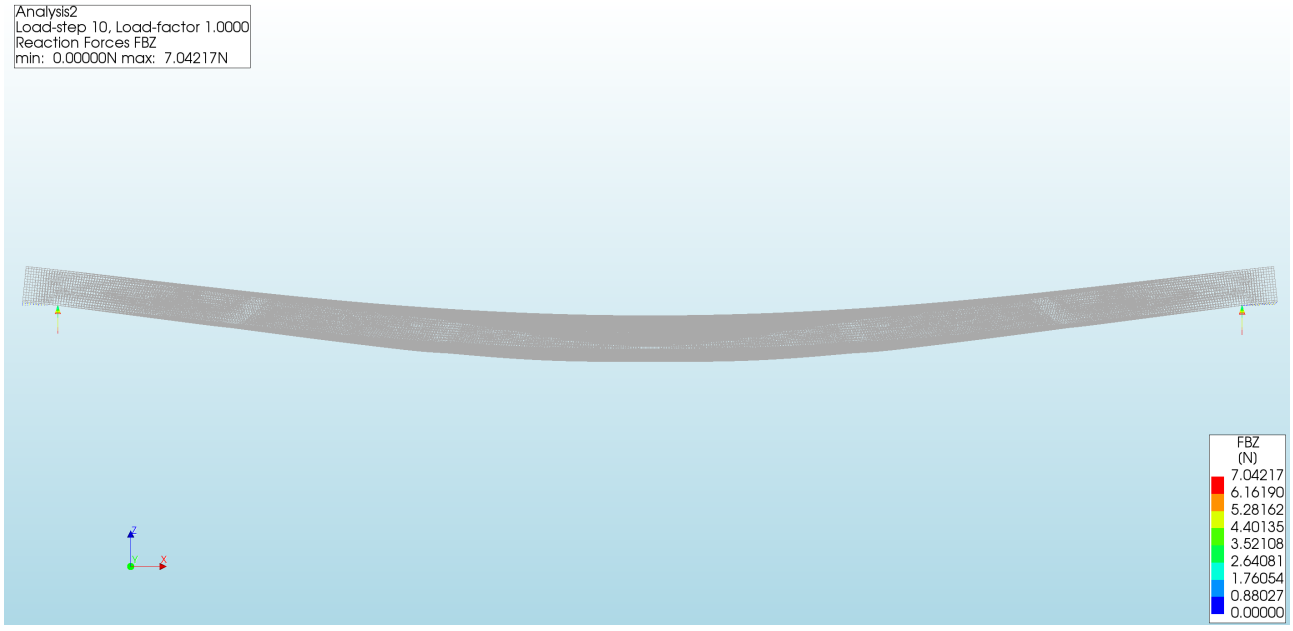


Figure 314: Reaction forces at last load step.

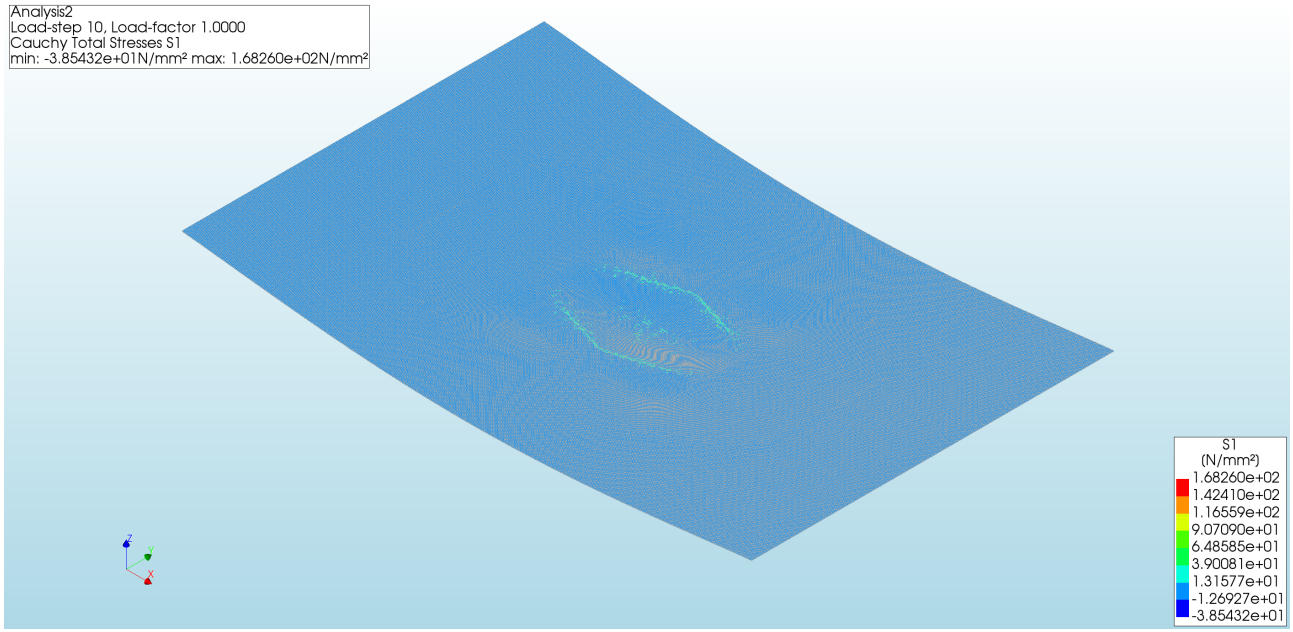


Figure 315: Cauchy principal stress S1 of the top glass layer.

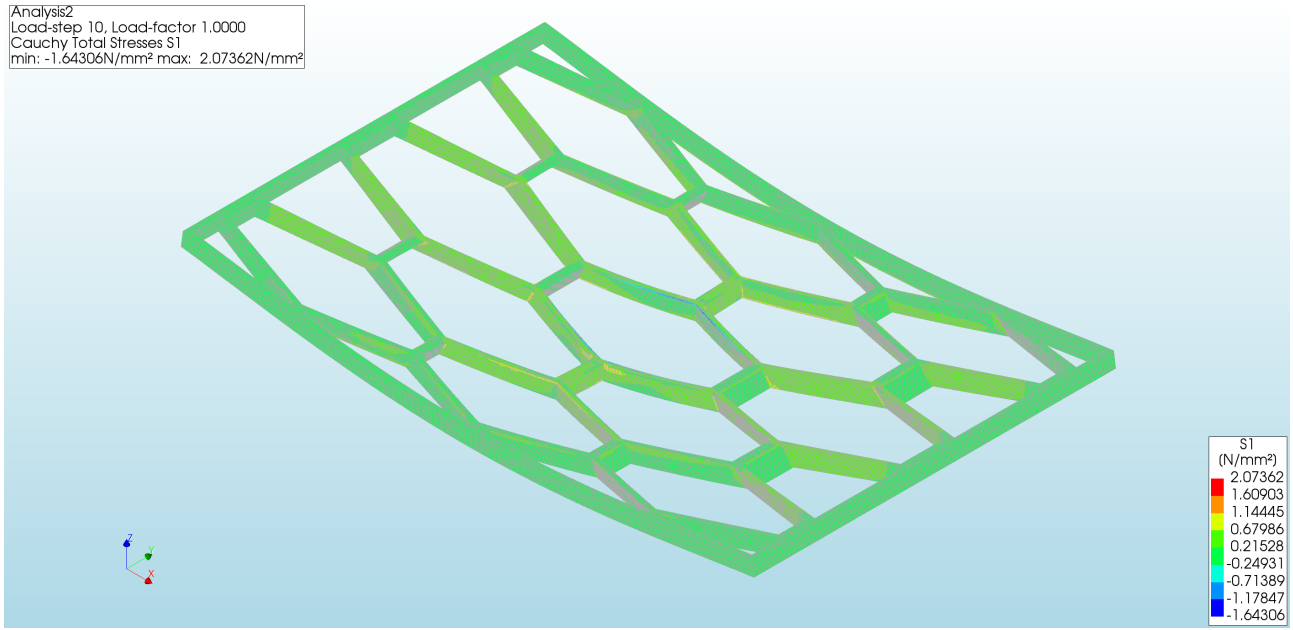


Figure 316: Cauchy principal stress S1 of the core layer.

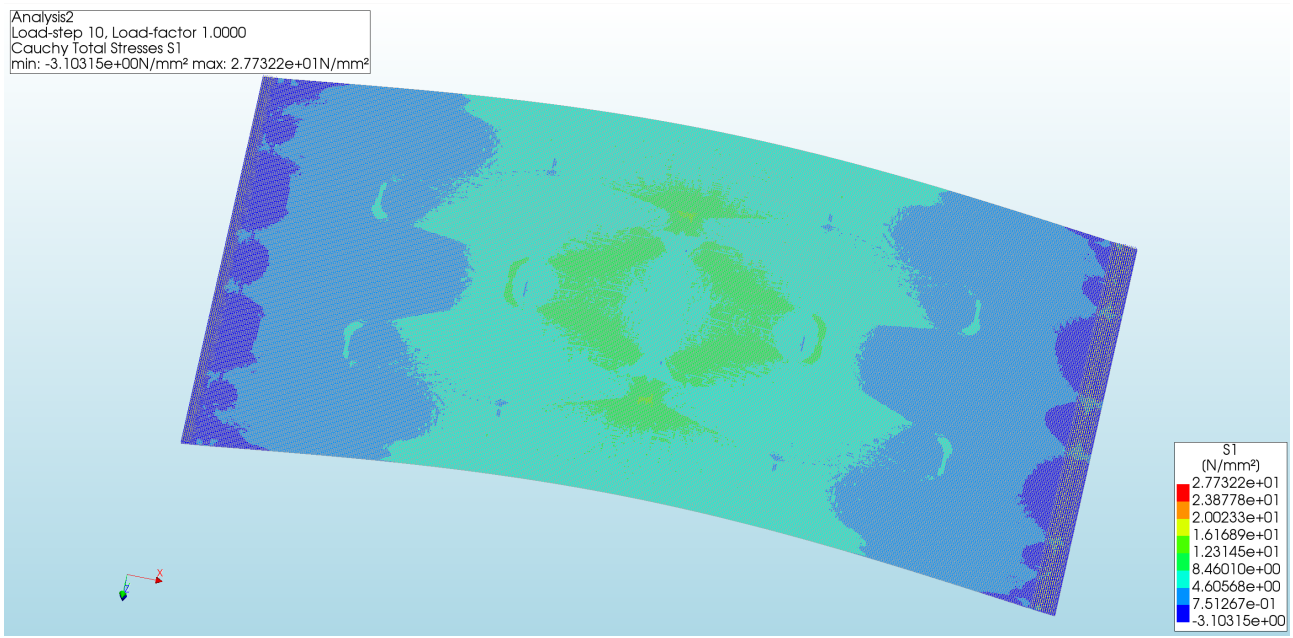


Figure 317: Cauchy principal stress S1 of the bottom glass layer.

Rotated pattern, 2 mm element size

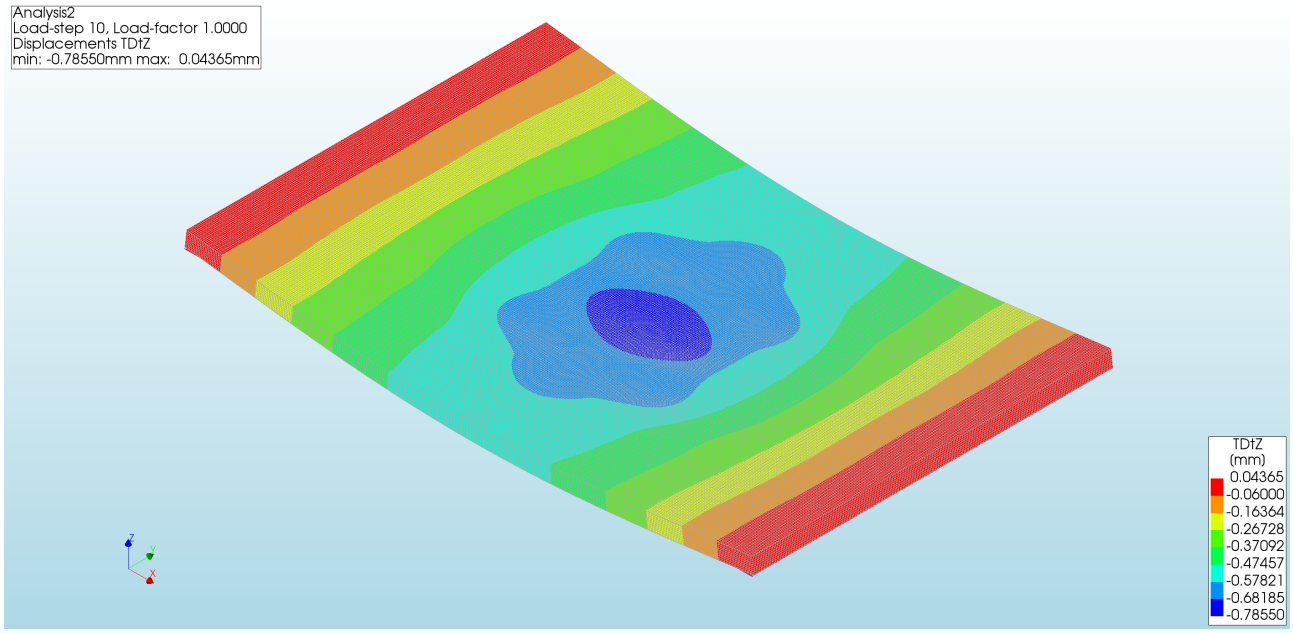


Figure 318: Displacements at last load step seen with top view.

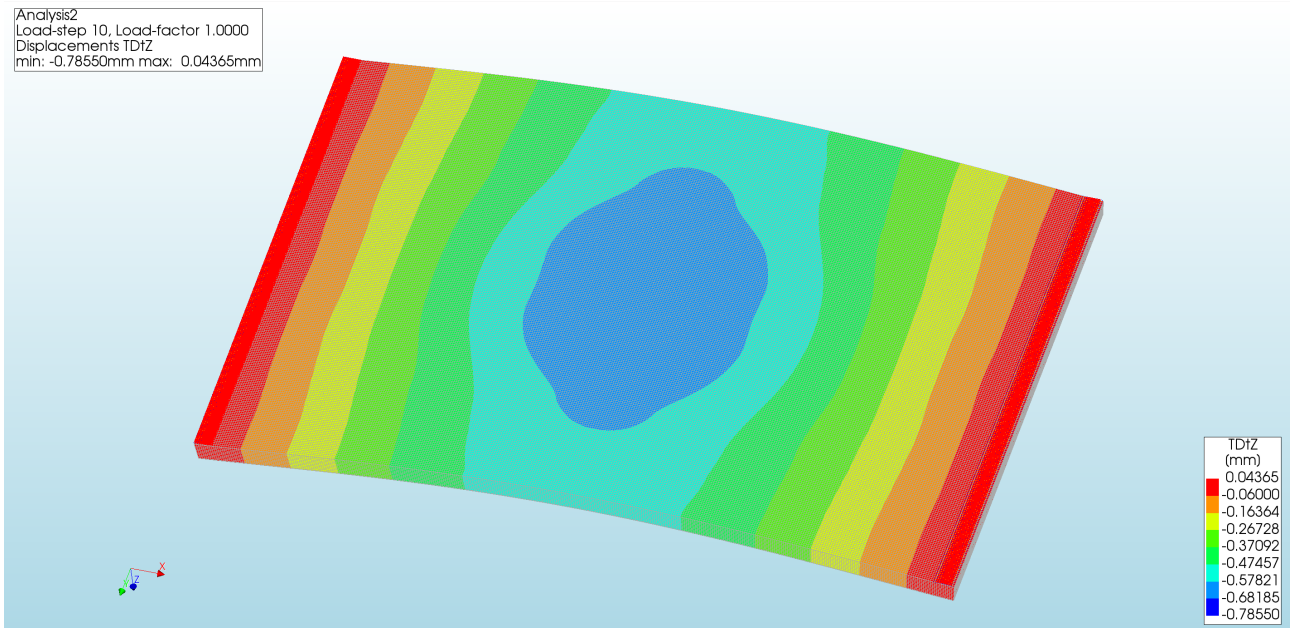


Figure 319: Displacements at last load step seen with bottom view.

Rotated pattern, 3 mm element size

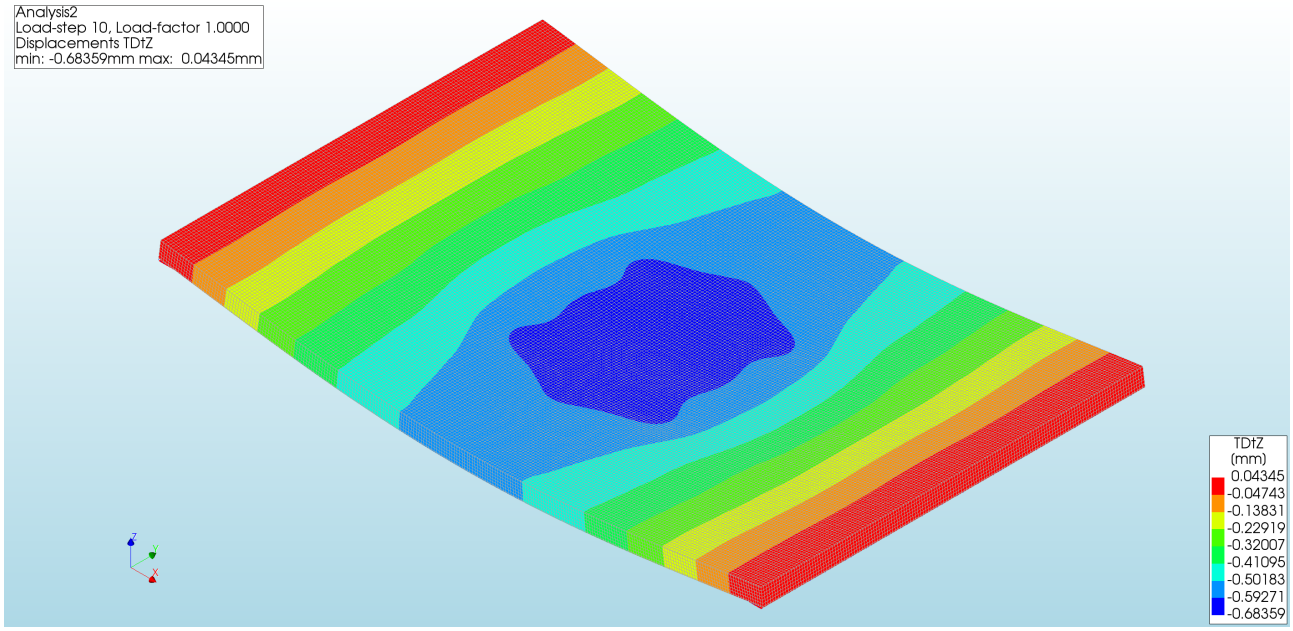


Figure 320: Displacements at last load step seen with top view.

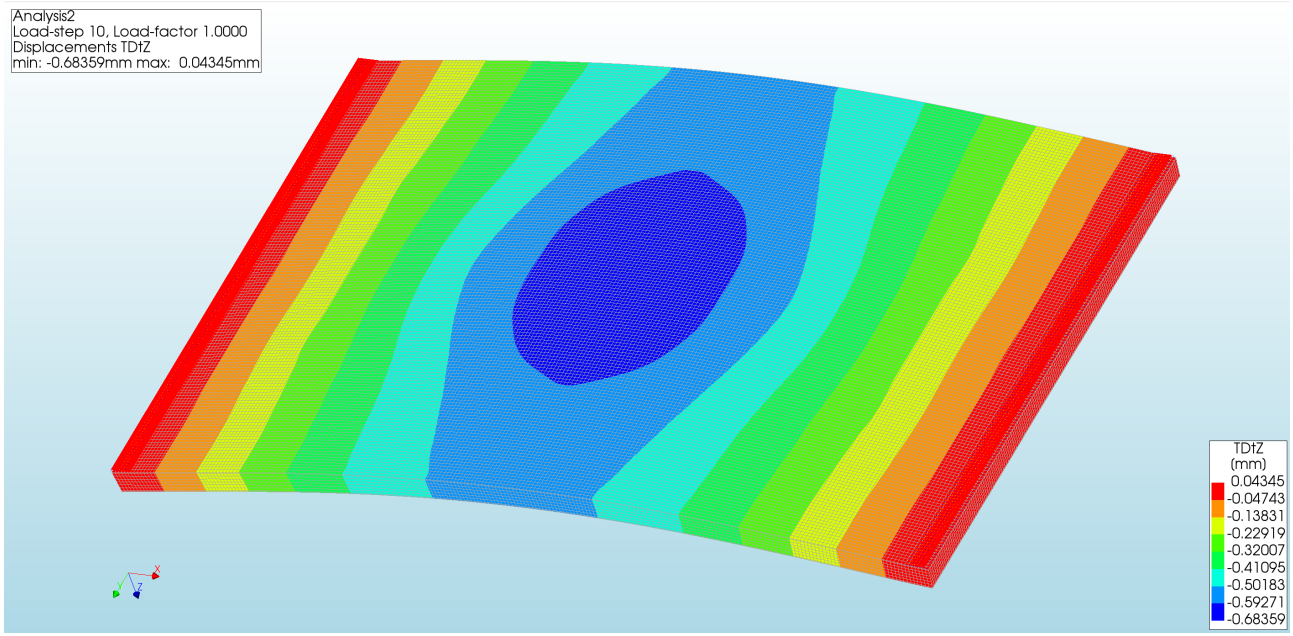


Figure 321: Displacements at last load step seen with bottom view.

Additional tests

Additional tests related to the core only

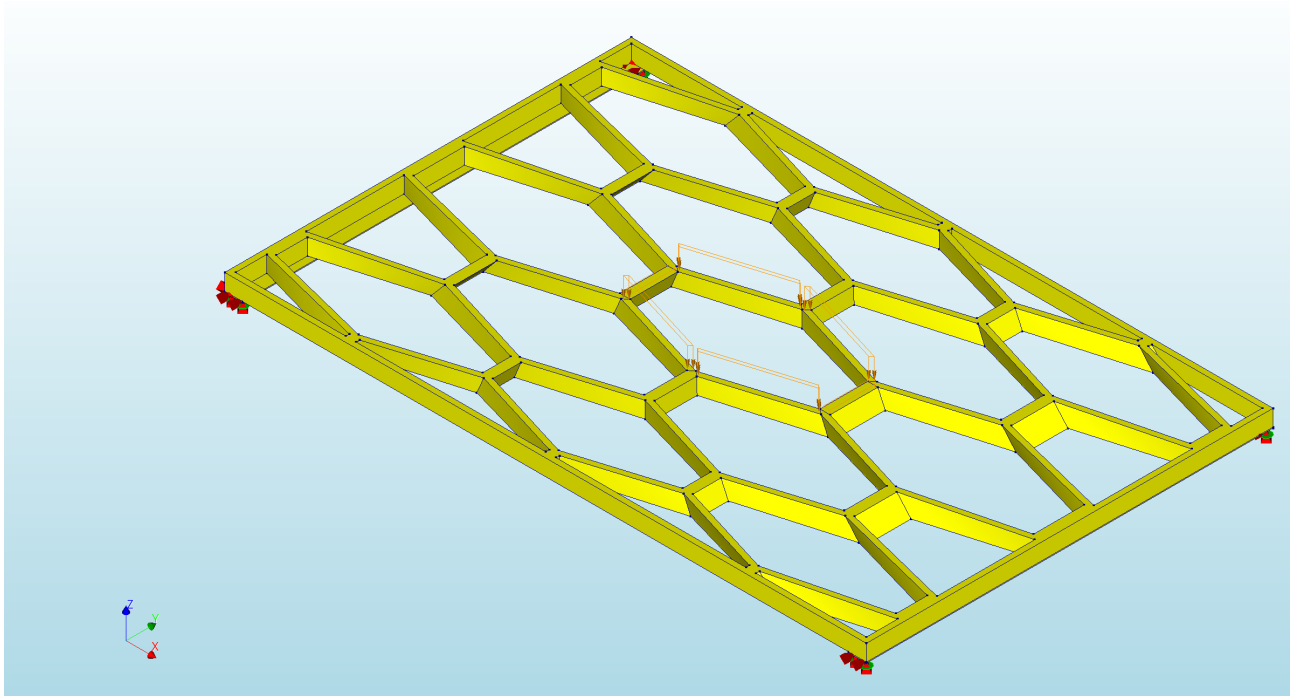


Figure 322: The Diana models of the rotated pattern

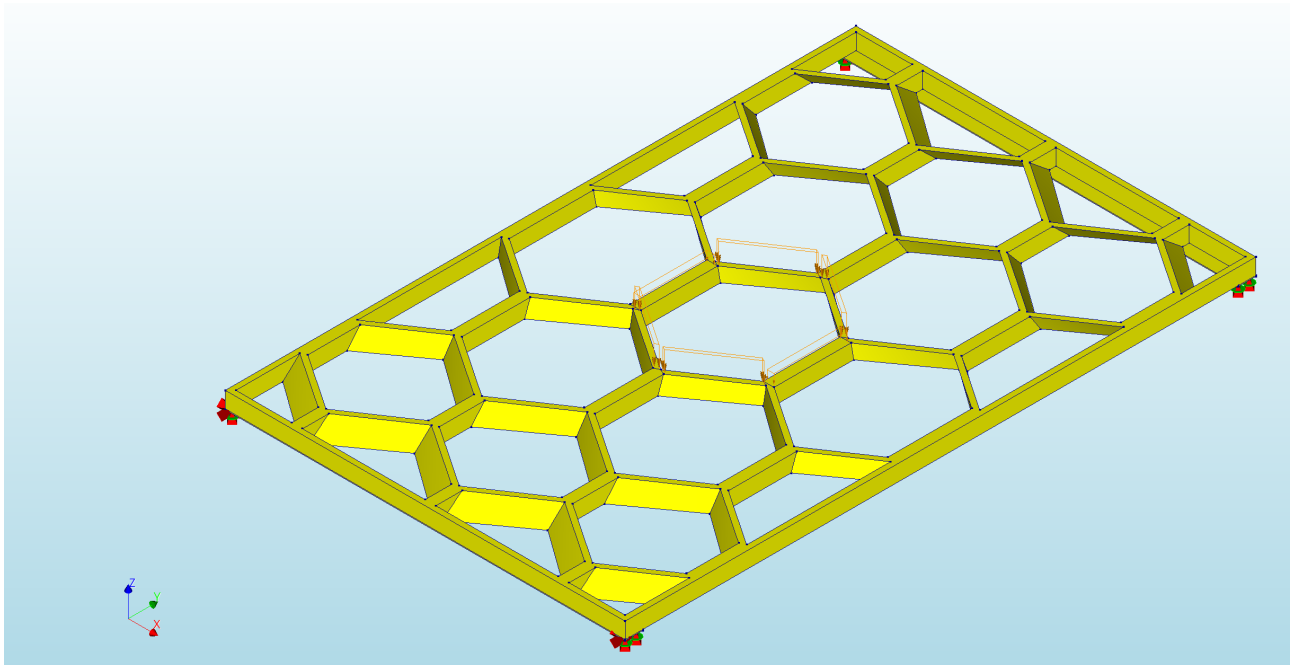


Figure 323: The Diana models of the non-rotated pattern

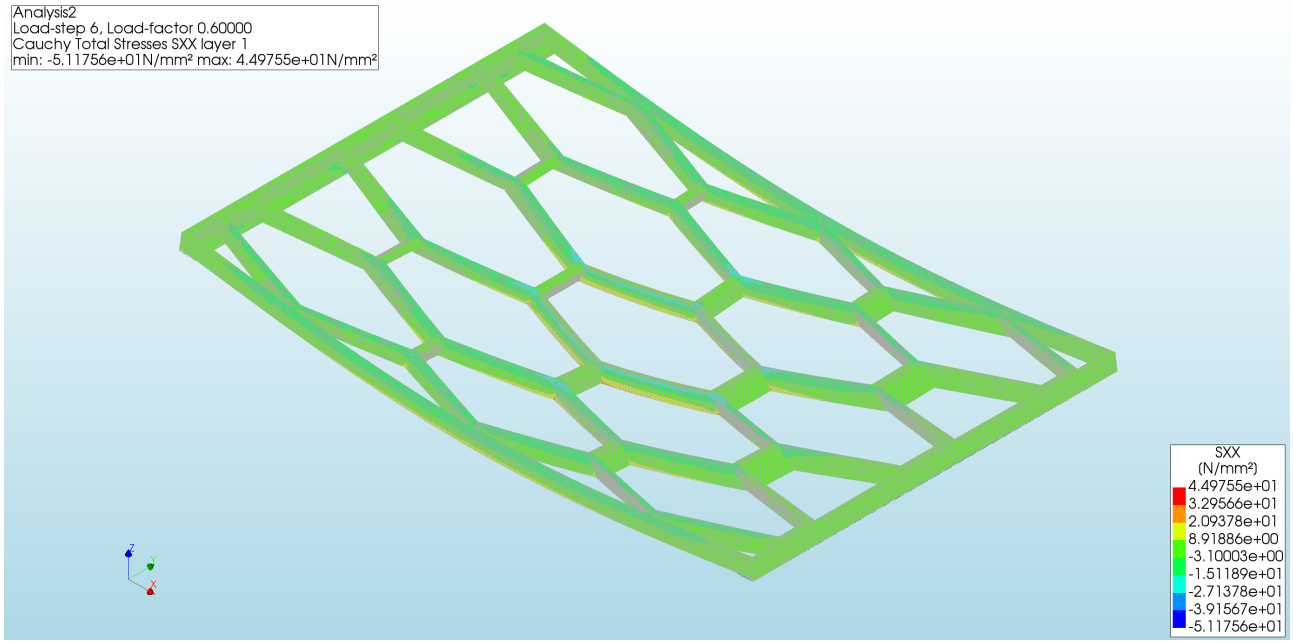


Figure 324: The S_{xx} of the rotated pattern with 3 mm elements at the load step that first reached the stress limit.

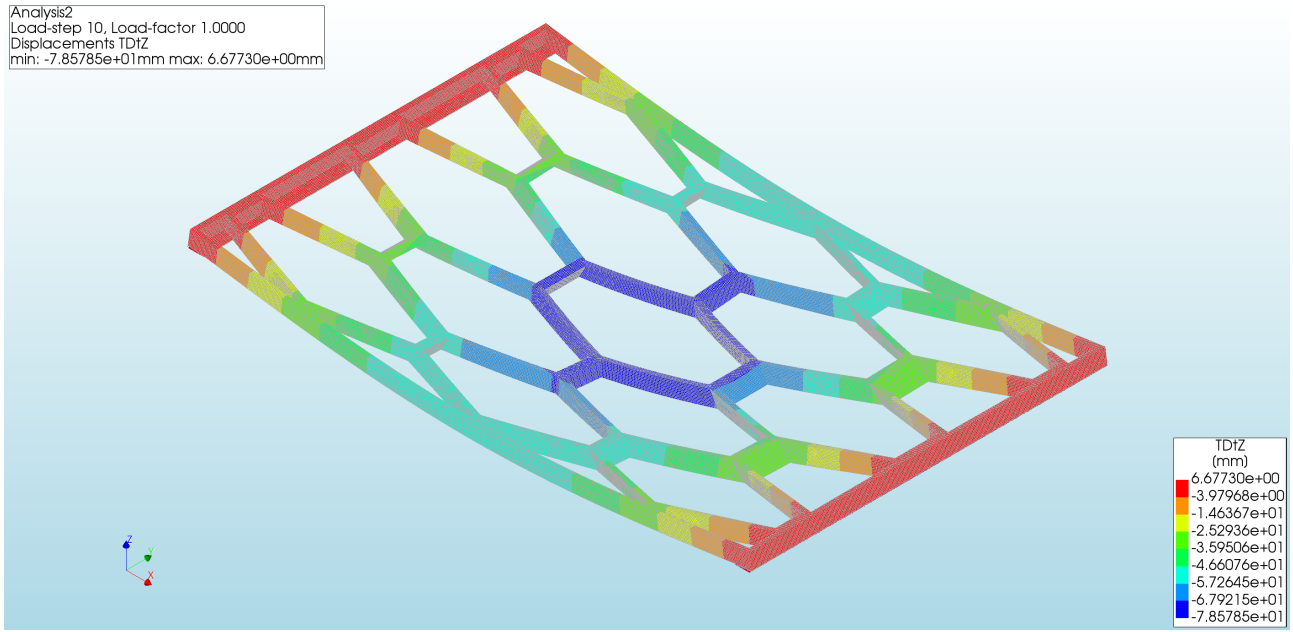


Figure 325: The displacements of the rotated pattern with 1.5 mm elements.

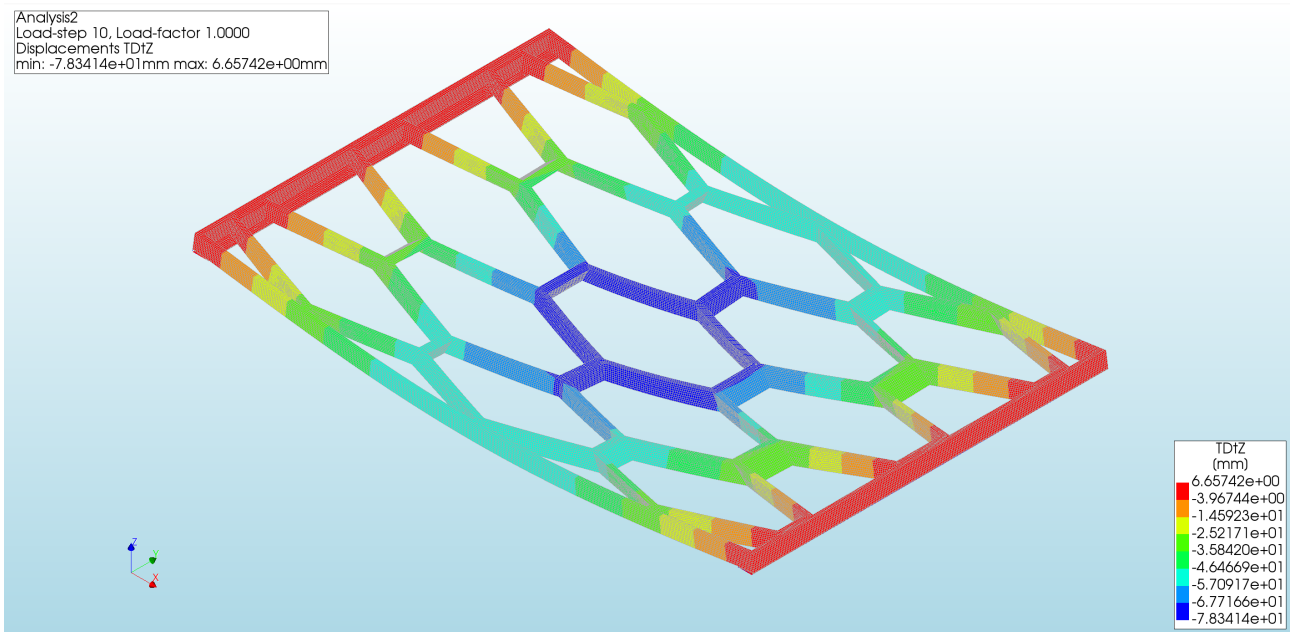


Figure 326: The displacements of the rotated pattern with 2 mm elements.

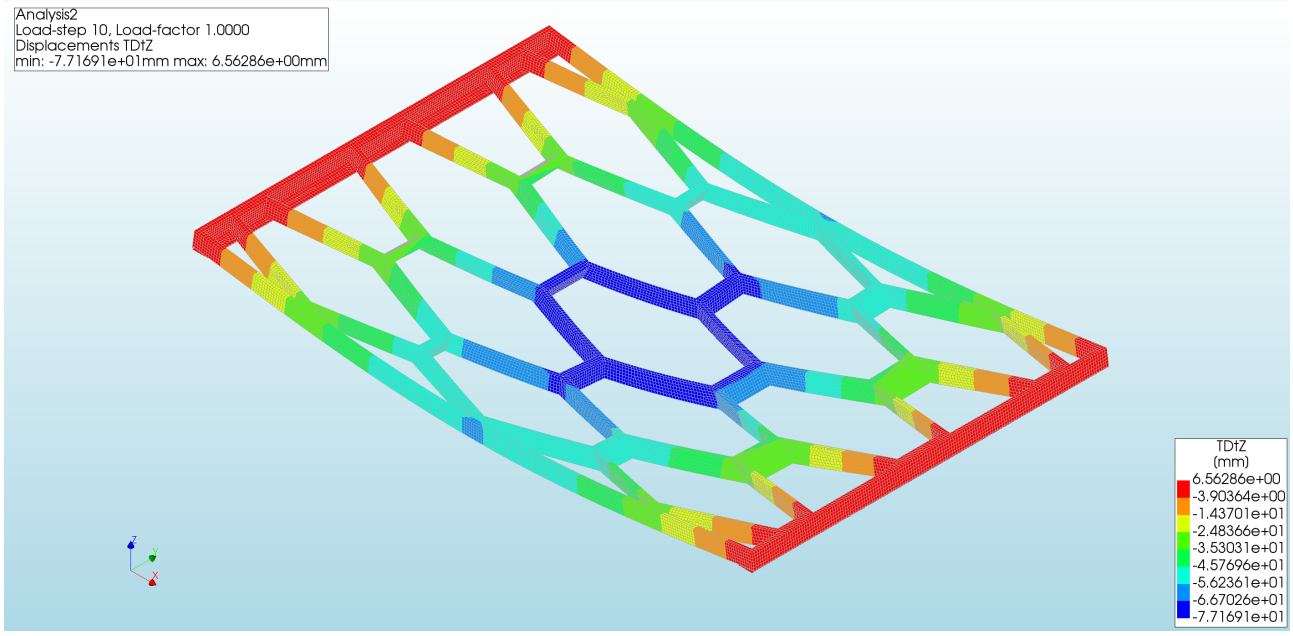


Figure 327: The displacements of the rotated pattern with 3 mm elements.

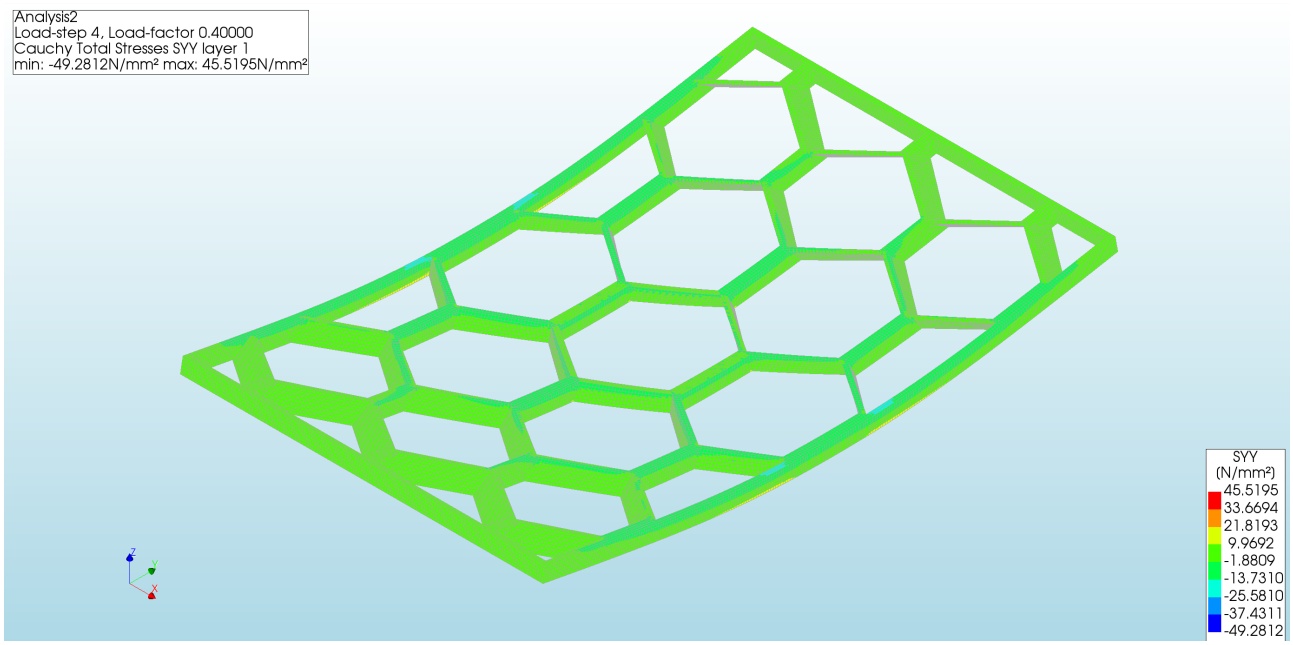


Figure 328: The S_{yy} of the non-rotated pattern with 3 mm elements at the load step just below the stress limit.

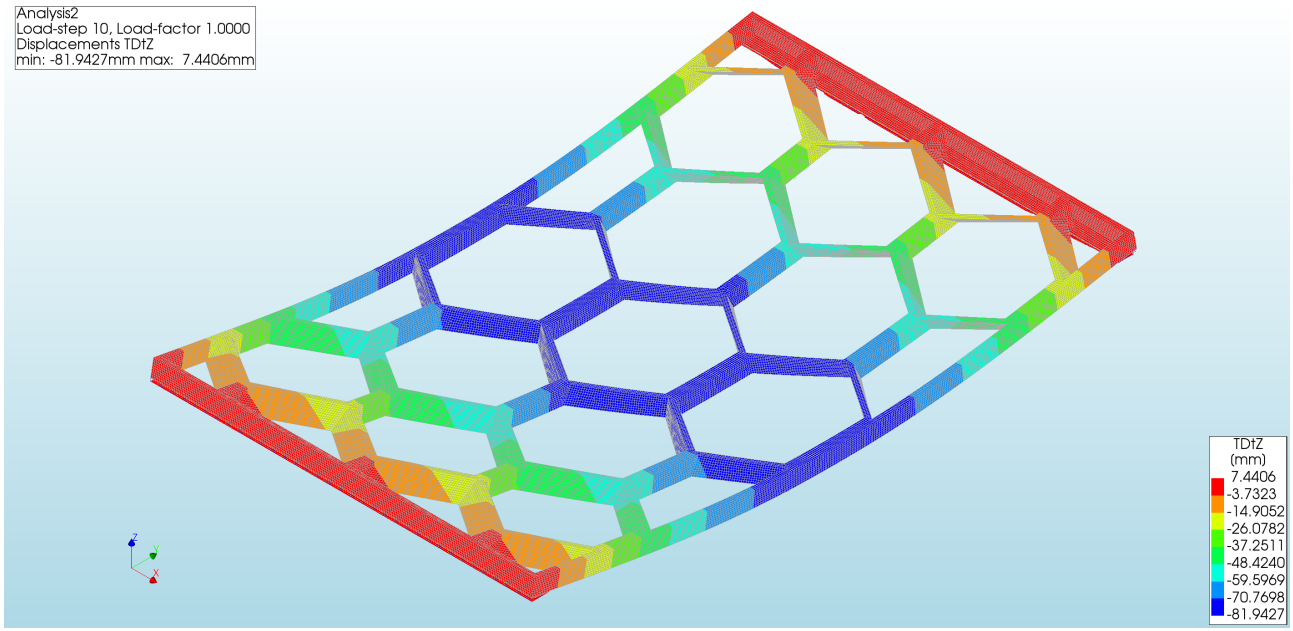


Figure 329: The displacements of the non-rotated pattern with 2 mm elements.

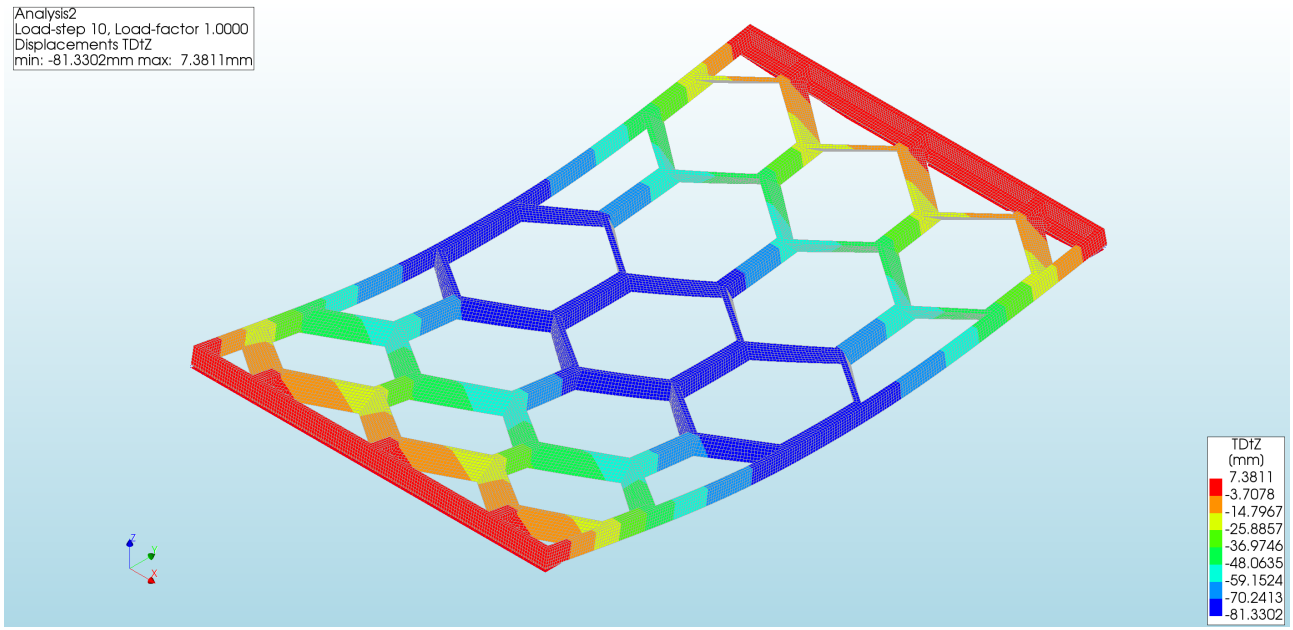


Figure 330: The displacements of the non-rotated pattern with 3 mm elements.

Additional tests: Diana vs analytical calculations

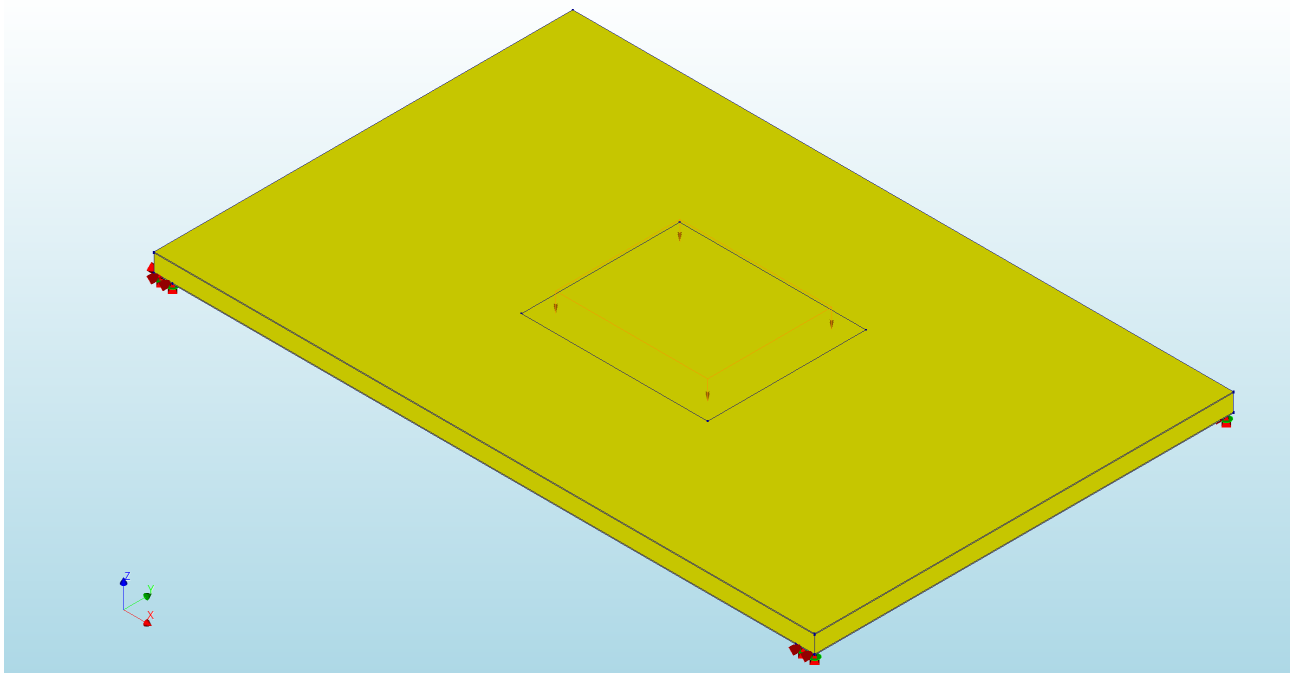


Figure 331: The geometry of the glass panel with boundary interface elements.

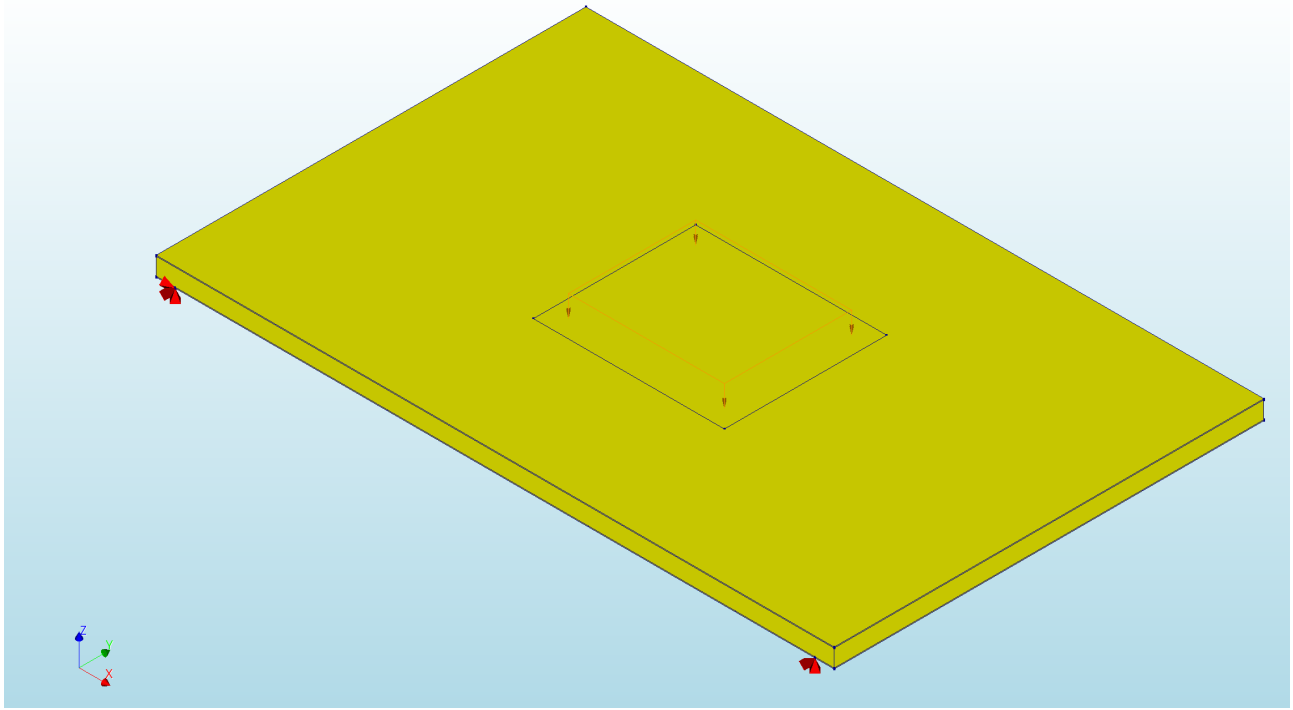


Figure 332: The geometry of the glass panel with line supports.

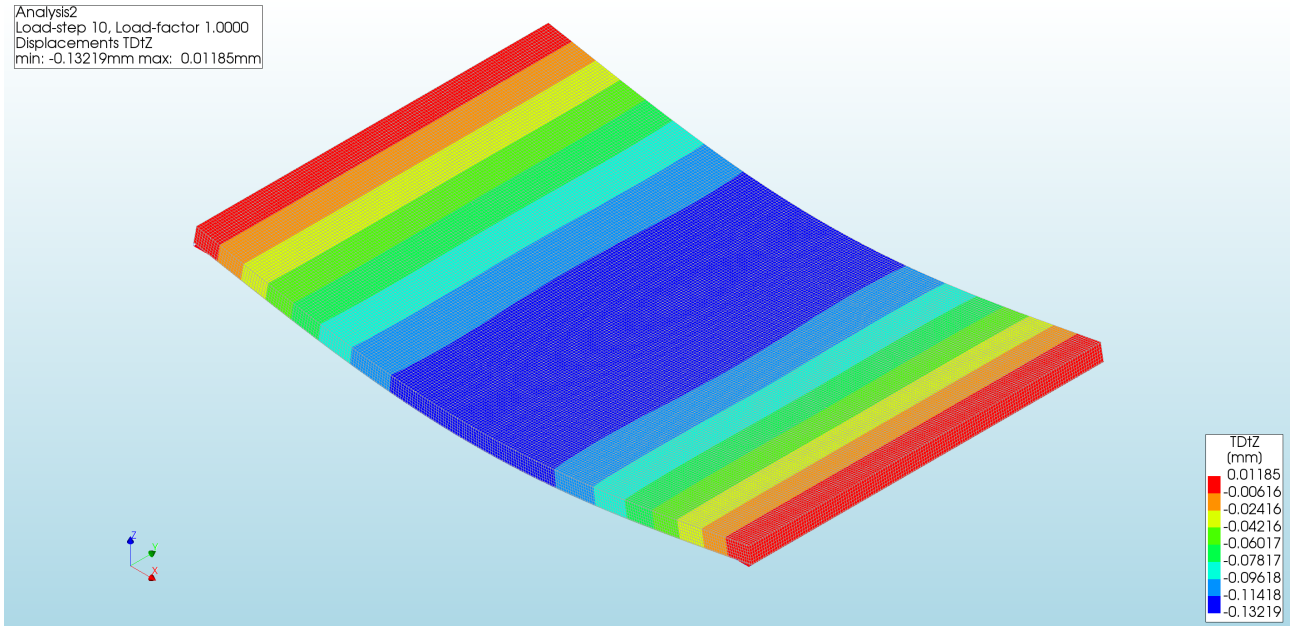


Figure 333: The deflection of the glass panel with boundary interface elements and 3 mm elements.

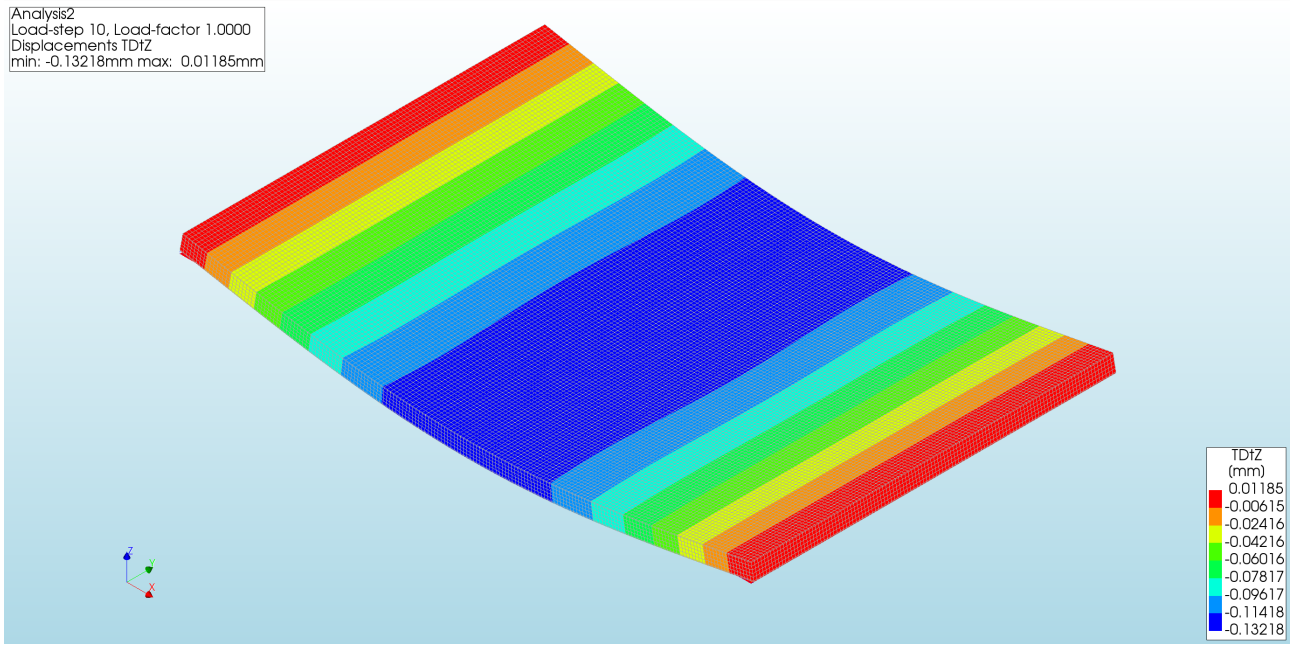


Figure 334: The deflection of the glass panel with boundary interface elements and 4 mm elements.

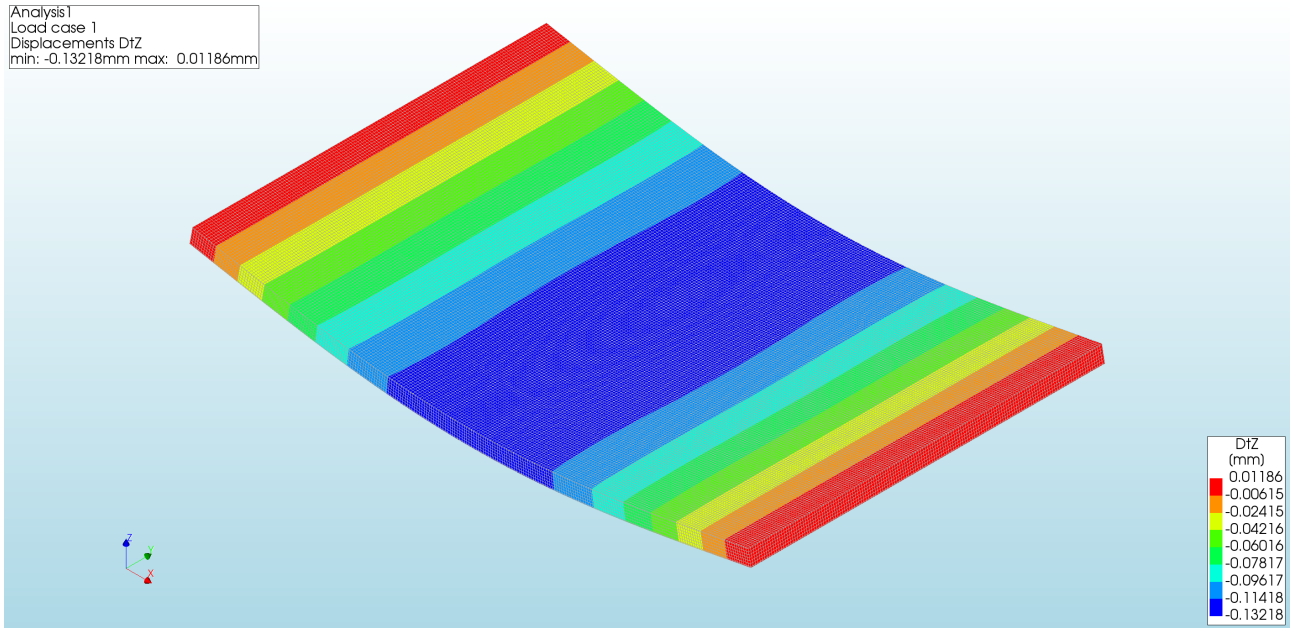


Figure 335: The deflection of the glass panel with line supports and 3 mm elements.

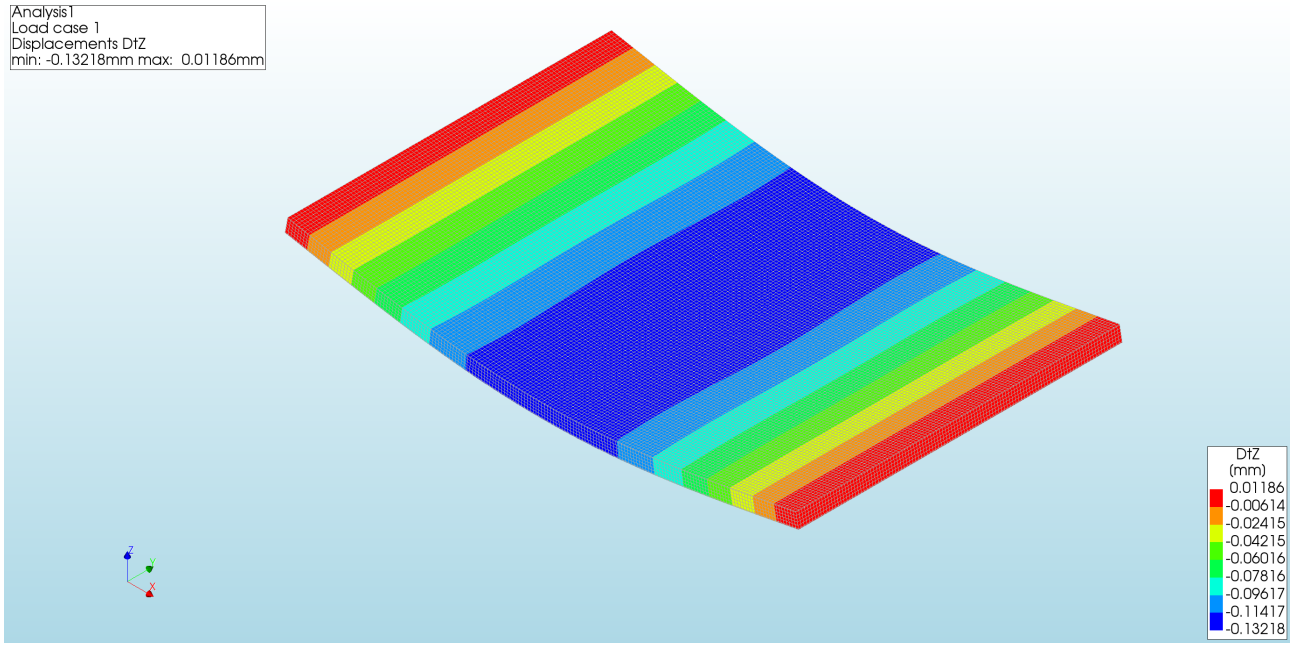


Figure 336: The deflection of the glass panel with line supports and 4 mm elements.

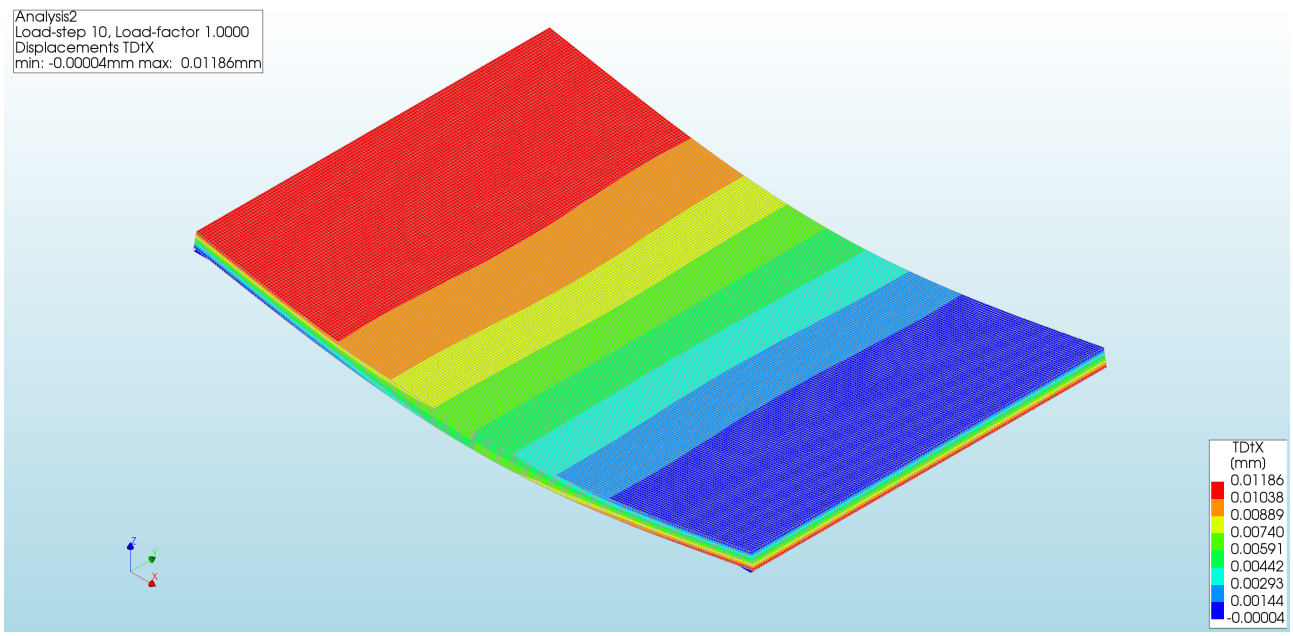


Figure 337: The horizontal displacement of the glass panel with boundary interface elements and 3 mm elements.

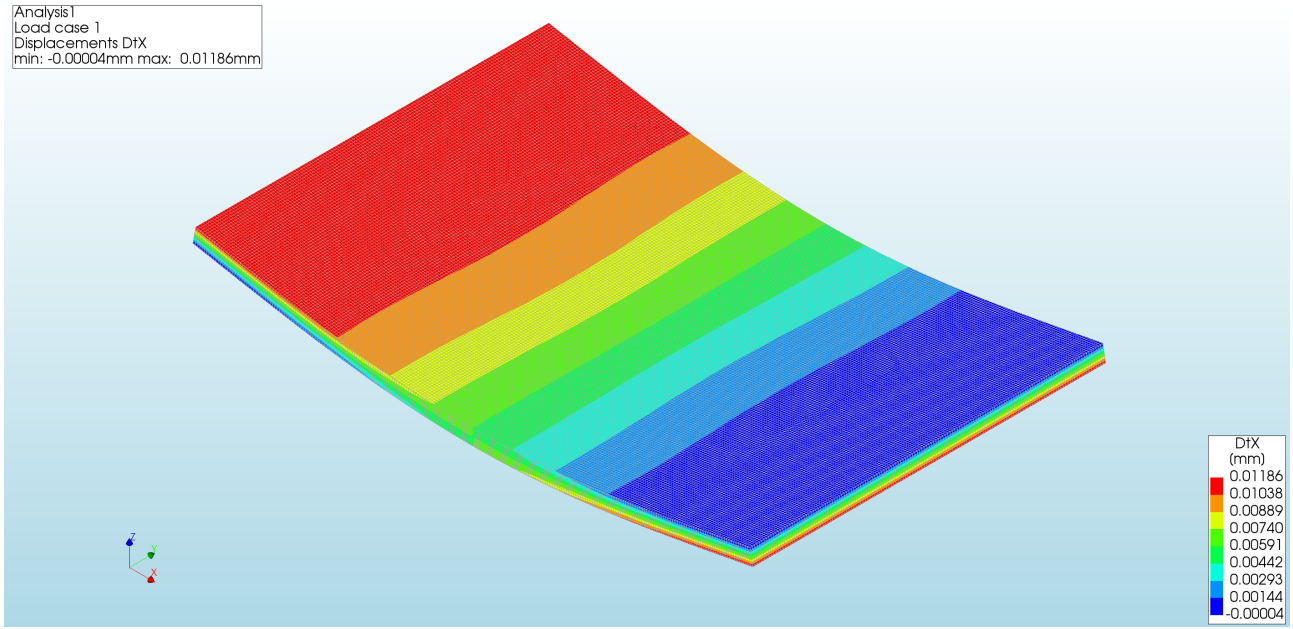
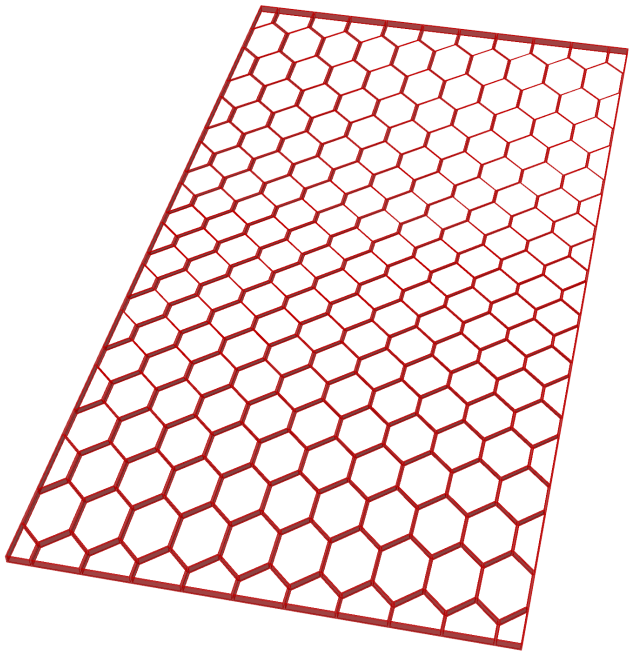


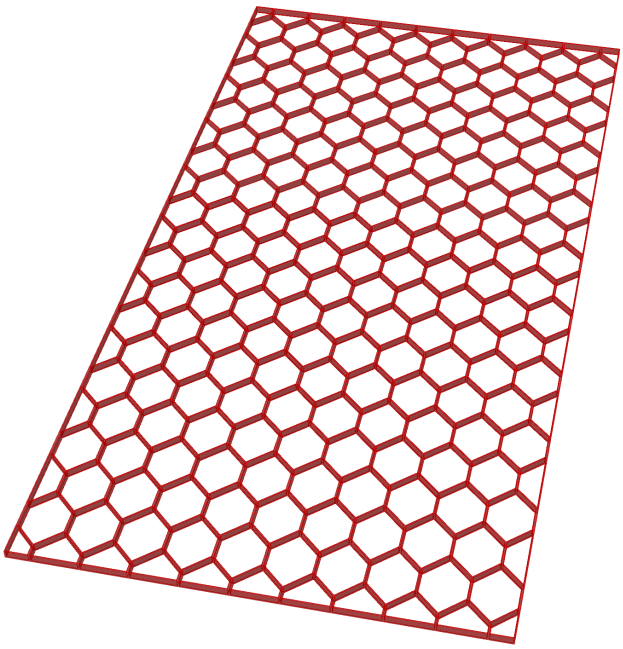
Figure 338: The horizontal displacement of the glass panel with line supports and 3 mm elements.

Appendix J: Case Study

Designs normal orientation

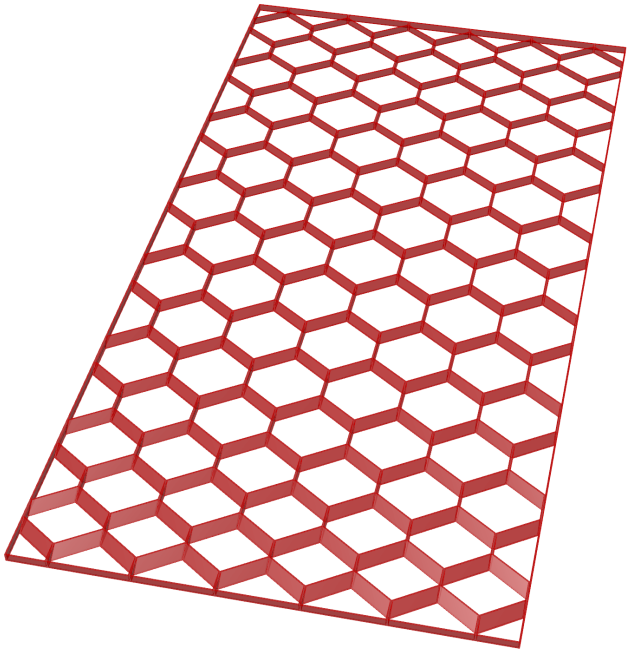


(a) Least optimal for transparency criterion

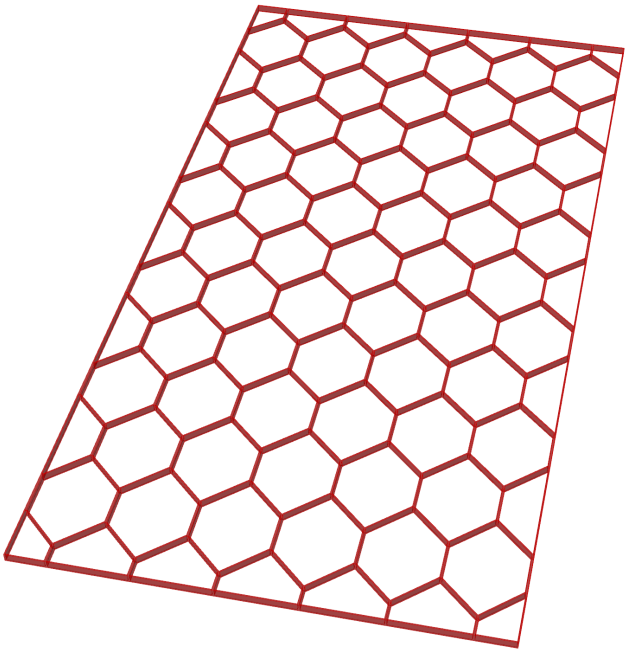


(b) Least optimal for insulation criterion

Figure 339: Overview of designs normal orientation.



(a) Least optimal for daylight criterion



(b) Test 5

Figure 340: Overview of designs normal orientation.

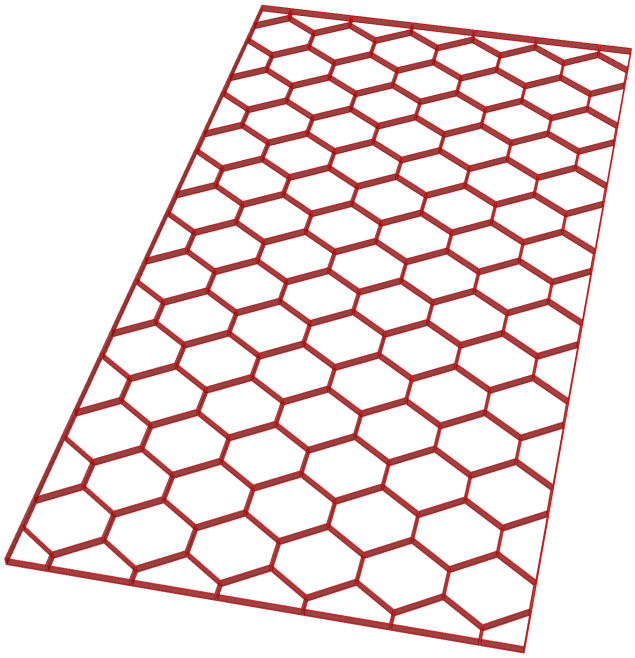
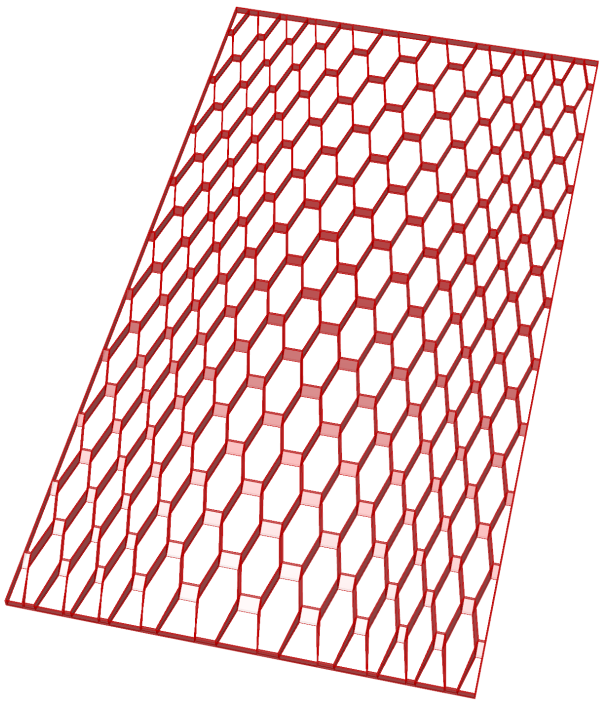
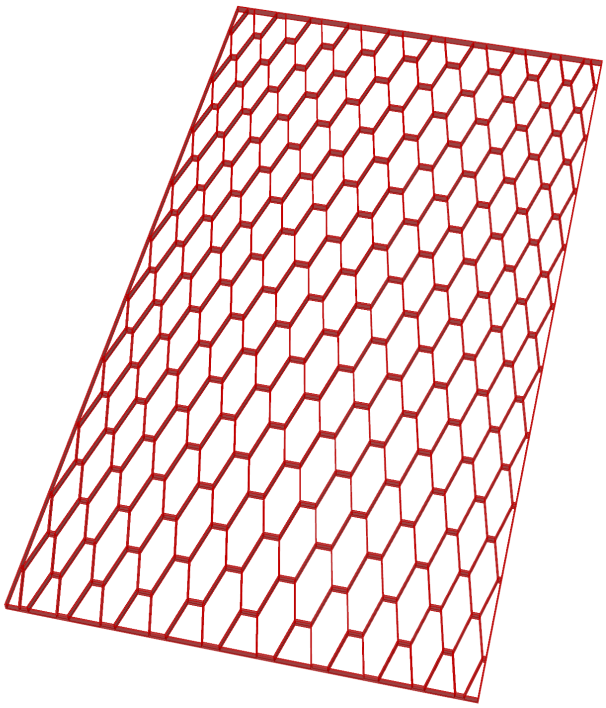


Figure 341: Test 16

Designs rotated orientation

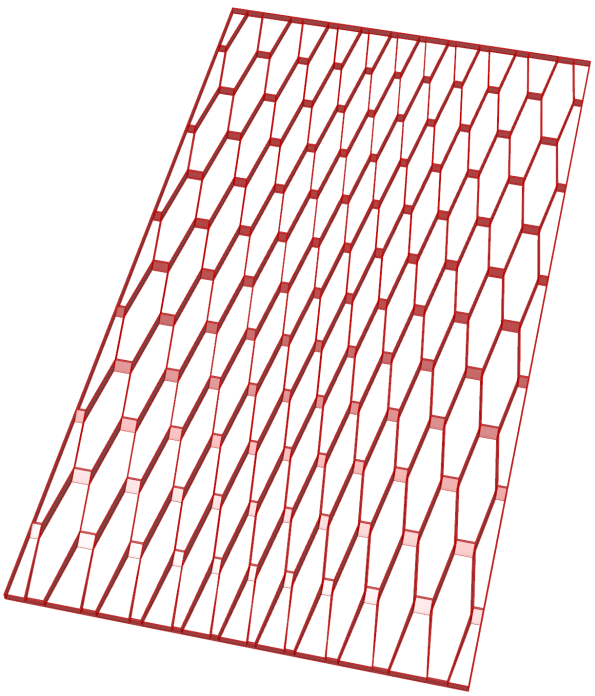


(a) Least optimal for transparency criterion

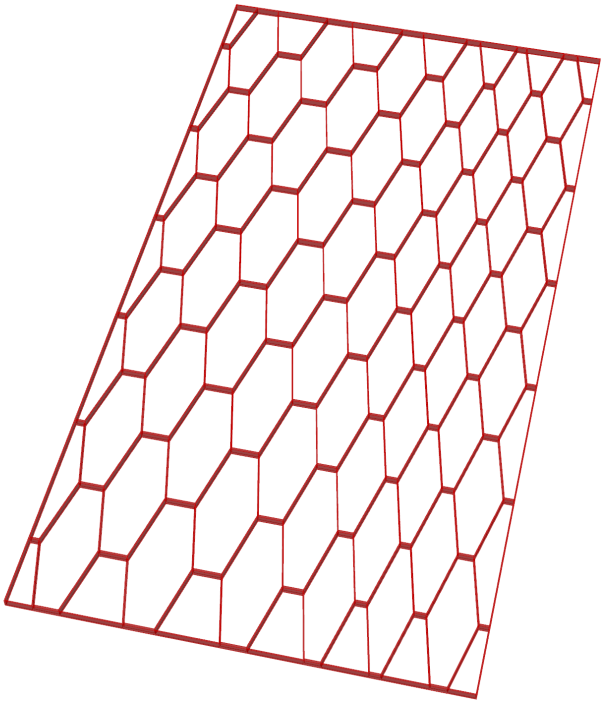


(b) Least optimal for insulation criterion

Figure 342: Overview of designs rotated orientation.

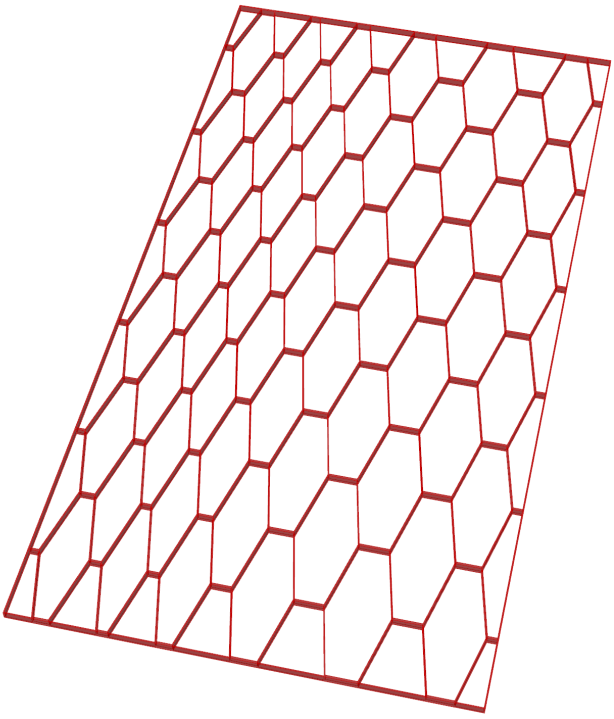


(a) Least optimal for daylight criterion

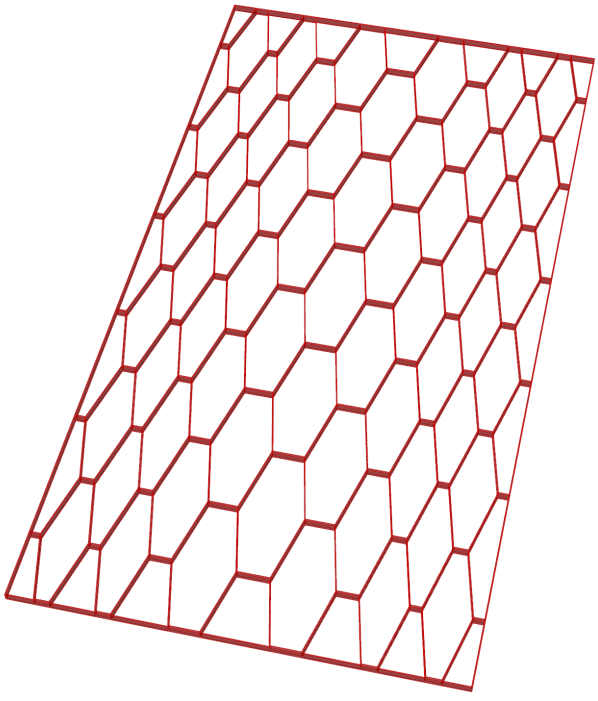


(b) Test 4, 6, and 10

Figure 343: Overview of designs rotated orientation.



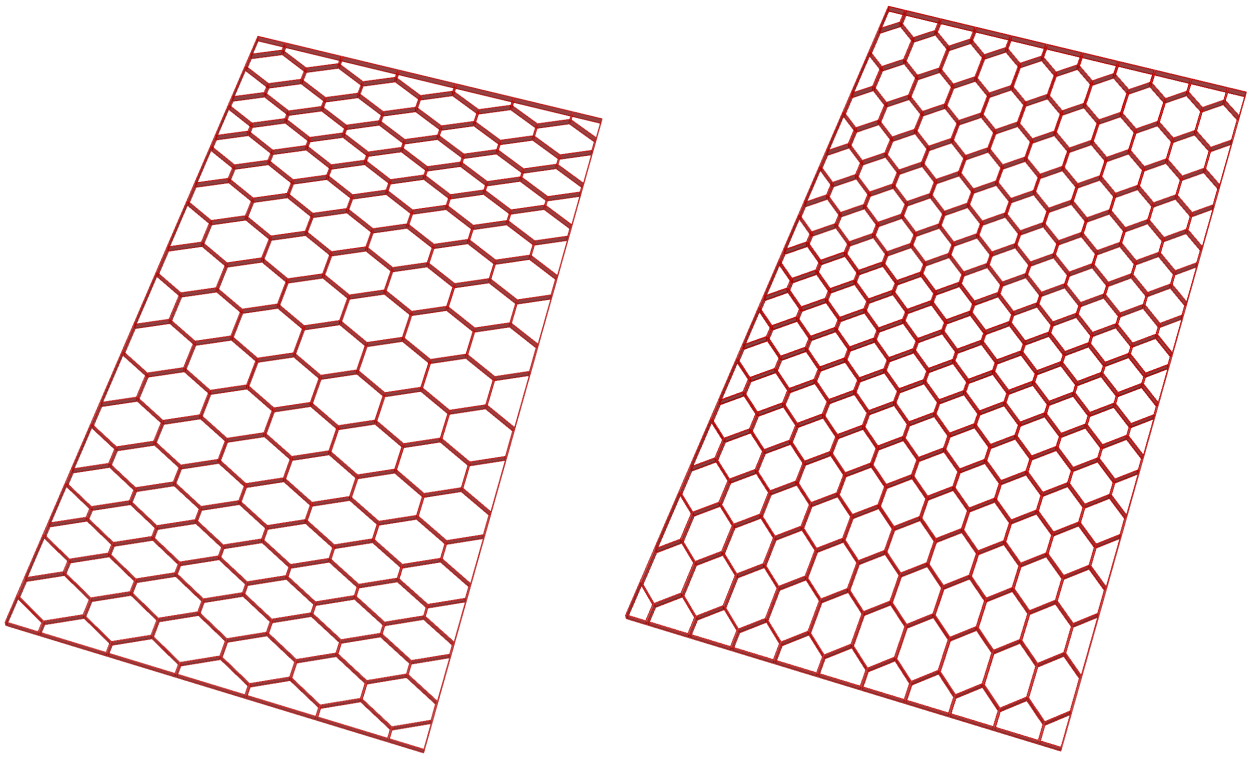
(a) Test 8



(b) Test 9

Figure 344: Overview of designs rotated orientation.

Sensitivity analysis



(a) NR2012A (b) NR2020

Figure 345: Overview of designs.

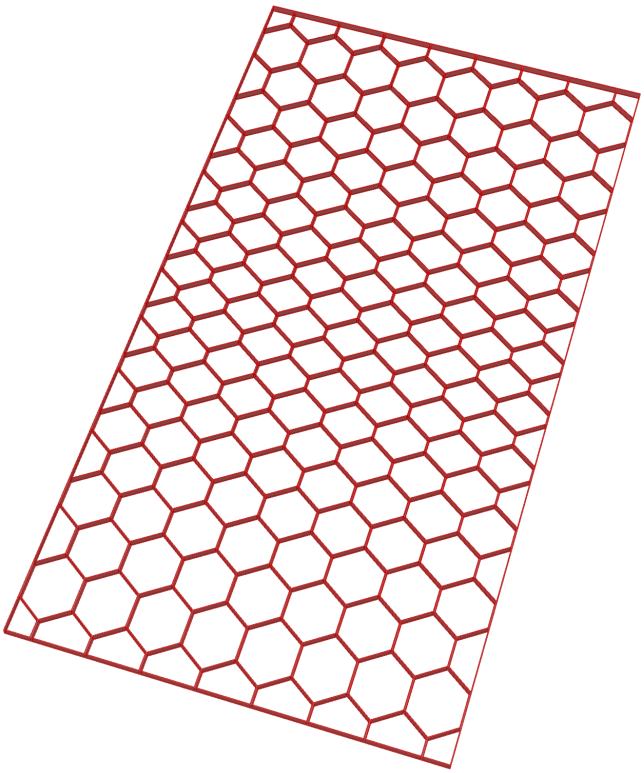
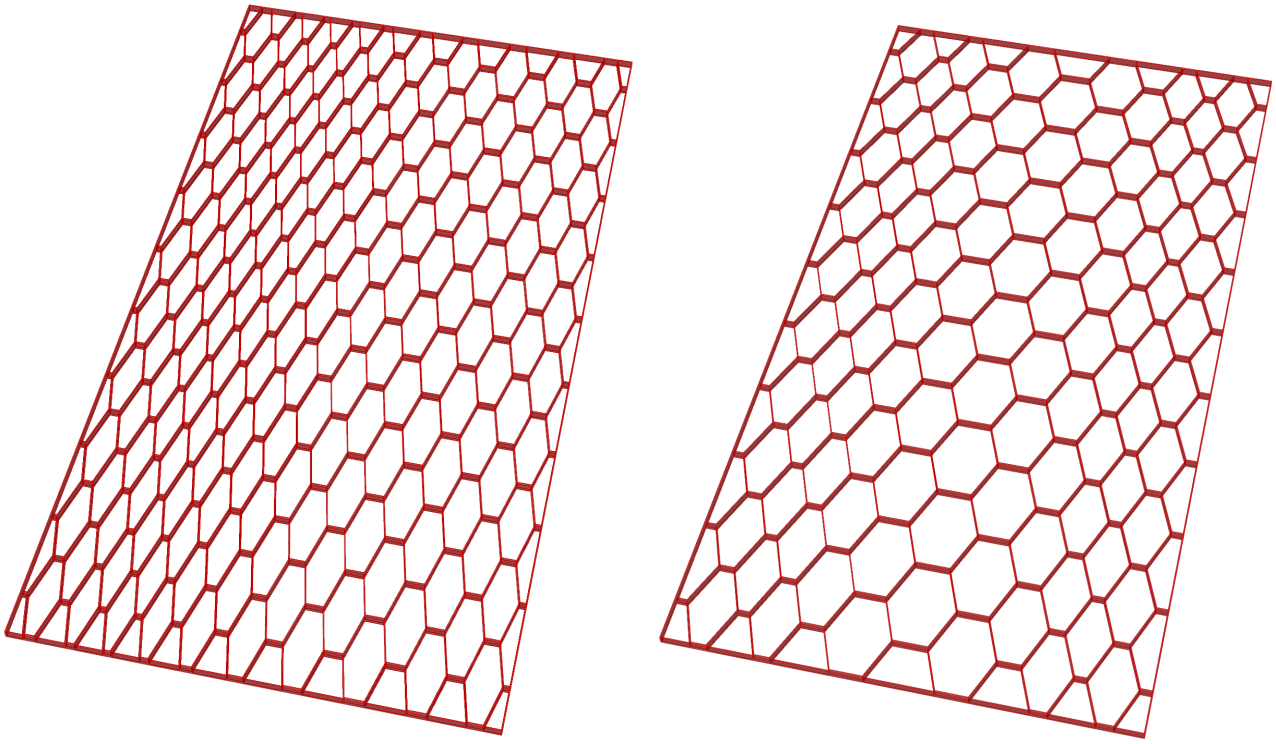


Figure 346: NR2016



(a) R2020 (b) R2012D

Figure 347: Overview of designs.