

An algorithmic design method for structural glass roof structures

Research into the technical viability

Master's thesis

A.M. Vos

July 2017

Student

A.M. Vos

Student ID: 1512749

Email: aurinke.vos@gmail.com

Educational institution

Delft University of Technology	Stevinweg 1
Faculty of Civil Engineering and	2628 CN Delft
Geosciences	
Building Engineering	The Netherlands

Graduation committee

prof. ir. R. Nijse	Delft University of Technology
dr. ir. J.L. Coenders	White Lioness technologies
dr. ir C. Louter	Delft University of Technology
ir. F.J. van Gijn	Van Rossum Raadgevende Ingenieurs b.v.
ir T. Bristogianni	Delft University of Technology

Preface

This Master 's thesis was written as part of the curriculum of Building Engineering with a specialization in Structural Design, a track of the Master of Science in Civil Engineering at the Delft University of Technology. The research was conducted in collaboration with the faculty of Civil Engineering and Geosciences, the BEMNext lab and Van Rossum Raadgevende Ingenieurs.

I would like to take this opportunity to express my gratitude to everyone who helped me completing this thesis. Starting with the members of my graduation committee, prof. ir. Rob Nijsse, who supervised the general progress and organization of the process, dr. ir. Jeroen Coenders, dr. ir. Christian Louter, ir. Fokke van Gijn and ir. Telesilla Bristogianni, for their knowledge, feedback and inspiration. In particular, I would like to thank dr. ir. Jeroen Coenders for the guidance throughout the entire process, starting as faculty member of the faculty of Civil Engineering and Geosciences and continued to do so voluntarily.

Next, I would like to thank Van Rossum for providing the necessary resources, pleasant work environment and flexibility regarding the work hours.

Subsequently, a special thanks goes to my father, Edward Vos, who had an essential contribution throughout the entire process and improved the thesis with his editorial comments. I also would like to express my deepest gratitude and appreciation to my mother, friends and family for their support and patience during this busy period. Last but not least, I would like to thank Ruben Stokx, the love of my life, for his encouragement, faith, immense patience and support.

*Aurinke Vos
Rotterdam, July 2017*

Summary

'This thesis arises from my fascination for glass structures, the ambition to bridge the gap between architects and engineers and the awareness of the potential of parametric design.'

Where traditionally the architect functions as an entire design team, nowadays we see that the complexity of design situations calls for division of disciplines and expertise. With this separation, the need for integral design and decision making becomes larger. An algorithmic design format might provide the means to integrate the knowledge of different experts at the start of the process, where design choices have the largest impact on all aspects ranging from aesthetics to costs.

With structural glass design it is important to bear in mind the different possibilities and limitations of the structure from the beginning of the design as these have a huge influence on the detailing and dimensioning of the elements. Therefore, it seems to be an excellent opportunity to assign the development of an algorithmic design methodology to the structural glass designs processes, to explore the possibilities within the boundaries. The research conducted in this thesis, focuses on the development and application of an algorithmic computational design tool for the conceptual design phase.

The prototype of this model focuses on all glass roof cover designs. The thesis is composed of 9 chapters, dealing with various aspects of the research process and to explore the possibilities of parametric tool design and the applicability to structural glass covers. Chapter one introduces the research problem: it describes the motive and driving force behind this thesis with a plan of approach. The chapter is divided in three: Part one describes the introduction to the problem. Part two states the motivation, research question and objectives. Part three shows the methodology to compose the thesis.

Chapter two provides a brief overview of the literature review regarding structural glass, and shows its limitations and possibilities. Chapter three is an important chapter, that illustrates the strategy for the tool development. The knowledge gained during the literature study is used to provide background information on algorithmic design, to discuss the characteristics of the initial design phase and to formulate the requirements for functionality, usability, reliability and performance, that should be met by the methodology. To increase the legibility of parametric schemata a flow chart that represents the working of the tool is divided into several modules: the boundary conditions, base geometry, structural properties, load generator, structural optimization, structural analysis and the output report. In order to create a tool template, that functions as a solid foundation for further development, this chapter generates a strategy that exceeds the scope of this research.

Chapter four implements the theoretical framework and flow chart, composed in the strategy chapter, to create a functioning prototype. The tool is evolved by test driven development and is pushed until the basic function of the modules are working. Subsequently the modules were expanded to become more comprehensive. The tool enables the user to assign a project specific perimeter, generate a large variety of design alternatives. and specify standardized values. The tool provides automatic allocation of cross section, meeting the requirements checked with the structural analysis.

Chapter five is written to examine the verification of the tool, in order to check that the modules are in line with the requirements and physical laws. This chapter discusses whether the tool is functioning as supposed to, whereas Chapter 6 discusses whether the right product is produced. This is tested by means of a case study at the Grote Marktstraat in The Hague, where the unnoticeable entrance of a subway station is provided with a glass roof cover. During this case study, the applicability of a scenario like this with a span of more than 30 meters is explored. Several possibilities of the tool are investigated. It is shown that with the simple input of the boundary conditions, it is easy to create all kinds of designs in an instant. However, with completion of the design process it becomes clear that the prototype needs the proposed adjustments to be a guidance tool for the structural glass design process and to provide a better understanding of the consequences of certain design decisions. Since the prototype is designed, keeping the bigger picture in mind, this chapter also tests the feasibility of the development. This showed that the script can be easily adjusted, by someone familiar with the original script and visual programming method and

the tool enables the user to create an improved design, both structurally and aesthetically.

Chapter 7 discusses the results, which led to the Conclusions drawn in Chapter 8 that were used to compose the Recommendations. The test case shows potential as integral (structural) design tool. It is plausible that the tool can contribute to the everyday use of glass as structural material. The tool also has potential for further optimization of different aspects. The main aim of this thesis is to prove feasibility and the study shows that there is good potential for an algorithmic methodology for structural glass covers in the conceptual design phase. The results are also promising for other types of structures during the entire design process, unto production. Integration of use of other building materials in the tool as wood, concrete and steel seems feasible as well. However, user-based testing and development accordingly is recommended.

Contents

1	Introduction	13
1.1	Structural glass	13
1.2	Algorithmic design	14
1.3	Design process	14
1.4	Problem introduction	15
1.5	Research design	15
	Thesis objective	15
	Research question	16
	Research objectives	16
	Research methodology	16
	Thesis outline	17
2	Theoretical framework	19
2.1	Material properties	19
2.2	Structural safety	21
2.3	Manufacturing process	23
2.4	Conclusions	25
3	Strategy of tool development	27
3.1	Introduction	27
3.2	Software requirement specification	28
3.3	Framework	32
3.4	Flowchart	41
4	Tool development	45
4.1	Introduction	45
4.2	Test-driven development	46
4.3	Framework	49
	Graphical user interface (GUI)	49
	Script structure	52
	Output and visualization	72
	Workflow	73

4.4	Work in progress	74
4.5	Conclusions	77
5	Tool verification	79
5.1	Introduction	79
5.2	Testing methods	80
5.3	Modules	80
5.4	Conclusions	93
6	Tool validation and testing	95
6.1	Design	95
6.2	Algorithmic design approach	97
6.3	Roof design	103
6.4	Further development	105
6.5	Conclusions	108
7	Discussion	111
7.1	Reflection on achievements	112
7.2	Reflection on research objectives	114
7.3	Reflection on limitations	117
8	Conclusions	121
	Bridging the gap between architects and engineers	121
	Facilitate structural glass design	122
	Further development	122
9	Recommendations	123
9.1	Usability	123
9.2	Functionality	124
	Bibliography	126
A	Requirement tests	129
B	Structural verification of the prototype	132
C	Case Study 1	137
D	Case Study2	142
E	Case Study 3	164

Chapter 1

Introduction

This chapter describes the motive and driving force behind the thesis and summarizes the plan of approach.

1.1 Structural glass

Glass is already used for different purposes for decades, the implementation of structural elements however, has been done quite recently. In the past decade, there has been a rapid development of structural glass. With its high potential for compressive strength it is a material suitable for construction. An elegant example of an all glass structure is the roof cover of the Victoria & Albert museum in London, completed in 2009, comprising 73 laminated beams that span up to 11 meters.



Figure 1.1: Victoria & Albert museum (<https://architectenweb.nl/>)

1.2 Algorithmic design

Computational design is capable to provide faster, better and more efficient solutions. Computational design focuses on the link between technology and practice to use the theory of computer science. There are several finite element method (FEM)-based software packages, capable to analyze all types of structures. These however, are not suitable for the preliminary design phase, because they require a very detailed representation of the structure.

There is an increasing demand for integral solutions. Tools that are capable to be part of the visual and aesthetic process of the design, can easily be connected to FEM of complex geometry's, production costs and materials. The advantage of this kind of methodology is the ease of adjustment, for instance with technological progress and improvement.

1.3 Design process

Although departed from a common origin, there is a discrepancy between the architectural and engineering design models, education and practice. In short, the engineering model is characterized by its linear, sequential nature whereas the architectural model has a more spiral, cyclical disposition. Study to the similarities and differences regarding these models concluded that there is a need for reintegration across the disciplines. (Cross & Roozenburg 1992)

Architects are often stereotyped dreamers and visionaries, likely to loose touch with reality. Whereas the structural engineers are considered rigid and conservative, without imagination. During the bachelor of Architecture and master at Civil Engineering such a trend was noticeable to a certain extent. At the faculty of Architecture there is much more attention to innovation and durability as compared to the faculty of Civil Engineering. The ambition for this research study attempts to combine the best of both worlds.

1.4 Problem introduction

Though the fabrication boundaries of glass are pushed wider and the applicability has increased, the cost of structural glass is still relatively high and realization of an assignment happens only in a few cases. Consequently, not many engineers have experience with glass and general tools are lacking. It is desirable to have a structural design methodology to explore the possibilities of glass as a structural material and to use it more commonly in the early design phase. A methodology that combines the existing knowledge, though the principles of the structural design remain the same.

In the past decade, computational and algorithmic aid is entering the construction industry rapidly. This offers designers more freedom in complex architectural design. This research project will be used to find a methodology to fully utilize the computational developments to improve the structural advice and calculation capacity.

Focus remains on the early design phase, where the design decisions have the largest impact, in an attempt to close the gap between the architect and structural engineer.

In this report, the term “all-glass” indicates the use of glass for load bearing elements and minimization of steel support-elements.

1.5 Research design

This section discusses the main question and objectives of the research study, shows the research methodology and the outline of the thesis.

Thesis objective

The objective of this research is to investigate the possibilities of creating an algorithmic design for structural glass covers, to create an innovative methodology with a broad spectrum of potential applications. The methodology should stimulate the easy use of glass as a structural material. It should also contribute to the bridging of the gap between architects and structural engineers and take advantage of the potential of parametric design for architectural and structural purposes.

Research question

In order to achieve the objective of this thesis a research question is formulated:

‘How to develop a prototype for an algorithmic design methodology, considering geometrical control with structural feasibility, intended for structural glass covers in the conceptual design stage, with the opportunity for further expansion?’

Research objectives

Seven sub objectives are composed to answer the research question.

1. Provide a theoretical framework concerning the structural glass and algorithmic design.
2. Define which requirements, concerning functionality, usability, reliability and performance, should be accomplished by the methodology.
3. Define a general outline for the algorithmic method taking the requirements (see Objective 2) into account.
4. Implement the theoretical framework (see Objective 1) into the defined general structure.
5. Validate the outcome of the produced methodology.
6. Investigate the impact of the methodology on the workflow/functioning of engineers and designers by implementation on a case study.
7. Determine the required successive steps to obtain a fully functional method/model.

Research methodology

To achieve all objectives, and eventually answer the main question, the research project will be subdivided into various stages. Firstly, a literature study is conducted, to obtain a thorough understanding of the material properties of glass, the state of the art of structural glass and the characteristics of compressive structures and its conditions for successful application. To be able to efficiently design a parametric tool, this stage will also be used to get acquainted with Grasshopper and its possibilities and limitations. When this study is completed the acquired information will be analyzed and processed and used to determine the exact parameters, function and requirements of the design tool.

Secondly, the tool will be developed; before the programming starts a strategy of the design will be made. This method will be applied and the performance and outcome of the tool will be checked. The process will be repeated until the outcome satisfies the requirements set at the end of previous phase. Thirdly, when a satisfying result of the tool has been obtained, a case study for the design of a glass cover with complicated geometry will be implemented in the design tool and then evaluated for its functionality. Lastly the conclusions and recommendations regarding the tool and process can be drawn.

Thesis outline

The outline of this subsequent report content is described below:

Chapter 2 Theoretical Framework. This chapter shows a summary of the literature review conducted at the first part of the thesis. This chapter covers research to glass as a structural material, parametric design and the role of the designer and engineer in a design process.

Chapter 3 Strategy of tool development. This chapter provides a better image of the user and summarizes the requirements and demands for the methodology. A detailed architecture of the tool is constructed.

Chapter 4 Tool development. This chapter describes the working, content and structure of the developed prototype.

Chapter 5 Tool verification. In this chapter verification of the tool is conducted. Evaluation is performed to check that the tool satisfies the conditions imposed in the strategy. Additionally, a technical verification, based on structural mechanics, is performed using analytic programs and hand calculations.

Chapter 6 Tool validation. In this chapter three case studies are performed to explore some of the many applications of the tool and to validate the prototype.

Chapter 7 Discussion. Discusses the impact of an algorithmic design method for structural glass covers in the initial design phase compared to a more traditional approach. We consider the option to augment the tool with extra functions (e.g. calculating building costs), but also the possibility to incorporate the use of other structural materials like concrete, steel and wood in the design tool.

Chapter 8 Conclusions. In this chapter conclusions resulting from the research are provided. We also hope to conclude that the tool can help to bridge the discrepancy between the architectural and engineering design models, education and practice.

Chapter 9 Recommendations for further research. Recommendations concerning the conducted research and suggestions for future development are provided.

Chapter 2

Theoretical framework

During the literature study, the researched topics involved are: glass as structural material, algorithmic design methods - more specific the working of Grasshopper- and the characteristics of the conceptual design phase. This chapter briefly describes the most notable findings regarding structural glass, to be used during the research. In the appendices one can find more information gathered during this study.

2.1 Material properties

In theory, glass has a remarkable high compressive- and tensile strength. However, when applying glass knowledge and understanding of the mechanical properties are necessary.

Brittleness

The mechanical behaviour of glass can be characterized as perfectly linear elastic; a very brittle material with a maximal elongation approximately 0,1%. The lack of plastic deformation results in abrupt failure and makes it impossible to foresee failure. (Weller et al. 2009) Figure 2.1 shows the stress strain curve of glass compared to the ductile material steel.

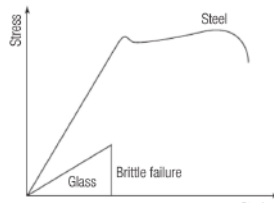


Figure 2.1: The stress/strain relation, compared to steel. (Barou 2016)

Strength

In theory, under ideal circumstances, the tensile strength for float glass is 6500 to 8000 N/mm². Nonetheless, the strength achieved in practice is merely a fraction of this theoretical value. Normative is the state of the surface subjected to tension, depending on the inevitable flaws. The surface of this glass is full of flaws due to the production and handling of the glass panes. (Haldimann et al. 2008)

The flaws and damage of a surface will not influence the structural behaviour when under compression. When under tension, on the contrary, the flaws induce local peak stresses, which might lead to growth of the initial flaws until it becomes unstable and rapidly leads to failure. (Balkow & Schittich 1999)

The strength is depending on several factors, regarding the type of loading, age, size and type of the panes. Due to the flaws in the glass there is a wide range of test results, which makes it meaningless to assign an average strength. Research shows that the quality of the edges are often determining the strength. Veer (2007)

For the characteristic values for (tensile) strength the following lower bound values are adopted:

- Annealed glass 20 [N/mm²]
- Heat-strengthened glass 45 [N/mm²]
- Fully tempered glass 80 [N/mm²]

The Young's Modulus of glass is 70.000 N/mm²

2.2 Structural safety

Glass is a very brittle material that shows no warning when it is overloaded. The material lacks ductility, plastic behaviour; so when cracking is visible, the glass will shatter. Several methods have been developed to address this unpredictability, due to the lack of ductility, and increase the safety of glass structures. Also the post breakage behaviour is an important aspect of the structural design. There are several procedures for strengthening the glass and there are several manners of processing the material

Pre-tensioning

One of the most important techniques to increase the safety of a structure is by pre-tensioning the glass panes. Aimed to reduce the tension stresses, by heat strengthening. Because the practical compressive strength is much higher than the tensile strength, this method introduces compressive stresses in the outer parts of the cross section, resulting in a higher capacity of the bending moment and reduction of the growth of cracks due to compression of the flaws. (Schipper 2011)

The image below shows the principle of tempering.

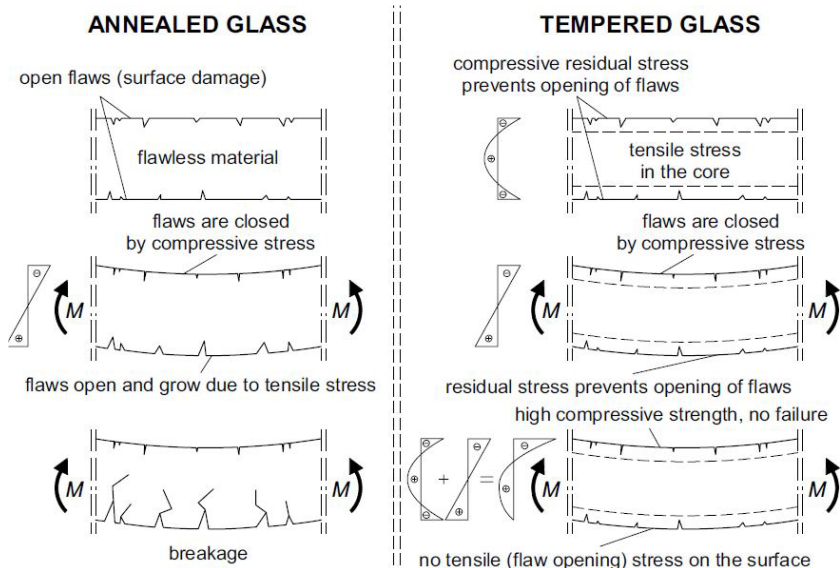


Figure 2.2: Principle of tempering glass. (Haldimann et al. 2008)

Due to residual stresses, It is not possible to process the tempered panels mechanically any further. Annealed glass is free of internal stresses and can therefore be processed by drilling and cutting without consequences. The same method is used with mechanical or chemical pre-tensioning as well.

Lamination

Safety can be augmented by lamination (the layering of glass panes through bonding of interlayers). These interlayers, developed for extreme circumstances like hurricanes and vandalism, enhance the strength and stiffness. There are high requirements for the glue used in the lamination; this needs to be transparent and able to cope with high stress to prevent failure by shear stresses.

The residual stresses of the tempering process influence the fracture pattern and thereby the structural safety as well. Due to these residual stresses, the fully tempered glass pane breaks in small fragments and thereby loses its structural integrity immediately after the first crack. Whereas annealed glass retains, even though limited, residual load bearing capacity.

Fully tempered glass

The tempering is done by rapid cooling of the glass panes, which leaves residual stresses. Penetration of the outer surface, for example for cutting or drilling cause immediate imbalance with complete scattering as a result. The internal forces cause an imbalanced situation, with complete shattering of the entire panel as result of failure, leaving no post-breakage load bearing capacity.

Heat-strengthened glass

The process of heat-strengthened glass has a slower cooling rate than the process for the fully tempered glass. The residual stresses and thus the tensile strength is lower than that of fully tempered panes. But the post breakage behaviour is much more desirable due the capacity of the larger fragments. (Haldimann et al. 2008)

Annealed glass

Due to the gradual cooling process the material is free of internal residual stresses. Therefore the material can be conducted to drilling

and cutting, but does not supply additional load bearing capacity. The fracture pattern of annealed glass is the most favourable, due to the largest size of the pattern and therefore higher safety post breakage.

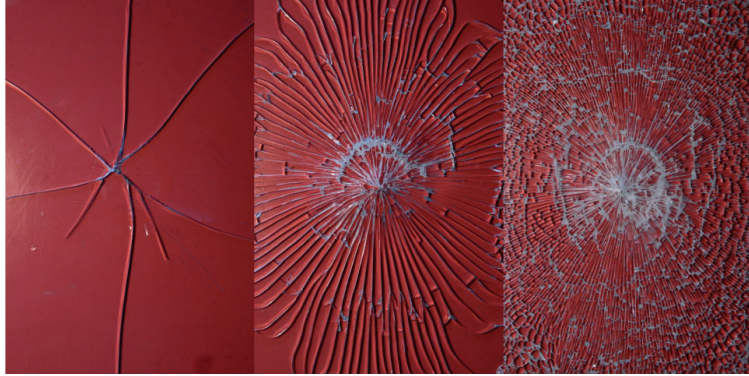


Figure 2.3: Comparison of the fracture pattern: annealed glass (left), heat strengthened glass (middle), fully tempered glass (right). (Haldimann et al. 2008)

Other technologies to improve the structural feasibility are the use of sacrificial layers, to function as extra buffer to add safety and protection to the inner panels, and the use of reinforcement. The latter, known for providing ductility to the comparable brittle material concrete, has a lot of potential to improve the structural behaviour of glass elements.

2.3 Manufacturing process

This paragraph describes the practical limitations arising from the manufacturing and transportation process. The figure below show certain aspects of the manufacturing process.

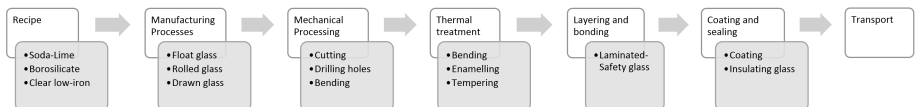


Figure 2.4: Characteristics of the manufacturing process of glass, based on Schipper (2011).

The use of float glass is beneficial due to its optical qualities and low tolerances, the latter is an important aspect because of the limited margins when constructing with glass.

PROCESS	RESTRICTION
FLOAT GLASS	3.21 x 6.00
LAMINATING TUNNEL	2,50 x 4.50
TEMPERING TUNNEL	2.50 x 6.50
AUTOCLAVE	3.20 x 7.50
COATING	3.30 x 6.00

Figure 2.5: Size restrictions, based on (Wurm 2007).

Figure 2.6 shows the size limitation of different aspects of the manufacturing process. Using smaller pieces is not always more profitable; the smaller the more process costs (think of edge grinding or polishing).

	THICKNESS [MM]	SIZES [M]	SURFACE COMPRESSIVE	FRACTURE STRENGTH	THERMAL SHOCK RESISTANCE	AFTER FAILURE	RISK SPONTANEOUS FRACTURE	CAN BE CUT
FLOAT GLASS	2 – 21	3.21 x 6	-	±	±	+	NO	YES
HEAT- STRENGTHENED	4 – 12	3 x 6	+	+	+	±	NO	NO
TEMPERED	4 – 19	2.8 x 6	++	++	++	-	YES	NO
CHEMICALLY	1 – 19	0.9 x 0.8	++	++	+	+	YES	YES

Figure 2.6: Comparison of tempered glass, based on (Schipper 2011, Wurm 2007, Haldimann et al. 2008).

For a big curvature hot bending is expensive and inefficient for transport.

GLASS TYPE	TOLERANCES	OPTICAL	COSTS	CURVATURE	RESIDUAL σ	DOUBLE	GUIDABLE	THICKNESS	FIXATION	TRANSPORT
WARM BENT	-	-	-	++	NO	+	+	++	NO	-
COLD BENT	±	+	+	-	YES	±	-	-	YES	+
LAMINATED	±	±	-	-	YES	±	-	-	NO	±
FLAT	+	++	++	--	NO	-	+	++	NO	+

Figure 2.7: Characteristics of the bending of glass, based on Wurm (2007), Schipper (2011).

2.4 Conclusions

In conclusion:

- Glass is remarkable strong in compression, approximately 21000N/mm^2
- Glass is a very brittle material, that shows no warning when it is overloaded. The material lacks ductility, plastic behaviour.
- The safety of the structure can be increased by using strengthened glass
- Safety can be augmented by lamination. These interlayers, developed for extreme circumstances like hurricanes and vandalism, enhance the strength and stiffness.
- Lamination can improve the post-breakage behaviour
- The residual stresses influence the fracture pattern and thereby the structural safety as well. Due to these residual stresses, the fully tempered glass pane breaks into small fragments and then loses its structural integrity immediately after the first crack. Whereas annealed glass retains, even though limited, residual load bearing capacity.
- Even more safety can be provided by a steel reinforcement of the structural elements
- Often a sacrificial layer is used to add safety and protection of the inner panels

Chapter 3

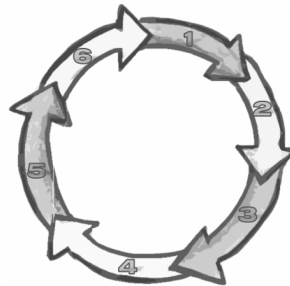
Strategy of tool development

3.1 Introduction

This research project focuses on the feasibility of the envisioned tool and is therefore only a Proof of Concept (PoC) development and is meant as a prototype to discuss its added value and potential for implementation in the design process. The PoC is just a small part of the envisioned tool and therefore not production-ready. Later in this thesis the possible extensions will be described and discussed in the requirements. From here on, the part of the software that lies within the scope of this PoC is referred to as the tool or software.

The life cycle of software development can be divided in 6 stages (Blanchard et al. 1990);

1. Requirement analysis
2. Requirement specification
3. Software framework
4. Programming/implementation
5. Testing
6. Maintenance/Evaluation



The information gained in the literature study is used to develop an adequate software design. To finalize this envisioned product, a strategy

for its content and development needs to be provided. The foundation of the strategy is described in the software requirement specification, derived from the user's need and can be found in this chapter. Hereafter the details for the software design are described. Chapter 4 illustrates the implementation and realisation of the prototype and the testing is described in Chapter 5 and 6. The maintenance stage will be discussed in the recommendations. Partially this means, keeping the tool up to date by adding of extensions, to enhance its use. An example of this is also discussed in Chapter 6.

3.2 Software requirement specification

Prior to this thesis no considerable experience in the field of software engineering was obtained; therefore the standards and procedures of the Institute of Electrical and Electronics Engineering (IEEE) are used as inspiration for the tool development and the base for the structure of this research. (IEEE 1998)

Product scope

As this research project is a proof of concept, it was decided to concentrate on the initial design phase. This part of the process offers a good opportunity to bridge the gap between architects and structural engineers. In an attempt to create an integral and workable method, this research implements the contrasting requirements of structural necessity, functional utility and aesthetic value. Assuming such collaborative development is the only way to create better designs, both architecturally and structurally. (Larsen & Tyas 2003) Decisions made in the early design phase have the most significant impact on design, feasibility and cost. The later the decisions are made, the more their influence decreases. (Frederick & Kuprenas 2013) Therefore, the focus of this research lies in the initial design phase and it is attempted to shift the knowledge of different parties to the beginning of the design process; this is shown in figure 3.1. We seek to enlarge the availability of essential design knowledge in the early phase, where it has a decisive influence on the overall process, as opposed to the more traditional design process, where this knowledge is introduced gradually into the process (Zaal 2014).

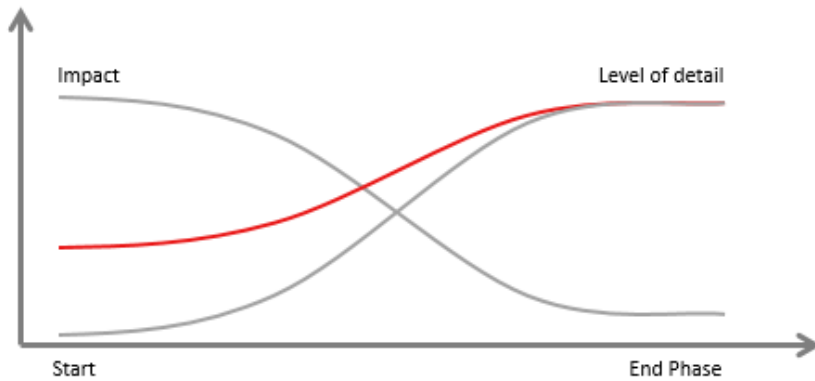


Figure 3.1: Characteristics of the design phase.

To stimulate the use of glass structures, the PoC is limited to the development of structural glass covers. The tool provides a platform with the capacity to collect the available knowledge and will link this to an integral design method. This might prevent people from reinventing the wheel. This will lower the barriers for and provide an introduction for working with this relatively new structural material. A fully functioning tool will avoid unnecessary repetition of work, calculations and drawings. The scope does not cover a ready to use product. It is, however, envisioned for the overall functioning of the tool to be valuable for further development. This includes the entire design and production process, as well as all types of structures and materials.

Overall description

This paragraph describes the user and his needs as well as the design phase and its requirements.

Product features:

The key features of this tool are:

- To generate design variants for glass covers;
- Dimensioning of structural elements;
- Structural calculation of structures;
- Structural optimisation of design;
- Form finding; Exportation of design.

User characteristics:

The intended user for the tool are both the architect or designer and the structural engineer. A distinction between experienced and inexperienced users with glass design is made as well.

The designer, or architect, often has a leading function in a design team and emphasizes on visual and aesthetic aspects of a design and aesthetic appearance. The tool will facilitate creation of integrated designs where concessions, due to structural soundness, are limited and gives the designer freedom to discover the possibilities with glass. Designers need the ability to work with standardized conditions to focus on aesthetics.

The structural engineer emphasizes the structural soundness and feasibility of a design. Generalized, the engineer often lacks creative education or imagination to fulfil the aesthetic demands of the designer. The tool will save the engineer a lot of repetitive work, hand calculations and eventually the writing of reports. However, the engineer needs control over the different variables and therefore full access to all standardized values to gain more freedom for optimisation and representation of reality and Eurocode standards. The inexperienced user in glass design could also benefit from the exploration of the possibilities glass structures offer. In future developments, the tool can be complemented and used as guide for glass design.

Naturally the distribution of roles and dynamics between the architect and engineer differ per project and design teams will generate a more integrated design. However, there is still plenty to gain in the integration of the design process. Even with a more integrated design process a lot of adjustment and refining will be necessary because of inadequate and time consuming communication. This tool provides the possibility to shift a part of constructors knowledge to the beginning of the design process and on the other hand to give engineers the possibility to take the role of a designer. It also has the potential to facilitate contact with the client and authorities; adjustments and additional requirements can be included in the design right away during gatherings.

For the development of the product, the focus lies on these two target groups, future development could, however, customize the tool for other users, like researchers, municipalities and manufacturers.

Operating environment/(product perspective):

The 3D modelling program Rhinoceros (Rhino) proves to be very efficient for parametric design. Rhino offers a visual scripting add-on Grasshopper, which is used as base for this tool. The versions used are Rhinoceros 5.0 and Grasshopper (Build 0.9.0076). For the analysis Karamba (version 1.2.2) and, as form finding generator, Kangaroo (version 2.2.1) are used.

Design and implementation constraints

The user's needs are leading design objectives for the development and dictate the structure and boundaries. To develop a user-friendly tool, it is important to keep in mind the two extremes: the designer or architect, inexperienced with Grasshopper and the structural engineer, experienced with Grasshopper. The inexperienced user should be provided with:

- a complete and simple tool with proper guidance;
- explanatory notes at different elements of the script;
- preset values or suggestions with common ranges of values;
- automated input and limited design variables;
- a clear overview of the necessary input and optional variables;
- notable signs when the structure is insufficient or overloaded.

Whereas the more experienced user with affinity for structural mechanics desires more freedom of design for advanced adjustments, modifications and optimization. In case the user needs to modify certain aspects of the script, it is important that it is readable and easy to adjust to avoid time loss. Even with simple algorithms, this is often not the case.

The aim to stay useful and up to date, is only feasible if the script is redeveloped. To do this a legible structure is required. Another important aspect for the tool to be implemented in the initial design process, is instant visual feedback, providing the user visualizations of the consequences of design choices; the only way to become an extension of the brain. To realize this everything needs to be interconnected.

3.3 Framework

For the continuity and future development, the framework of the tool is designed with care. The visual method of scripting comes with a downside; the numerous connections between the different components cause a spaghetti of connections and obstruct the legibility of the tool. The need for modification of the script is inevitable and described as “erase, edit, relate and repair” (Woodbury 2010). When the structure of a script and the internal relationships are not clear, changes are hard to make; it might then be easier to start all over with a blank sheet (Burry 1996). To prevent this and increase the clarity, the computational framework is an essential part of the strategy for development. Modular programming appears to increase the legibility of parametric schemata (Davis et al. 2011). For this research, the schema is divided into interchangeable modules, each responsible for a specific task or aspect of the desired tool, separating the functionality of the script. The extra time put in generating a clear structure, will be beneficial for the development, use and redevelopment.

All modules have the following general characteristics (Sharp 1992):

- A module performs one task and is named accordingly.
- Data enter the module through defined input parameters.
- Data leave the module through defined output parameters.
- Interconnectivity between the modules is accomplished by data through these parameters.

Certain benefits of modular scripting are described below (Mall 2009):

- The independent code can easily be reused and shared.
- Simultaneous development of separate modules by different developers is possible.
- Troubleshooting can also be done at a module level as opposed to the program level.
- Due to clearly naming of modules and in- and output parameters the function of the module is understood without the need of external documentation.

The division of the tool corresponds with the key features. A distinction is made in different modules: the boundary conditions, base geometry (form finding), structural properties, loads, analysis, optimization and

report. Each module is supposed to be easily adaptable or replaceable. In addition, new modules can be implemented to increase flexibility. The image below depicts the key features that establish the tool. The modular structure and its colors are implemented in the developed tool. Each module is subdivided in different units and components. This content will be elaborated on further in the next chapter.

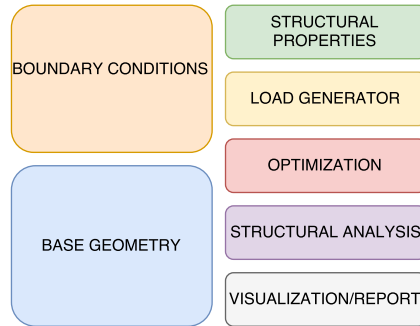


Figure 3.2: Structure of the script.

- Module1 *Boundary Conditions* captures or assigns the coordinates of the outer perimeter of the structure.
- Module2 *Base Geometry* is used to create the overall shape of the design.
- Module3 *Structural Properties* assigns material properties, cross sections, supports and nodes to the design
- Module4 *Load Generator* assigns a gravity simulator and variable loading to the design.
- Module5 *Optimization* is responsible for optimization of the cross section of the structural elements.
- Module6 *Structural analysis* performs a structural analysis on the input provided by the modules mentioned above.
- Module7 *Visualization* is mentioned separately, but functions throughout the entire process and is responsible for the visualization of the design and analysis.

System features

The tool can be used by both designers and engineers to generate architectural and structural designs, while respecting the mechanical and structural properties. Besides support of the design process, the tool contributes with form finding and structural optimization. The structural analysis of the tool can be viewed with real time values. A fully functioning tool will avoid unnecessary repetition of work, calculations and drawings. The fundamental factor for the architecture of the prototype are the functional and non-functional requirements, determined by the research conducted in the previous stages. The requirements state the needs, desires and purpose of the tool. For the continuity of the tool it is important to consider all relevant requirements needed for the tool extension. The functional requirements capture the intended capability and are clustered per main module and divided into three sections: input, process and output and are listed in the tables below. When requirements have only relevance for future developments they stated in this section in grey.

Table 3.1: Requirements Boundary Conditions.

Module: Boundary conditions		
	Identifier	Requirement
Input	REQ-1	Input of different potential outlines in the three-dimensional space.
	REQ-2	Standardized possibilities to facilitate the perimeter input.
	REQ-3	Provide for the possibility to import the outer perimeter from AutoCAD into Rhino and from Rhino to Grasshopper.
	REQ-4	Possibility to define the shortest span.
Process	REQ-5	Tool provides the possibility to create surface of the enclosure.
	REQ-6	Tool provides the possibility to visualize the perimeter.
	REQ-7	Tool provides the possibility to show dimension in view port.
	REQ-8	Tool provides the possibility to find the shortest span.
Output	REQ-9	Tool provides the possibility to create a mesh of surface enclosed by perimeter.
	REQ-10	Provides 2D visualization of the perimeter.

Table 3.2: Requirements Base Geometry.

Module: Base geometry		
	Identifier	Requirement
Input	REQ-11	Tool shall provide the possibility to explore different types of structures. Namely: grid, catenary and a projection on a random surface.
	REQ-12	Give the user freedom to adjust centre to centre distance among process.
	REQ-13	Give freedom to adjust maximal height of structure for form finding.
	REQ-14	Give freedom to adjust the direction and number of beams.
	REQ-15	Provide possibility to select materials, building codes, and structural analysis.
Process	REQ-16	Provide both 2D and 3D visualization of base geometry.
	REQ-17	Automate centre to centre distance.
	REQ-18	Provide instant visual feedback.
	REQ-19	Incorporate form finding based on limitations concerning flat, single and double curved plates and beams.
	REQ-20	Provide user notifications with consequences of design discussions and explanations of limits.
	REQ-21	Automated location of supports.
Output	REQ-22	2D and 3D visualization of base geometry.

Table 3.3: Requirements Structural Properties.

Module: Structural properties		
	Identifier	Requirement
Input	REQ-23	Tool provides user the possibility to adjust supports, material, cross sections and add hinges.
	REQ-24	The tool shall use rules of thumb for dimensioning the height of the elements.
Process	REQ-25	Tool shall apply simply supported beams as standard input.
	REQ-26	Tool will assign glass as standard material.
	REQ-27	Tool shall contain data-bases of material's properties and cross section profile. (Strength, stiffness, max. size, types, thicknesses.)
	REQ-28	User notifications with consequences of design discussions and explanations of limits.
	REQ-29	Implementation of the Eurocode in the hard code of the tool.
Output	REQ-30	The module shall extract the information needed for the structural analysis.

Table 3.4: Requirements Load Generator.

Module: Load Generator		
	Identifier	Requirement
Input	REQ-31	Tool provides user the possibility to adjust type and magnitude of loads and load combinations.
	REQ-32	The user will be provided with 3 loads that can easily be changed in magnitude.
Process	REQ-33	The tool shall apply gravity and variable loads that interact with the parametric structure.
	REQ-34	Implementation of the Eurocode in the hard code of the tool.
Output	REQ-35	The module shall extract the information needed for the structural analysis.

Table 3.5: Requirements Optimization.

Module: Optimization		
	Identifier	Requirement
Process	REQ-36	The tool shall integrate a module for structural optimization of the cross section.
	REQ-37	The tool enables allocation of multiple, laminated, rectangular cross sections.
	REQ-38	Implementation of the Eurocode.
	REQ-39	The system takes design and glass limitation into consideration, and informs the user about these restrictions.
	REQ-40	Tool shall implement limitations from glass production and processing for optimisation.

Table 3.6: Requirements Analysis.

Module: Analysis		
	Identifier	Requirement
Input	REQ-41	Tool shall provide user the ability to decide which calculations are performed and shown.
Process	REQ-42	The system supports principle stress analysis and visualization in the structure.
	REQ-43	The system shall provide calculations and checks for bending moment, torsion moment, shear force, displacement, principal moment and principal shear stresses.
	REQ-44	Implementation of the Eurocode.
	REQ-45	Tool shall provide both numerical as visual results of calculations.
	REQ-46	Tool shall provide real-time visual feedback.

Table 3.7: Report.

Module: Report		
	Identifier	Requirement
Input	REQ-49	Provide user the ability to visualize the stresses, deflections, reaction forces, moment distribution, normal and shear forces as well as cross sections, name tags, loads and materials used.
Process	REQ-50	Tool shall assign tags to cross sections, beams and loads.
	REQ-51	Provide real-time visual feedback of design choices.
	REQ-52	Tool shall implement analysis methods and design code checks.
Output	REQ-53	Tool provides users with report of geometry, calculations and analysis.
	REQ-54	The tool provides interoperability with other software.
	REQ-55	The tool provides 2D drawing extra extraction.
	REQ-56	Provide warnings if assumptions are made or tool is incomplete.

Other Non functional requirements

To facilitate the experimental and intuitive design process, while implementing simultaneous optimisation and structural analysis, some features concerning usability are required:

Table 3.8: Non Functional Requirements.

Non Functional Requirements	
Identifier	Requirement
NFR-1	No experience needed to make use of tool.
NFR-2	The response time shall not interfere with the user's natural workflow.
NFR-3	Provide immediate feedback.
NFR-4	Diminish creative obstruction of designer.
NFR-5	Provide real-time visual and numerical feedback.
NFR-6	Provide freedom to adapt structural properties.
NFR-7	Provide freedom to choose analysis.
NFR-8	The hard code shall be easily alterable and extendable.
NFR-9	The system will be transparent to avoid a black box.

3.4 Flowchart

As mentioned before, the continuation and handling of the tool demand a legible structure. Therefore, the flow chart depicted below will be leading for the tool development. The general constraints, preference for a modular structure, functional and non-functional requirements were taken into account for the generation of the flow chart.

An important constraint for the tool, if used as design aid, is to avoid obstruction of the creative design process. Therefore, it is possible to use standardized and automated input, supplementary to the manual input and adjustments. A distinction is made between the mandatory input and the optional input and design variables. This distinction can be seen in the horizontal division of the flow chart. The structure will also be obtained in the structure of the script.

The first column of the flow chart shows the manual input; the boundary conditions of the project. These data are necessary to start the process for design and dimensioning. Although these variables are limited as much as possible, to increase freedom for the designer, some basic input is inevitable. The adjacent column shows the actions that can be used as design variables, especially to create different aesthetic designs, but also to add several design conditions related to costs, transportation or properties of the existing building. In the third column, the user is offered the possibility to gain more control over the structural analysis and outcome. Here several input variables can be adjusted or added to the equation to give the engineer the freedom needed to create a desirable solution. This partition can be seen in the vertical division of the flow chart, situated on the next pages.

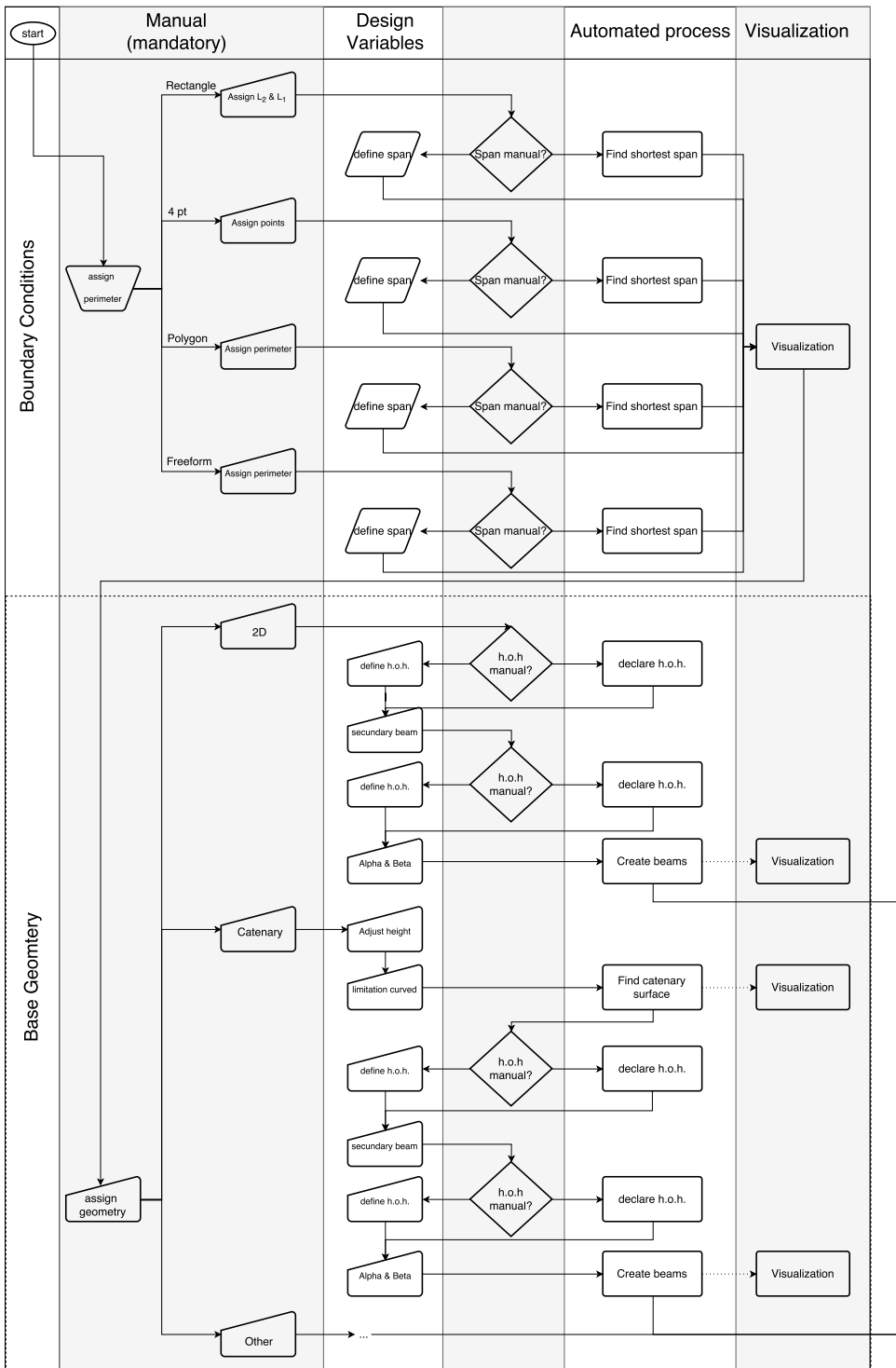
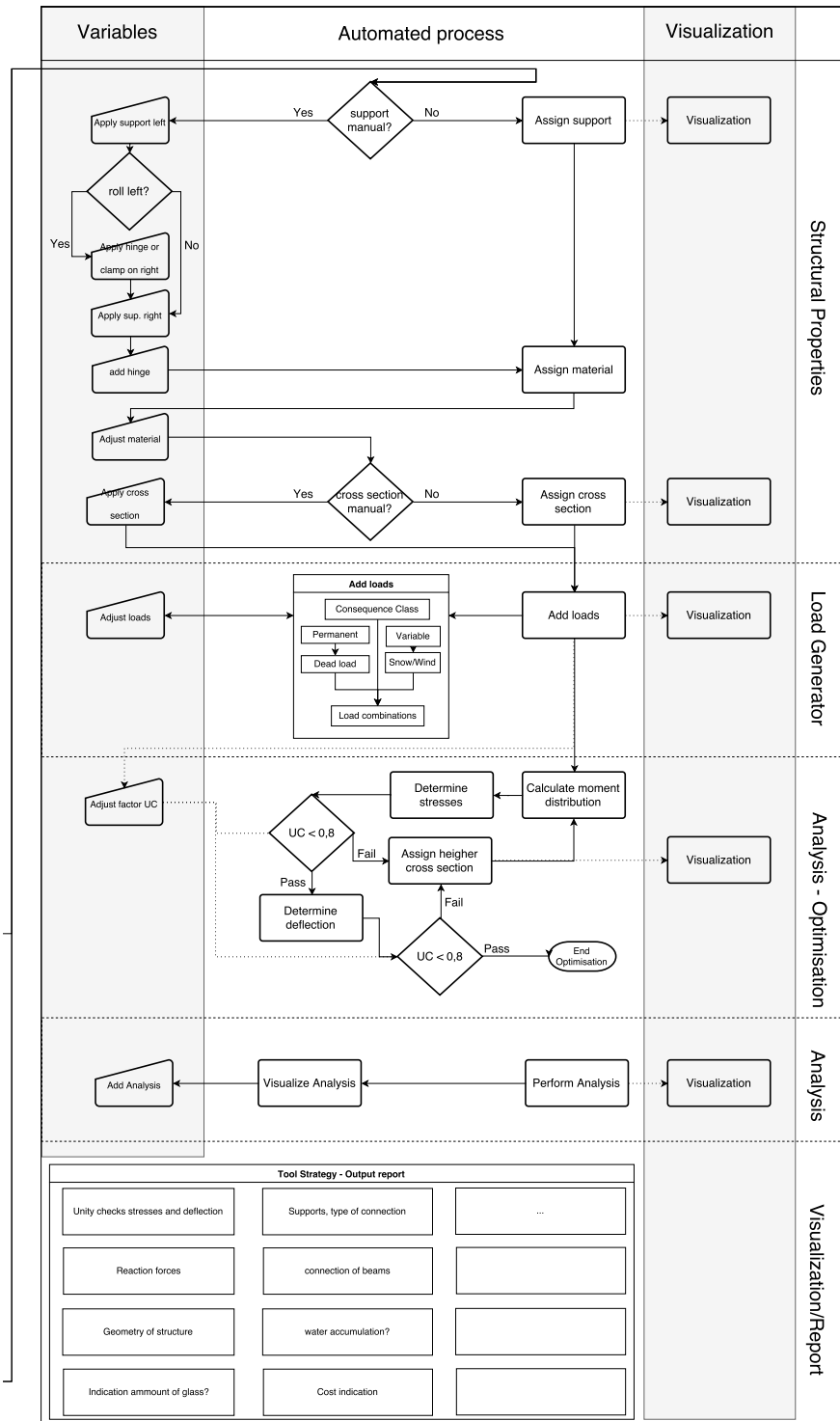


Figure 3.3: The intended flow chart for the tool.



Chapter 4

Tool development

4.1 Introduction

Aforementioned strategy and computational framework are leading for developing the final product and implementing its structure. Its modular configuration for the sake of future development and readability of the tool renders the ability to add, adjust and remove modules. In order not to hinder the intuitive design process, interference with creativity is avoided by minimizing the mandatory manual input. It is fully up to the designer to choose the design variables. Contrarily, for engineers, augmentation of the control is desired to gain more design freedom. In line with these desires, the partition of the computational framework is implemented in the tool to increase clarity.

The proof of concept (PoC), covered in this thesis, studies the feasibility of a design tool for structural glass roofing in the initial design phase and the potential for further use. As mentioned before, the focus lies on realization of a limited part of the envisioned tool, in order to enable the verification. This chapter described the process of development as well as the description of the prototype and its functioning.

4.2 Test-driven development

The previous chapter described the ambition for the tool, but the remaining question is: “where to start?”. In order to make the process manageable, the tool development is based on test driven development (Beck 2003). A process that starts with the writing of a small test; followed by quick adjustment of the test until it works; completed with a clean-up of the test, eliminating the duplication needed for the test to work. The process is characterized by a simple start, clarification of the script and addition of design decisions, one at a time. This principle functions as base for development of this research project and is shown in figure 4.1. New functions are adjusted until they work properly; only then a new function can be added.

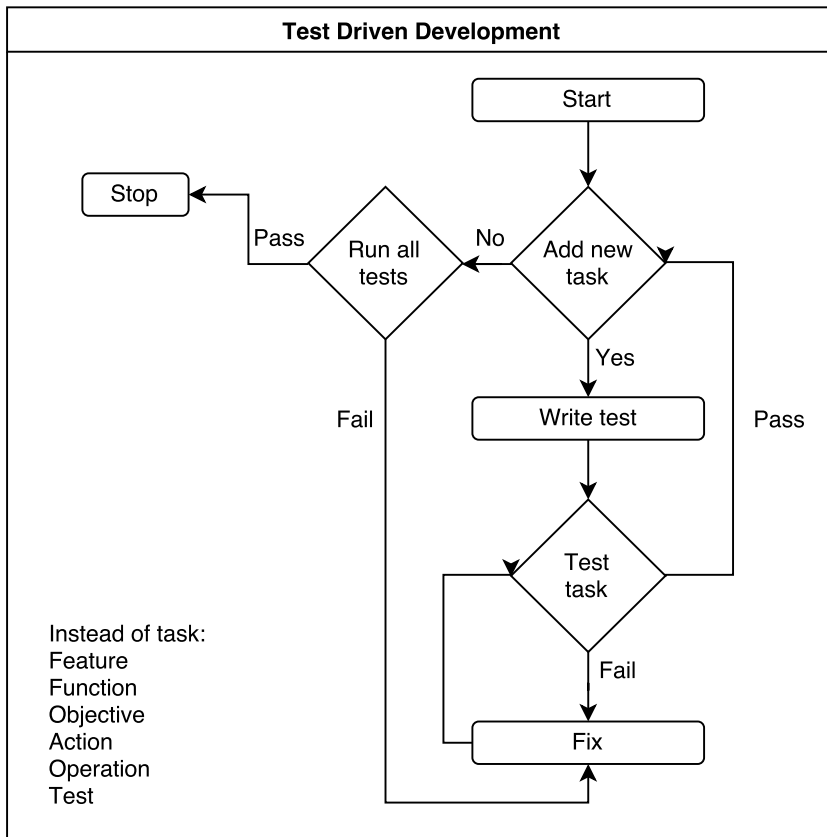


Figure 4.1: Test driven development.

Phasing

The tool development is divided into several stages, not per module, but with use of a hierarchical scale: from coarse to fine. First the vital functions are established, then the tool gets extended with the more advanced elements. Only when the first phase functions, new tasks and eventually entire modules can be added and the whole will become more comprehensive. In this way the tool will reach a higher level of detail and complexity until all requirements are met and a good foundation is made to check the feasibility of the concept. The extension of the tool for future development can be done following the same principle. In the Appendix A an overview is given for the tasks per phase, envisioned to create the prototype of tool. Further extension is recommended to amplify the usability of the tool. For example the use of other materials or a link to cost efficiency, to be suitable for usage in all phases of the design process, even during construction.

For the prototype, we discern five phases:

Phase 1: Outline

Phase 2: Optimization

Phase 3: Form finding

Phase 4: Extension

Phase 5: Future developments

Phase 1 – Outline. The development in this stage is restricted to the vital functions of the tool, the connection between the geometry, the analysis and the immediate feedback between the two. The outline therefore covers merely the basic functions of the tool. The perimeter is limited to a rectangular shape, the user can vary the width and length. The span is in shortest direction and the centre to centre distance is adaptable. This can be done with either the distance or the number of beams leading. The dimensions, span and centre to centre distance are displayed in Rhino's interface. For this phase, the structure is bound to the 2D surface and either spans in one or two directions. There are standardized values for the cross section, material and loads. One can adjust the supports and loads. An analysis is made, with implementation of Karamba, for the stress distribution, reaction forces and deformation. The focus lies on the interactivity between all parameters and functions that induce real-time feedback in Rhino's interface.

Phase 2 – Optimization. This stage focuses on the optimization algorithm and follows the first phase. The optimization of the cross section in this stage depends on the maximum stress and deflection. The algorithm relates to the model, so in case of adjustment of the loads, span, centre to centre distance or material properties another cross section will be assigned. Real-time feedback is expected and for the output a name tag of the profiles should be interactive with the dimensions.

Phase 3 – Form Finding. This stage focuses on the implementation of form finding in the existing tool design. This will be done with different methods, one of these intended methods is inspired by the hanging models as used by Gaudi. There are different possibilities to implement the hanging models, which will be explored. Due to the interactive design, any adjustment in geometry will result in an immediate adjustment of the structure. This stage will elaborate on the previous stage and should be connected to the optimization module, this way cross sections are assigned to different elements automatically.

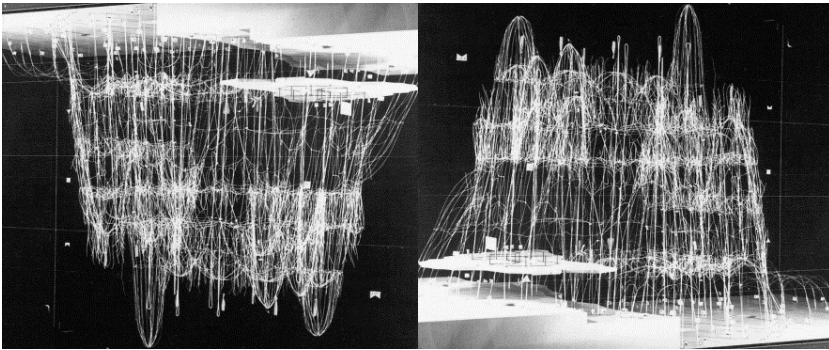


Figure 4.2: Hanging Models of Gaudi. (Xie et al. 2005)

Phase 4 – Extension. Now that the key features of the design are working, further expansion of the tool is started to explore some of the possibilities for future development. The outline of the tool is expanded to a polygon with 4 corner points, a third dimension is added as well. Manual input of free-form perimeters, made in Rhino or imported from AutoCAD, is also made possible in this stage of development. For the geometry, a new method is introduced where the beams, of any pattern, are projected on a surface which can be changed during the process. Triangulation of the catenary structure will also be added to the script.

Phase 5 – Future Developments. The execution of this phase is not a part of the research project, but important to bear in mind for development of a functional foundation. It is acknowledged that the prototype needs more extensive testing and development to be fully functional. The vision for future development connects to the prototype, but should contain a more detailed approach of the work than is done in this study and extensions are advisable.

For each phase, mentioned above, different tests were written as guidance. The descriptions of the tests, per phase, are illustrated in Appendix A.

4.3 Framework

This section discusses the realization of the prototype.

Graphical user interface (GUI)

The graphical user interface of the researched design method is provided by Grasshopper. The main screen of Rhinoceros provides the visualization of the design decisions and the opportunity to ‘bake’ a design variant to make it manageable by Rhinoceros. The tool and its management are accessible in Grasshopper.

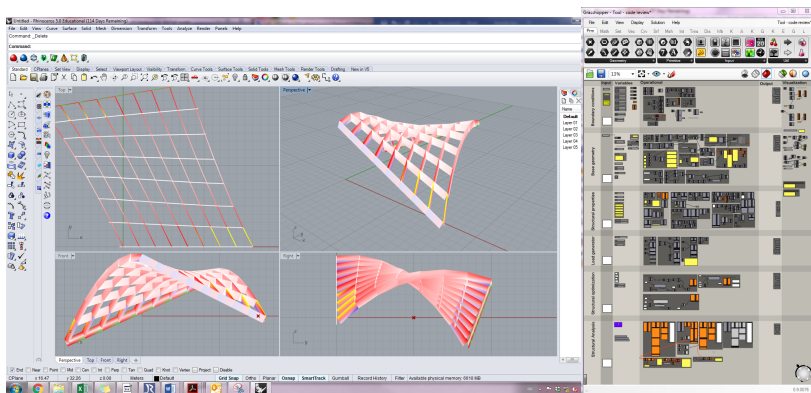


Figure 4.3: Graphical user interface prototype.

The modular set-up, used to divide the requirements and structure of the development process, is used to provide a legible structure for the tool. Each module is responsible for different features of the tool. The modules are separated with the horizontal division of the script. Another partition in order to increase the usability of the tool is visible in the vertical distribution.

- On the left, in the first column, the input that is required to start the process. To increase the simplicity of the tool, the required input has been minimalized.
- The second column contains all design parameters. Contrary to the first column the input of this columns is maximized, to enlarge the potential of the creative process and design possibilities by increasing the influence of the designer. Naturally there is an optimum of the number of parameters. Too little means a decrease of influence, whereas too many leads to a decrease of distinctness. A choice of limited variables is made, this should be customized to the user and its needs.
- The third column contains the elements responsible for the execution and performance of the functions and represents the operational process.
- The fourth column contains the output of each module.
- The fifth column contains enables the visualization of the secondary aspects needed to clarify the generated drawings.

There is an overlap between the second and third column, focused on the engineer regarding the structural model and analysis. It is possible to change the conservative standardized values, however it is recommended to do this only when one possesses the knowledge to do so. Therefore, an extra barrier is built-in to discourage the inexperienced designer. Changes for matters like supports, loading and analysis are done by referrals to the relevant part in the script to adjust the settings. Figure 4.4 shows the Grasshopper code. The modularity and vertical distribution can be recognized.

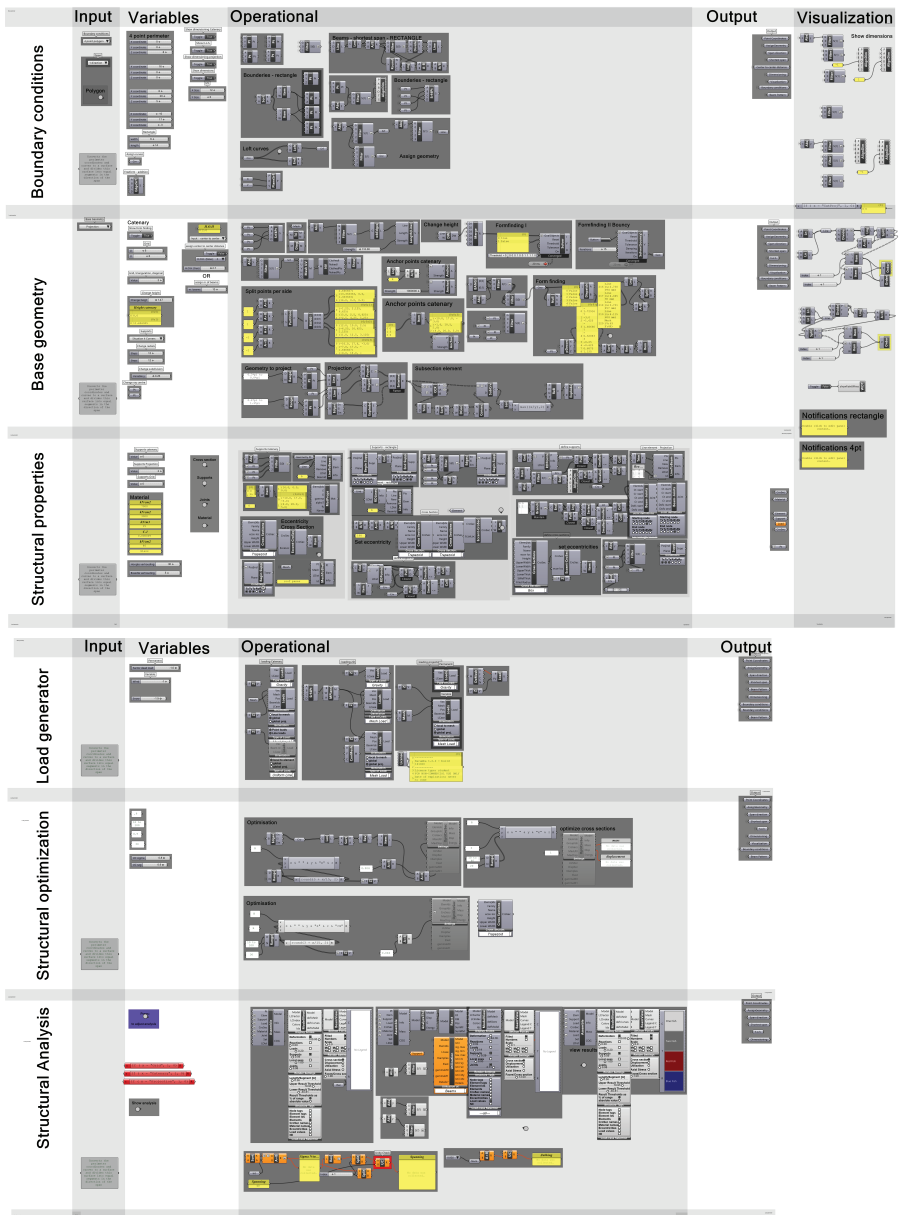


Figure 4.4: Overview of the script, corresponding to the flowchart.

Script structure

Certain aspects are essential for a clear set up of the modules (Davis et al. 2011) which can be adopted for parametric design methods:

1. Title of the module, concise and corresponding to its task
2. Inputs, defined and grouped on the left of the layout.
3. Outputs, defined and grouped on the right of the layout.
4. Group. The module is grouped together for easy identification
5. Description. A brief description of the procedure

These aspects are adopted for the structure of the individual modules. This section elaborates on the realized prototype that is based on the strategy and main flow chart. The functions, and to a lesser extent the content of the script, are described per module.

Boundary conditions

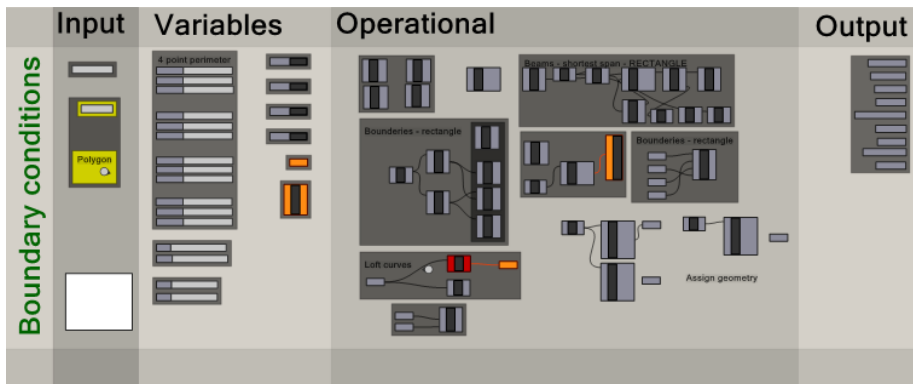


Figure 4.5: Overview of the script for the Boundary condition module.

This module handles the necessary input for the start of the design process. In consonance with the requirements it allows the user to assign the perimeter and to set the design and dimensioning process in motion. The number of possible outer perimeters is endless. There are different techniques to mimic them, but the most efficient technique differs per situation. Therefore it is not a solution to provide all alternatives, because the script will become illegible.

For the proof of concept, elaborated in this thesis, it is decided to provide different approaches that cover all possible perimeters. Depending on the specific user's needs, a script can be manually implemented for more efficient methods. The controllable input is limited to rectangles, four-point polygons and two approaches for free-form, composed of line and curve elements. Although it is not time consuming to extend these possibilities, this was not considered as adding much value for this PoC.

The mandatory input, shown in the left column of the script, is reduced to the assignment of the method of input and the dimensioning of the outer perimeter shape. The specific dimensioning is done with the variables in the second column of the script. This part also provides the dimensioning of the perimeter in the Rhino interface, later on this can be easily implemented in the output report. For the prototype the user can choose to show these dimensions during the design process. The operational process, placed in the third column is responsible for the generation of the perimeter. The requisite input for the continuity of the design method and adjacent module is a surface, which can function as base for the geometry of the roof cover. The surface that represents the roof area is used as input for the next modules and will stay interactive during the rest of the process. This is one of the main advantages of the parameterized script. For example, the user can vary the height of the supports to gain aesthetic and structural improvement or even the location of the supports. While doing so the calculations and dimensioning will be adapted at the same time.

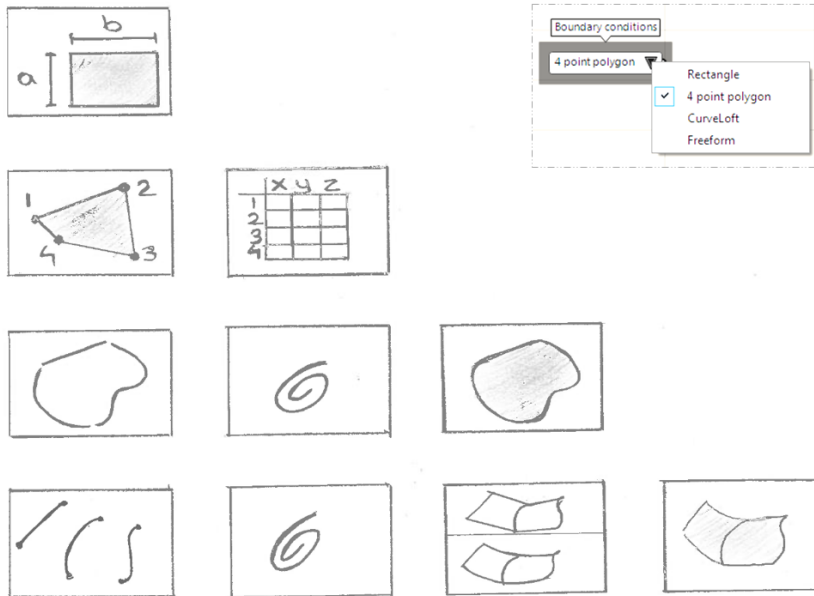


Figure 4.6: Selection of method for the boundary condition module

As mentioned before, the development started with a basic rectangular geometry, of which the user can adjust the width and length of the base. The rectangle is scripted in a way that it is always oriented with the shortest span in y direction. The perimeter can be assigned with 4 corner points, each with a separate x-, y- and z-coordinate as well. There are different possibilities to assign the coordinates of these points. The first is to add points placed in the Rhino interface. The second option is to adjust the location of the corner points by dragging the point manipulator in Rhino. Usually Rhino's interface is not interactive with Grasshopper and merely functions as visualization of the Grasshopper script unless one bakes the elements to make it tangible. The third option is to adjust the parameters of the x-, y- and z-coordinates of the points.

The user has the freedom to implement every single shape imaginable to create a more random perimeter in the 3D space. This is done by NURBS in Rhinoceros, that encloses the roof surface and represents the surrounding buildings or outer perimeter of a structure. Both straight and curved lines can be adapted. There are two methods to do so:

1. Assigning the curves that represent an enclosed outer perimeter and create a surface in between. These can be added by right click on the curve container in the tool and select 'set multiple curves'.

2. To 'loft' multiple NURBS curves that run from different corner points, to create a surface that represents the area of the roof. This last option provides more freedom and grip on the geometry, because these curves can be adjusted during the process.

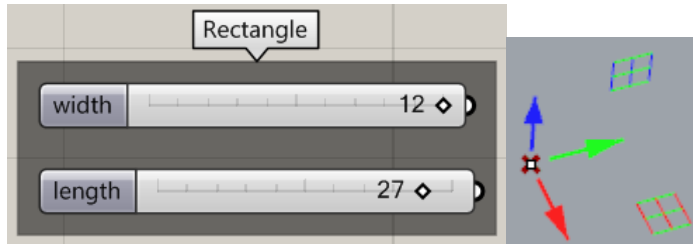


Figure 4.7: Width 'a' and length 'b' (left) and the physical manipulation of a 'GH point' in Rhino (right).

Therefore, the output of this module is a surface. One might think that at this point the remaining script and method is the same for all types, as all perimeters are converted to surfaces. However, to facilitate the user, it is attempted to maximize the automatic detection of supports, span and so on, which differ per scenario.

Base geometry

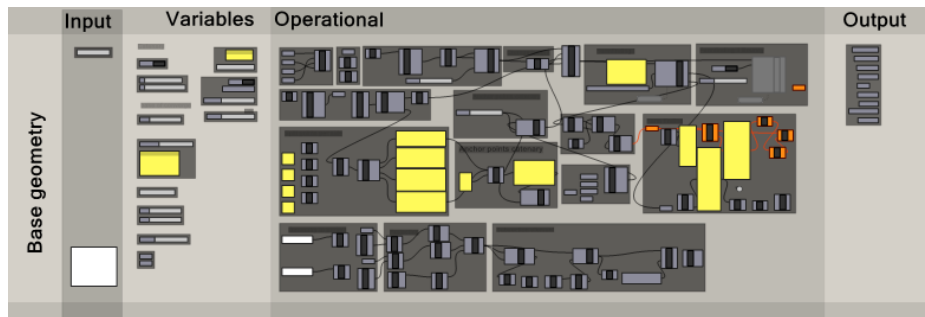


Figure 4.8: Overview of the script for the base geometry condition module.

This module has the largest influence on both aesthetics and the structural aspects of the design. It offers the user 4 different alternatives to create the geometry of the design. It is considered that on the one hand a large variation of geometries is desired, but on the other hand this detracts from the clarity of the script and its use.

Similar to the first module, a selection of the endless possibilities is elaborated upon in the script, to show the potential of the algorithmic design. The user can vary with multiple parameters and different possibilities can be explored and compared in an instant. Firstly, the module contains an algorithm for a geometry in the 2D plane, in either one or two directions and every random pattern; secondly an algorithm with a catenary form finding; a third option with projection, which gives much more freedom; fourthly is a combination of the catenary surface and projection of a beam pattern; and lastly, of course: a full manual input is possible as well. For future developments, every imaginable extension can be implemented.

The different types of geometries are discussed hereafter, but what they all have in common is the adaptability of the center to center distance or mesh size, direction of the span, use of secondary beams and the beam configuration. When the boundaries of the roof structure are imported, the direction of the span will be determined. For structural feasibility, it is desired to find the shortest span of a structure to avoid unnecessary use of material. Often another direction is preferred due aesthetic motives. Therefore, detection of the optimum span and freedom of the span direction is desired. The user has different options to determine the span. He can decide to do this manually, define either the shortest span or a direction which might be more pleasing for whatever reason. The direction of the shortest span can be visualized in the Rhino interface as well. To avoid devious spans, a sanity check will be performed, for the user. Especially for plans with protrusions this can save a lot of material. Another possibility in case of a non-rectangular plan is to let the user appoint one side and the opposing side. Then the tool can divide this in equal sections.

There are different possibilities for the user to define the center to center distance. The standardized value for a parallel system is 2.1 meters, so no secondary beams are needed. This module continues on the input of the previous module and uses this surface, to convert it to a surface mesh. The only mandatory input is the choice of geometry. All other properties can be adjusted with the design variables of the second column.

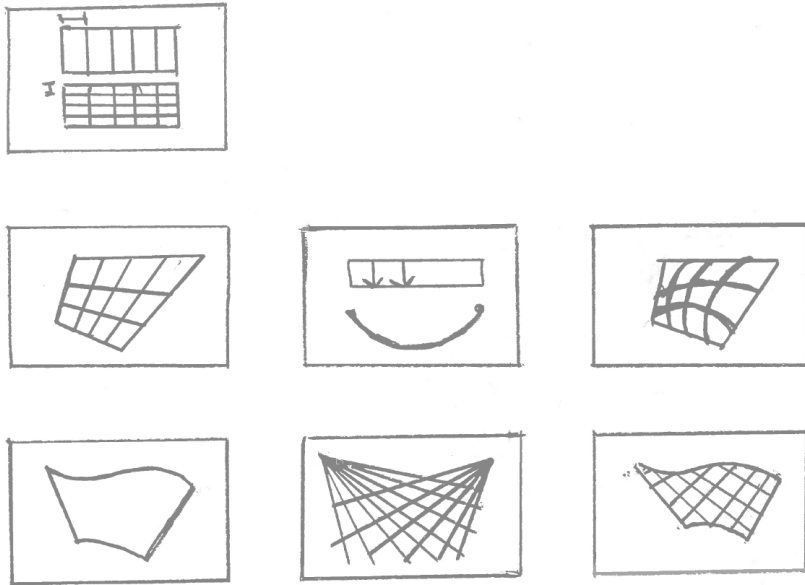


Figure 4.9: Selection of method for the boundary condition module

Form finding - grid To create the basic geometry, the user can influence the structure with several parameters. First a base is created in the 2-dimensional plane, all types of perimeter from the previous module, except the lofted curves, can function as a base for this 2D plane. The user is free to make use of secondary beams, since this has a substantial aesthetical influence. The next step in this sequence is to determine the center to center distance of the beams. There will be the possibility to do this manually, to choose between several options or automatic assignment of a suitable distance. When the beams are not parallel, the maximum distance will be leading. With the use of secondary beams the distance between the main beams can be increased. When the 2-dimensional distribution is provided, the user can choose to give the structure a third dimension. This can be done with several extensions. For the prototype several possibilities are added.

Form finding - hanging models The remarkably high compressive strength of glass led to the requirement to include a hanging model generator in the prototype. Efficient from a structural point of view and lots of design freedom with visually interesting results. This principle of a structure under pure compression, solely loaded by gravity, executed by hanging a chain model and turning it upside down, can be

simulated by several numerical form finding techniques categorized in three groups: (Adriaenssens et al. 2014)

- Stiffness matrix methods, based on using the standard elastic and geometric stiffness matrices.
- Geometric stiffness methods, using only a geometric stiffness, being material independent.
- Dynamic equilibrium methods, solving the problem of dynamic equilibrium, to arrive at a steady-state solution, equivalent to the static solution of static equilibrium.

The traditional physical form finding methods provide great insight in the structural behaviour, but are time consuming to compose. These models can also be simulated digitally with the particle spring system, a dynamic equilibrium method. A system consisting of particles, springs, forces and anchor points to simulate a structure. The springs are hingedly connected to the particles, which might be subjected to a force or fixed to a specified point. Since the system has no capacity to resist moment forces, all loads are transferred through axial forces. This method is suitable for a quick exploration and offers a lot of freedom of design parameters and offers insight in the structural behaviour.

There are several software packages that use the particle spring method, but most of them cannot be implemented into computer aided design and are difficult to use. However, Kangaroo 3d, a plug-in for Grasshopper developed by Daniel Piker, offers the user interaction with real time simulations, and will be the base of the hanging model in this module. (Tedeschi 2014)

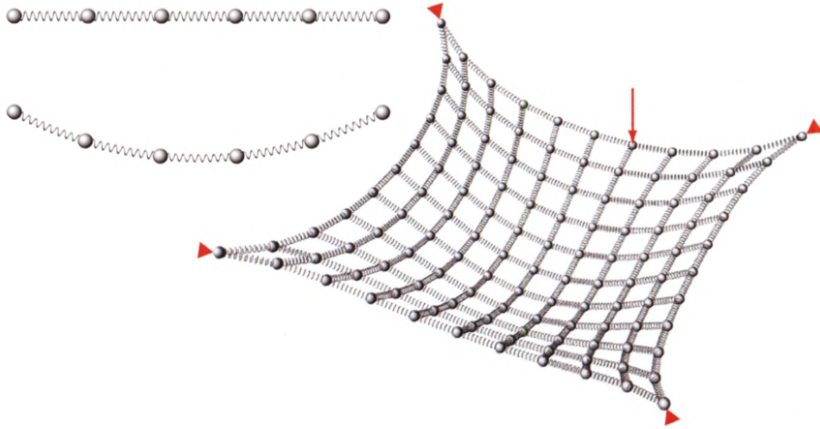


Figure 4.10: Particle spring method.(Tedeschi 2014)

The Kangaroo plug-in offers a lot of possibilities; the ones implemented in the prototype are described below. The main component is the solver, and stores all 'goals'. The solver is turned on and a threshold is added with a low value, $1.00E10^{-6}$, to automatically stop the iterations when a steady state solution is reached and start the calculation when adjustments are made.

The goals connected to the solver provide information regarding the starting length, rest length, spring stiffness and dynamic loading in the negative z-direction, opposite to the gravity. This dynamic loading is used to manually control and adjust the height of the structure. The difference can be observed in figure 4.11

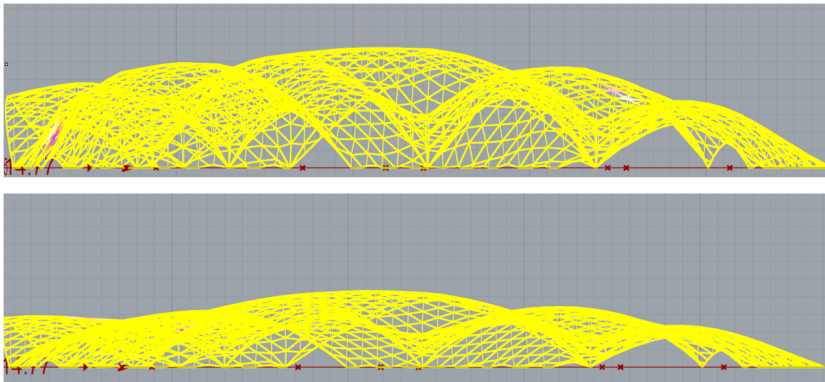


Figure 4.11: A change of height, induced by an increase of gravity.

The pattern of the particles and springs are converted from the surface output of the boundary condition module, both the 4-point, rectangle as free-form principle. This surface is used to create a 2D surface mesh, which in its turn is used for different patterns. This 2D structure will turn into a catenary net with an appointed height. The gravity generator is reversed in order to get immediate visualization of the result. At the moment a script is added for a simple grid, divided in u and v direction, triangulations of this grid and diagonalization of this grid. The u,v-division is a parameterized variable for the user. The triangulated grid offers the most stability, compared to the other distributions, however, it causes an extra difficulty for the connection of the glass elements. Therefore, the grid distribution is the default setting.

Another variable and goal for the solver are the anchor points. For the prototype several configurations are programmed for each principle of the boundary condition module. The supports are fixed at one point, done with a high anchor strength with a magnitude of approximately $1.00E+106$. The four configurations: fully supported on all sides, supports of the corners, supports of two opposite sides and supports of the corners and midpoints of two opposing sides. The selection is done with the component 'naked vertices' which sorts the vertices of a mesh into lists according to whether or not they are surrounded by faces.

The user can influence the form finding process by adjusting the mesh size and the location of the supports; these do not need to coincide with the end of the beams. Another function that is implemented in the prototype is triangulation of the mesh, which offers higher stability and a change of looks. Thus the hypnotizing patterns of Calatrava's steel structures can be the inspiration for glass design. Another, structural glass related, feature is the possibility to use straight or curved glass beams and flat, single or double curved glass panes. This might be preferable for the costs of a project due to the consequences of glass production and transportation.

Form finding - projection The third form finding technique allows the user great freedom, possibilities and influence. The principle is quite simple, it involves a surface and beam pattern, to be projected onto the surface. All perimeters generated with the boundary condition module can function as base surface. The lofted surface, as shown in the first module, offers a lot of design freedom. The user can assign or adjust several NURBS curves which are used to compose the basic area surface. The end of these curves and the outer curves are in line with

the outer perimeter. The curves get lofted to create a surface, which will be adaptable during the entire process with help of the control points on the curves. But, virtually all surfaces might be used. For the composition of the beam, also practically all patterns can be used.

One of the possible patterns that are integrated in the prototype is described below. It is a technique with radial curves originated from two source points, located outside the perimeter. The position of these two points can easily be adjusted, both 'physically' from Rhino's interface, by adjusting the script in Grasshopper or by assigning another point. The user can adjust the number of rays and the angle over which the rays appear. These lines get projected onto the surface and will become the beam elements of the design. This leaves the user great freedom for both the shape of the base surface as well as the beam pattern. The range of the radial projection is set to 0.3π to 0.8π , respectively 0.8π to 1.3π , so that it covers the projection surface. The number of radials, distributed over this range, can be manually adjusted by the user. In this way one can alternate and optimize between the number of beams and height of the beams. For future developments, this could be added to a multi-objective optimization.

The next step in the script is to extract the edges of the lofted surface and project the rays, as defined before, onto the surface. All curves are flattened and entwined into a collection of data streams. Next, the parameters of the intersection are solved, sorted and filtered for duplications. Subsequently a domain is drafted between each intersection and used to divide the original curves. These elements are rebuild into smaller segments. The user is free to change the subdivision of the elements. Varying from one governing height, up to a more organic distribution revealing the load transfer. The latter option assigns the cross section only where needed, but might be less economic, due to lack of standardization, depending on the production process.

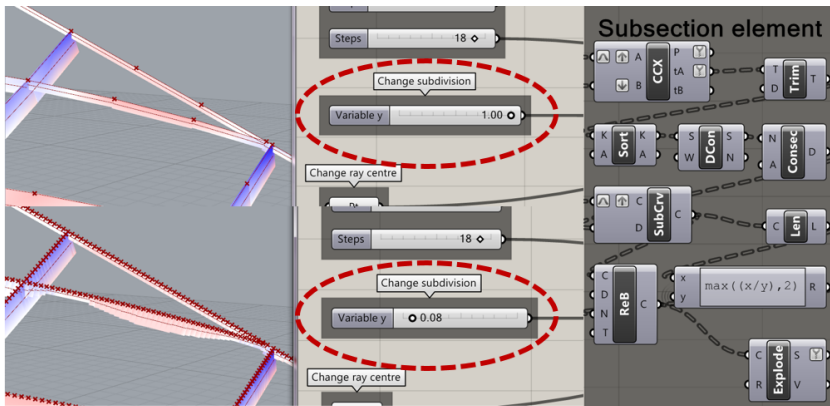


Figure 4.12: The difference of the segmentation.

Due to the parametric nature of the script, this process is repeated after each adjustment. Note, that during the entire process both the surface and the beam pattern can easily be adjusted, while simultaneous an update of the analysis and optimization occurs.

Output In order to implement this module into the prototype, it is important that the generated information is transformed into a format usable for continuation of the process. When the aspects mentioned above are assigned and defined, all beam elements are assembled in a collection of line segment, which is needed later on to be converted to beam elements, detectable by the module for the structural analysis. This is the end of this module, the output can be connected to other modules.

Structural properties



Figure 4.13: Overview of the script for the module of structural properties.

This module is the link between the aesthetical and structural part of the prototype and forms the base for the analysis. For this part of the script all parameters, that are usually needed to be specified repeatedly in structural analysis programs, are standardized; the mandatory input has been eliminated. The main reason for this is the reduction of creative obstruction, when combining the structural analysis in the early design process. However, the ability to easily influence and adjust all aspects of this module still remains.

The elements needed for the analysis procedure are supports, hinges (if present), cross sections and material allocation. These elements are all standardized to increase the design freedom. For this PoC the default settings are as follow:

Supports: The default location of the supports coincide with the supports used for the form finding, extracted from the naked vertices component, and are linked to the support components. A different part of the script is needed for each type of boundary condition and are therefore automatically regulated with a stream filter to assign the supports corresponding to the assigned perimeter. With a double click on the icon, the user has the possibility to move the script to the part where the supports can be adjusted. By default this is set as simple support, but the user can adjust the translational and rotational restrictions of the supports to create a roller, hinged or fixed support on either sides.

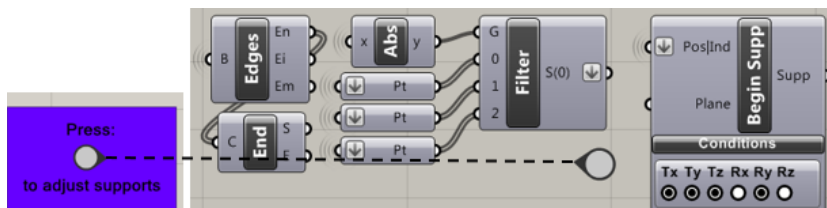


Figure 4.14: The link to adjust the operational code.

A notification is given for the consequences of the type of supports on the glass design, as they need to be designed with care to avoid local peak stresses. This applies also for the influence it has when a connection should be demountable or not.

Cross section: The default setting for the cross section is a rectangular shape, composed of laminated panes with available varying thicknesses. For the first phase of the development the rule of thumb is used to provide the initial dimensioning of the cross section, because the

optimization module was not yet implemented. The rule of thumb was used to determine the initial height and width of the elements. In case of a horizontal beam the height becomes 1/10th of the span and lower for an arched beam. This depends on the angle for the arched beam, 1/30th of the span, when the total height exceeds 1/6th of the span. The formula for the width is nonlinear, and rounded off to combine with existing pane thicknesses.

Karamba provides a component, that can generate several cross sections. For the prototype, several cross sections are available besides the rectangular shape: the I-section, hollow box, circular, trapezoid and plate or shell cross section. The height of the profiles has a lower limit of 100mm and upper limit of 1000mm. Automatic naming of the cross sections, according to their dimensioning, is provided as well. These standards will be overridden, when the optimization module is activated. For the optimization, the complete list of cross sections and its smallest member is needed. This list is sorted from most to least favourable, starting with the smallest cross section and can be directly implemented in the module for optimization. The plate thicknesses and composition of the profiles are discussed in the optimization part. All cross sections are collected in one component to be further used as input for implementation of other modules. Manual input is always possible and other shapes can also be realized. The beams are aligned from the upper part of the beams to fit the glass panes, and therefore are shifted half of its height to its local z-direction. If necessary this alignment can also be changed.

Material: The part of the script that assigns the material to the design is identical for all alternatives and therefore no extra arrangements were made. Karamba provides components to assign all kinds of material. Therefore the prototype already provides the possibility to use different strength classes for concrete, steel and timber. The default settings are set to match the properties of structural glass. The user has the ability to adjust the material; this provides the possibility to easily compare the application of different materials. Optimization between materials is something that could be a future addition for the tool. The used material properties are as follow.

Young's modulus E : 7000 kN/cm²

Shear modulus: 3000 kN/cm²

Gamma: 25 kN/m³

Thermal expansion: 0,000009 kN/cm²

Yield strength 40kN/cm²

Hinges: By default, no hinges are applied to the structure. An extra feature would be the hinged connection (single, or double in case of two clamped supports). The optimal position could easily be determined by the program. This will not be part for the PoC, but could be a first extension to get a better grip on the design.

Output: The mandatory input of this module is completely eliminated, by standardization. The design variables provide easy selection and exploration of different pre-set options. The output of this module is collected, in order to create an uncluttered overview of the information generated by the module, so that other modules can be easily implemented.

Load generator

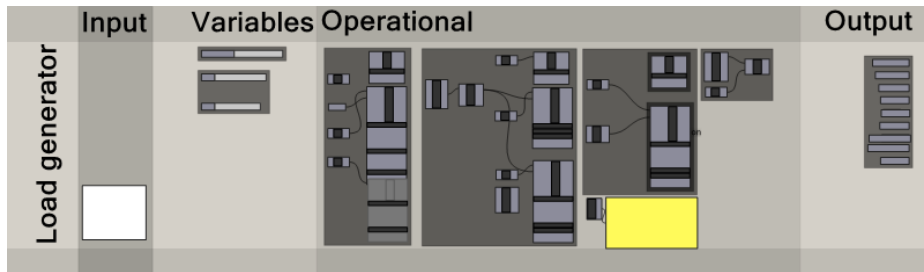


Figure 4.15: Overview of the script for the module of load generator.

This prototype is intended for the initial design phase, therefore it is decided to implement a simplified and conservative representation of the loads described in the Eurocode. Any load module could be added as separate load to complement or replace the default settings. The extension of this module is quite limited for the proof of concept. It is considered important to implement the dead load of the structure as well as imposed loads and to apply basic load combinations to show the many possibilities. Of course, the tool offers possibilities of optimization and this should be exploited in future developments. The load generator can easily be automated and connected to the Consequence Classes of the norms. This automatization enlarges the suitability for the more detailed design phase.

Also for this module, components of the Karamba plug-in offer a vast variety of loads for a structural model. The user can add point, line and

mesh loads, but can even apply load due to initial strain, temperature and imperfections, later on to be used for the module of structural analysis.

The script assigns automatically a gravity loading to the structural elements and a roof surface, mimicking the roof plates. The load generator provides different load cases, the applied factors for the loading 0.9, 1.0 and 1.2 in the negative z-direction and two load cases are applied, one for the ultimate limit state (ULS) and one for the serviceable limit state (SLS). The area over which the variable loads need to be applied is depending on the shape of the structure and therefore depending on the type specified in the module for boundary conditions. This means that the output from the other modules is filtered again with the stream filter component before it is connected to the script of the load generator.

There are several methods to apply the variable load, representing the wind-, snow- and maintenance loading, to the structure. It's possible to use point loads on all intersection points of the beams, line loads on the beams in one direction or a mesh load that is projected onto the structure. For the latter, the load will be automatically distributed uniformly over all beams, however, it is possible to apply beam identifications of specific beams to control this. The factors used for the load vector are 1.5 and 1 for the ULS and SLS. For all loads there is the possibility to set the orientation of the load with respect to the elements or the global orientation. To prevent mistakes from happening, it is decided to avoid dependence of the local orientation and use the global orientation to assign the loading.

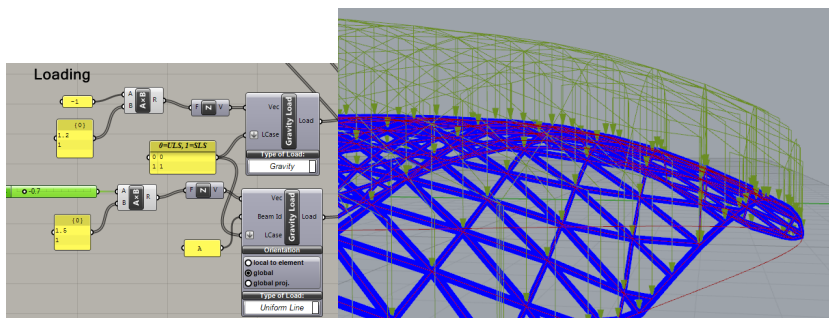


Figure 4.16: A random example of the assignment of a mesh load to a mesh surface.

One of the central key features is, like the rest of the modules, the interactivity with the geometry of the roof. When this changes, the shape of the loads will change with it and when larger cross sections are applied, the dead load will change accordingly. This saves the user a lot of extra steps and prevents mistakes from happening. The output of the loads is connected to the assemble component, together with the structural properties, so that the analysis can be performed.

The mandatory manual input of this module is eliminated. The values for the load generator are quite conservative, as they are used for the initial design phase, and are intended to be connected to the Eurocode. This would require basic project information as input. The design variables are responsible to easily adjust these factors. The automated process can also be adjusted, for example to add point loads and asymmetric loading. All types of loading are collected in an output parameter for easy connection with other parts of the script.

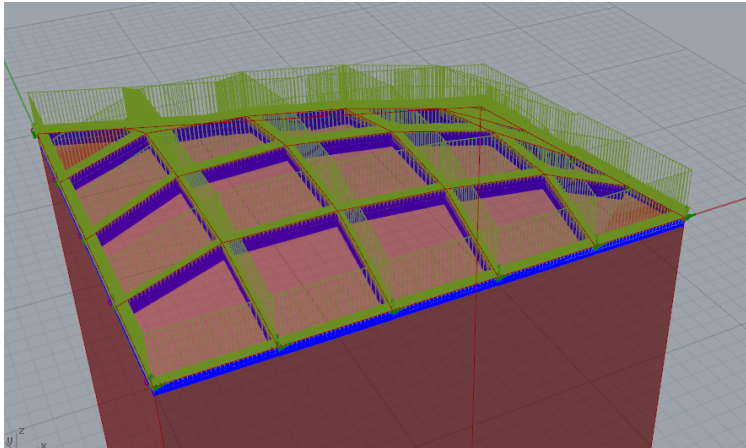


Figure 4.17: The load distribution depends on the geometry.

Structural optimization



Figure 4.18: Overview of the script for the structural optimization module.

There are many possibilities to implement algorithms for optimization. In the prototype the focus lies on the optimization of the cross section lead by the mechanical behaviour. As can be seen in the flow chart the maximum stresses and deflection are leading. One of the variables is the factor for this unity check (UC), the ratio between the maximum allowed value and occurring value. The standardized values for the UC for the stresses and deflection are taken into account with a factor of 0.8, to incorporate a buffer for working in the initial design phase.

The 4th phase of development is used to interconnect the optimization module with the other modules, in accordance with the setup of the tool and therefore no extra actions are demanded to apply to all scenario's. This is done so with the stream filter in order to determine and filter the elements subjected to the optimization module, the maximum deflection and the structural model.

Within one scenario, assignment of different elements or identifiers of multiple elements gives the opportunity to apply a different type of cross section to specific parts of the structure.

For the dimension optimization of the elements are several possibilities. The size of the objects to be optimized can differ from infinite small to a single element that represent the entire structure. The user is free to change this and can find one governing size for all beams or a completely specialized output, where the beams fluently differ in height. This module could therefore also be used as another design algorithm, with structural feasibility as leading factor. Figure [xx] shows the difference between a small interval and the entire elements, cut between intersections.

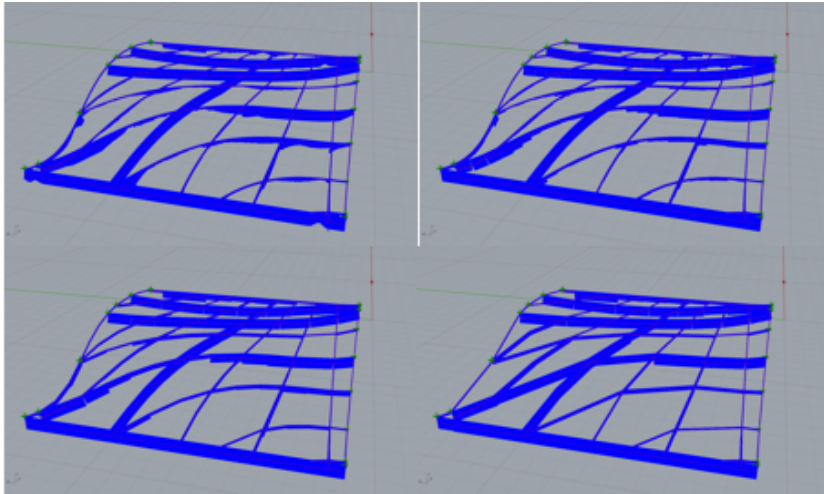


Figure 4.19: Small segments (upper left) to large segment (lower right).

As mentioned before, the optimization module uses input from the structural properties module. The flow-chart, as described in the strategy for the tool development, shows the unity check for both stresses and deflection of the elements, hence the need to interact, and iterate, with the module for structural analysis as well.

This module automatically assigns the first cross section that fulfils the given requirements on the list of cross sections; the tool calculates the stresses and deflection when the smallest cross section is assigned. A script is composed to detect the span of the structure, which is used to determine the maximum deflection of 0,004 times the span. If the requirements are not met, a profile, one step bigger, is assigned. This is repeated until it satisfies the requirements.

The expression ' $\text{round}(3+x/15, 2)$ ' is used for the width and therefore grows non-linear with the height. Adjustments can be made easily. Each cross section gets a name tag that shows the dimension of the cross section. For this prototype, a range for the height is used, that varies from 100mm to 1000mm and increases in 30 steps. The standard rectangular cross section can also be varied

Structural analysis



Figure 4.20: Overview of the script for the structural analysis module.

This module is responsible to examine the generated designs. Its most important feature is to produce instant feedback, both numerical and visual. Accomplished by assembling all design variables and aspects into the analysis. Everything is interconnected, the boundary conditions, the basic geometry, the optimization, structural properties and the load generator. Even the slightest adjustment in one of these fields will lead to a re-calculation of all other modules in less than a second.

To explore the many possibilities the plug-in offers, the script is mainly composed of Karamba components. The first step of this module is collecting the output of previous modules. Karamba provides components to assemble this information and generate a model from these inputs. This is connected to the element responsible for the calculation. For this prototype, focused on the initial part of the design, the overall form and dimensions are most important. The structural analysis focuses mainly on the stress distribution, maximum deflection and support reactions of the structure.

The output is standardized and parameterized for the sake of simplicity. It allows the user to switch on or off certain aspects of the analysis. There are different options to display the internal forces. This can be done textual, showing the maximum values or the values per element or sub element. However, this last option is not very clear, because with an increasing number of elements, the clarity decreases rapidly. A more legible option projects the outcome in the visualization of the structure in Rhino's interface. It is possible to show the stress distribution or deflection along the structure, accompanied with a legend.

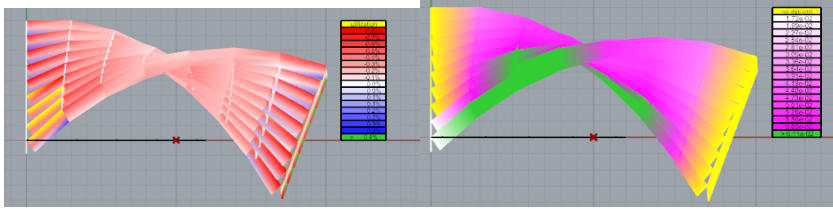


Figure 4.21: The stress distribution (left) and Deflection (right).

The internal forces can be projected along the elements axis. This provides a clear image of the stress distribution of the structure, and shows where adjustments or extra arrangements need to be made. The scale of these distributions can also be modified, to provide as much a clear oversight when looking at the entire structure, as when completely zooming in to examine a connection.

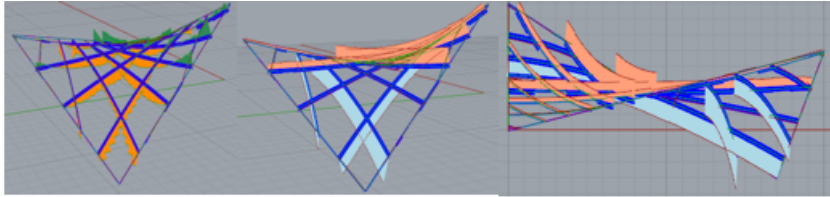


Figure 4.22: he anticline beams under compression (blue) and syn-clyne beams under tension.

The structure can be visualized in the deformed state; in the section 'display scales' the user can exaggerate this by adjusting the scale of the deformation as showed in the figure 4.23. This holds for the support reactions, as much as for loading, local axis and internal forces. One can choose to show and calculate the support reactions, the displacement, axial stresses, M_x M_y and M_z , as well as N_x , V_y and V_z . The calculation can be performed with different load cases, which are easily adjustable.

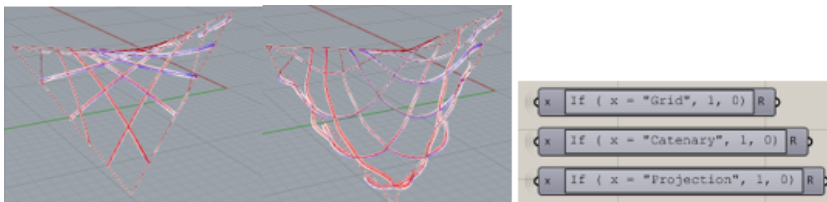


Figure 4.23: The highly exaggerated deformation

It is decided to provide the user with three sets of analysis, for each

base geometry. Therefore, a different stream filter sequence is used. In this way the user is free to keep the desired settings, that are principle specific. This can also be used to quickly compare a different analysis for the same type of structure.

For glass structures, the design of the connections might be the most important part of the design. The prototype offers a clear insight in the force distribution of these connections. This method has the potential to link a file to all connections, their conditions and limits, to automatically assign to the structure according to the occurring forces.

In case the tool will be used for a more detailed part of the design process, the module can easily be extended with tests for lateral buckling and stability. For the prototype, different possibilities are implemented already. The Karamba elements work like a black box and therefore will be checked in the next chapter to ensure that the working is as expected. It should be noted that the tool should not be executed without the permission of a specialist, a structural engineer. The tool should be a design assistant to avoid unnecessary repetition of work. Sanity checks and comparison between the input of the tool and the physical situation of the project always need to be done. One cannot blindly trust the output of any structural design tool, because mistakes in the input or inadequate use of the method are inevitable.

Output and visualization

During the design process, all individual steps and decisions are supported by the visual representation in the interface of Rhino. The user can choose to show or hide certain information. The name tags, dimensions or material of the elements and the outcome of the structural analysis can be switched on and off when desired.

For this prototype, the report that should be created by the tool hereafter is neglected. Nevertheless, some notifications are implemented in the tool that could be incorporated in a report. When the stresses or deflection are exceeded and also when the largest cross section is used, a notification appears. Moreover, a short overview with the maximum moment, stresses, deflection and support reactions is given, as well as the applied loads and type of supports.

When the design satisfies the expectations, it can be exported to other programs. First the structure will be baked, in order to transfer it to Rhinoceros, this means that the model is not connected any more to its parameters. It can only be changed by hand in Rhino or other programs. Another possibility is to open it in GSA for further structural

analysis. The script is constructed in such a way that this will perform automatically and the procedure will be described in the case study.

Workflow

This tool can be used on three levels. Firstly, for the recreation of a certain design, created outside the tool, mimicked with the settings provided by the tool. Secondly, for the creation of a design, with help of the tool, optimized in a conservative manner provided by design variables. And finally, for the creation of a design with help of a more advanced development, elaborating on the existing tool. A short description of the integral use is described below:

The only input needed from the user is the immediate surroundings of the project, which can be done manually or with help provided by the module for boundary conditions. From here on the designer is free to explore the design possibilities with the provided variables, options and written suggestions. Instantly he will see the visual result of the changes, both visually and structurally. Due to the module for optimization, not only can the tool provide instant structural feedback, it also assigns profiles fulfilling the structural requirements.

The standardized options enable the user to choose, which type to use as a base for the structure. This prototype provides four options, ranging in degree of adaptability from a minimal surface, spanning between the boundaries (that leaves no means for modification), to a surface lofted through control point curves, providing maximum influence on the shape. Different measures are taken to facilitate the process of entering this outer perimeter in the tool. Subsequently, the prototype already offers different options to generate different geometries; the most remarkable being the catenary function, the projection and the combination of the two. All options are also equipped with sub design variables. Think of the division patterns, influenced by shape and mesh grain. For the structural properties everything is standardized; the main design options are the location of the supports, divided into several configurations, ready to use. Also, besides the default rectangular shape, simple change of cross sections and material are provided for.

The last three modules are especially interesting for the engineer and offer a lot of freedom to adjust and optimize the design as discussed in the description of the concerning modules.

In short, the mandatory manual input is minimized and the design

freedom for inexperienced user is amplified with a substantial amount of possibilities and combinations. It is also encouraged to let the user explore the possibilities beyond the default setting and look into a further extension of the tool. In order to do so a basic understanding of the method and working of Grasshopper is necessary. The clear set-up and modularity contributes to this purpose. The amount of possibilities is endless as the combination of all types of supports with the different boundary conditions, the wide range of patterns etc., and five meshes translate into infinite possibilities. As mentioned in their description, the modules are not implemented using parallel computation, but are combined in an iterative process. At first, the different options were developed as loose scripts, processed simultaneously, later all different functions were integrated with the help of several stream-filters. Two types of stream-filters are distinguished. The ones that add all types and possibilities to a single function and the ones that divide these into groups.

4.4 Work in progress

This research is part of a bigger ambition, and therefore in order to see the feasibility of the tool and its further development, it is decided to explore certain parts, and make a start on the prototype to show the possibilities and uncover possible pitfalls.

The research conducted to further expansion includes the incorporation and adjustment of various functions regarding several modules as described below.

One of the most important extensions is the structural analysis for plate structures. At the moment this is limited to single plates or entire shell surfaces, but fully implemented in the structure of the prototype. Exposed to the same loading components, connected to the optimization module and varying the thicknesses. The analysis shows reliable results, however, the script needs to be evolved in order to simulate single panes, matching the glass plates. In order to do so it is important to simulate the possible connection method, varying from structural sealant along the edges to bolted point fittings. It was too much effort to implement the plate analysis into all the scenarios provided by the tool, therefore it is decided to check the principle based on a single plate. This turned out to be reliable and so it is recommended to pursue this research. This principle is already used to specify that the beams should be straight or curved.

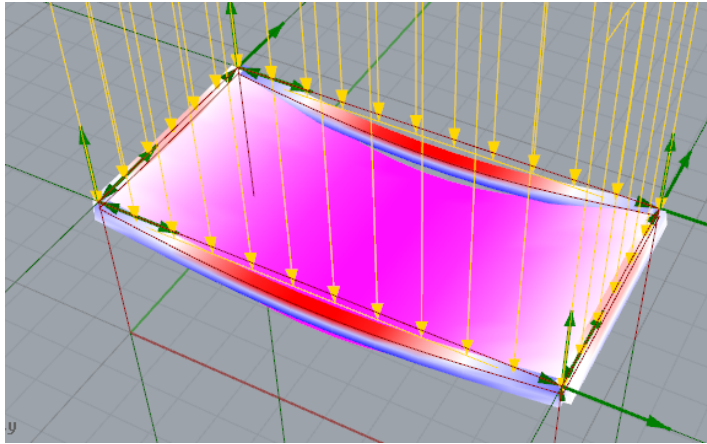


Figure 4.24: The plate analysis combined with the analysis for beams.

Another important progress regarding structural glass, is the development of the base geometry module controlling the curvature of the glass panes. This is, as mentioned in the literature review, because the costs increase significantly, when a curvature or double curvature is needed. In order to discover the possibilities, two methods are explored using Kangaroo's form finding components. Both methods include diagonalization of the mesh. The first gives the diagonals infinitely high spring stiffness and the second method ensures the diagonals to maintain an equal length relative to each other, both ensuring the panes to stay flat. Both methods do not yet provide a perfect result and the second creates distorted parts. However, it shows already promising results for a first try-out.

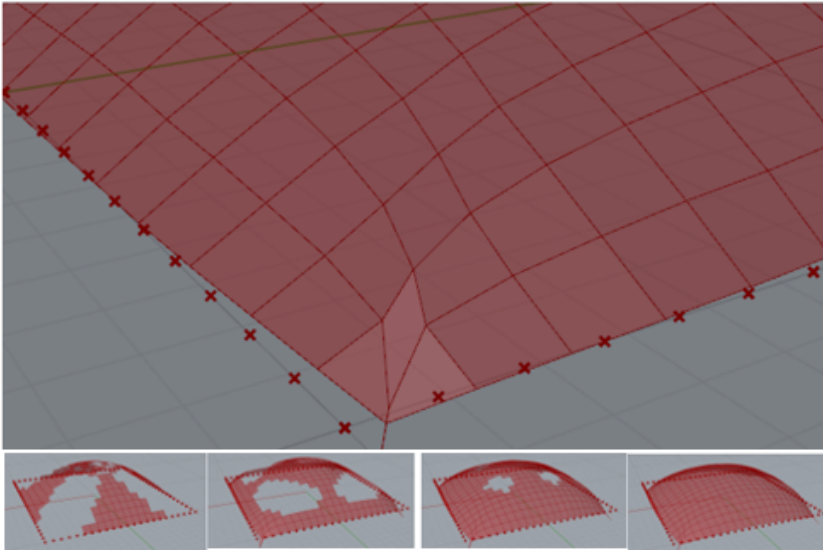


Figure 4.25: Restriction of the panelling to flat panels

Incorporation of automatic placing of panels to the structure and analysis, restricting the panelling to flat panels, single or double curved is possible as well. A simple start is made for rectangular perimeters.

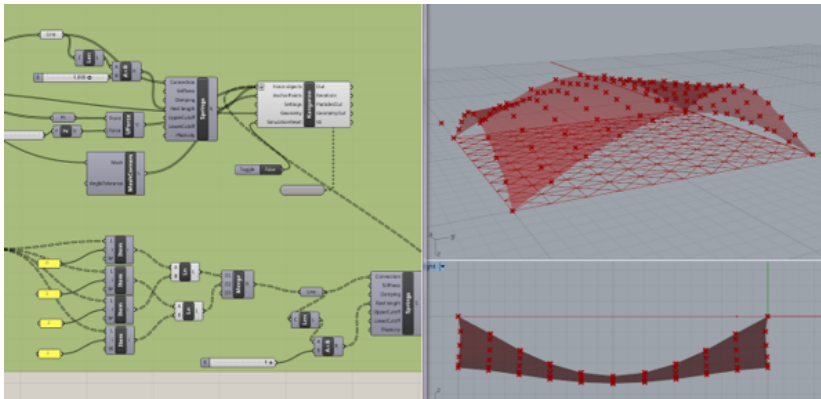


Figure 4.26: Restriction of the panelling to flat panels

4.5 Conclusions

The prototype already covers a large part of the design process and different possibilities can easily be discovered by the 'design handles', offered by the tool.

The downside is the complex character of the script and the required knowledge of not only the program language, but also the specific script itself. There are multiple strategies thinkable, therefore the script might still be experienced as a puzzle, even though a lot of time is put into the legible and logical arrangement, composed with logical reasoning.

Besides, it is impractical that the user is unable to 'get a hold' on the visualized elements. Everything, all alterations, with few exceptions, need to be scripted, even the smallest modifications. Whereas often the user is used to modify and alter the geometry from a visual point of view, instead from a theoretical point of view. So, the tool provides visual exploration, but cannot compete with the intuitive process: it only provides handles to mimic this.

For example, with a physical mock up, the designer can wonder what happens, if he turns it around, tweaks a certain part, removes this part, adds that, and so on. With the algorithmic method everything needs to be scripted, so in one way 'everything' is possible, but on the other hand the possibilities are limited until they are made possible. This causes a devious process, when it is not properly developed. A proper design language needs to be created to let the designer speak as he is used to. In order to cope with this discrepancy, it is of importance to become more familiar with the usual design process and extensive testing should be conducted in order to further research the needs that will lead to the requirements for further fine-tuning.

The prototype offers the designer already a lot of options, if the user is willing to adhere to the available options. When the designer is willing to explore these possibilities it's possible to refine the design in the physical space, for example baked in Rhino with a single click.

Chapter 5

Tool verification

5.1 Introduction

There is a difference between the, often interchanged, terms verification and validation. Verification is an objective process, done by developers, to ensure the functionality of a software system. This chapter contains a description of the verification and covers the question: Are we building the product right? This needs to be done to continue with the validation, that is covered in Chapter 6. The validation is a more subjective process, executed by testers, to check if the functionality meets the intended behaviour and covers the question: Are we building the right product?

Software testing is described, by the IEEE, as the process to detect discrepancies between existing and required conditions and to evaluate the features of the software item. The Institute prescribes several protocols for the development and testing of various kinds of software. These documents have provided guidance for the development and validation of the software prototype, but do not meet the requirements and protocols of the IEEE standards. (IEEE 1998)

5.2 Testing methods

There are different types of testing: (Copeland 2004)

1. *Unit-testing*: Testing of isolated software units.
2. *Integration-testing*: Testing the interaction of combined software parts.
3. *Functional/System-testing*: Testing of the complete, integrated system.
4. *Acceptance-testing*: Testing to verify that the acceptance criteria are met.
5. *Regression-testing*: Testing the existing functions after implementing new ones.

This thesis covers a combination of these methods.

5.3 Modules

Module Boundary conditions:

The requirements composed for this module, ensure the general applicability of roofing envelopes and the visual feedback of this input.

Verification: For the simple, rectangular shape, all requirements are fulfilled. Some features, however, require further development. At this moment, the automatic function to find the shortest span and dimensioning of the structure is not suitable for more elaborated shapes. The most important feature, implementation of all possible perimeters, is working well. As described in previous chapter, there are different algorithms to facilitate this process.

The boundary condition module (BC) functions as the base of the design and structural analysis. Therefore, the interconnectivity is important and will save unnecessary repetition of work. We limit ourselves to simple sanity checks for the internal collaboration of the individual and combined modules. The perimeter of the subsequent module, the base geometry, responds automatically to changes in BC with a regeneration of the geometry. The same applies for the input and output of the load generator, optimization and analysis modules. An increase of the span, causes an increase in the loads, cross section, stresses and deflection, as it is supposed to do. Hereafter, a more elaborated check concerning

the structural analysis is performed to see whether the analysis, mainly performed by Karamba, is reliable.

For verification, we also review the interoperability with other programs, since the geometry cannot be inserted in Grasshopper right away. An additional action is needed: Import of the geometry from software X into Rhino's interface and from Rhino into the tool. Architectural and structural drawings are often produced in programs like AutoCAD (*.dwg/ *.dxf), Revit (*.rvt) and Adobe (*.ai/ *.pdf). Files from AutoCAD, SketchUP and Adobe can be imported straight into Rhinoceros and there are several methods to import from Revit. DWG ACIS solids, for example, gives the user a lot of influence. The model can simply be exported from Revit into a .DWG file and imported in Rhino's interface. It is not possible to preserve materials, but objects can be assigned to different layers. For optimal use, it is important to keep a low level of detail and limit the selection to what is necessary.

Requirements	Test	Remark
REQ-1	Pass	-
REQ-2	Pass	Limited number of parameterized input
REQ-3	Pass	-
REQ-4	Pass	Limited to rectangle outlines
REQ-5	Pass	Not automated for all perimeters, manual input is possible
REQ-6	Pass	-
REQ-7	Pass	Limited to simple outlines
REQ-8	Pass	Limited to simple outlines
REQ-9	Pass	-
REQ-10	Pass	-

Figure 5.1: Verification test of boundary conditions module.

To verify the interoperability, we used a complicated outline that was drafted in Autocad and represents the streets in the Hague's city center, (see Figure 5.2). This went quickly and was easy to implement in the tool. However, it became clear that this module is not yet fully automated; we had to manually assign a surface between the facades. The knowledge of scripting was needed to select the right elements. Depending on the original geometry file and cleverness with Grasshopper this can be time consuming, therefore it is recommended for the inexperienced user, to limit the geometry solely to the curves that will be used.

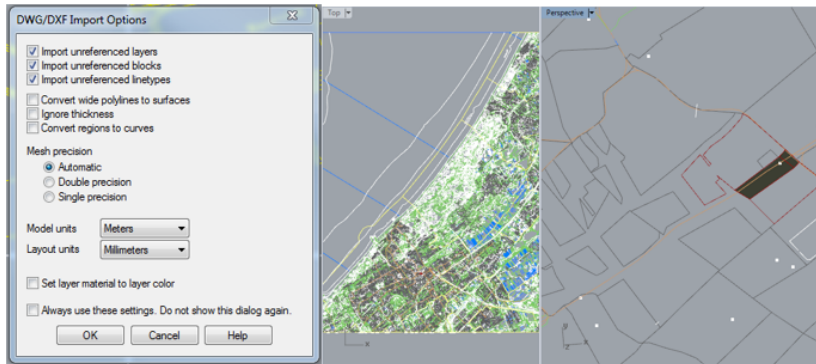


Figure 5.2: Import AutoCAD to Rhino and GH.

Conclusion: The module for boundary conditions offers the possibility to enter all imaginable roofing peripheries and a good visual representation of these perimeters. Further attention should be given to the automatic detection of the implemented geometry.

Module Base geometry:

The requirements for this module are set to provide the designer a quick overview of the design possibilities. An important feature is to offer the opportunity to play with the variables while providing instant visual and numerical feedback.

Verification: All algorithms are tested separately and offer a vast variety of possibilities. There is a choice between several geometries and each geometry has multiple design variables. Each adjustment gives instant visual response and the adjustments are interactive with the other modules.

The working of the module has been extensively described in Chapter 4, we see that the module already offers the user different possibilities to play with the structural design. The user can vary, for example, the center to center distance in both directions, the height of a structure, adjust the mesh and height of a catenary structure and can even manually adjust the geometry in every possible way. To check the interactivity, we look at the discrepancy between the expected and occurring consequences of the other modules in response to a change of design variables in the base geometry. An increase in the loads per element and so the cross section will occur, when the center to center distance increases significantly. Also, a change is expected of the load path,

when secondary beams are added or the mesh of a structure is adjusted. If the catenary height changes, it is expected to influence the horizontal and vertical reaction forces of the supports and the load distribution of the structure. When tested, all assumptions are in line with the working of the model.

Further verification of this module will be covered more extensively during the case study, where we will show the added value compared to a more traditional approach.

Requirements	Test	Remark
REQ-11	Pass	-
REQ-12	Pass	-
REQ-13	Pass	-
REQ-14	Pass	The direction of the beams is not yet easily adjustable for all perimeters
REQ-16	Pass	-
REQ-17	Pass	-
REQ-18	Pass	Further elaborated in the case study
REQ-19	Pass	-
REQ-23	Pass	-
REQ-24	Pass	-

Figure 5.3: Verification test of base geometry module.

Manual input is possible as well, but depending on the user's experiences, might be time consuming. It is also possible to import a design from other software, but as yet there is no automatic conversion, however. A standardized algorithm to recreate a design is still limited to very specific options (both simple and complicated).

Conclusion: The test case will tell if this module offers a helping hand for the early design process, to see if it can compete with the pencil as a tool for an extension of the brain. But we can already conclude that even a user without a 'structural' background can directly see the impact of a change and therefore gain a better understanding of a structure's mechanics and explore the boundaries of the different possibilities.

Module Structural properties:

From a structural engineer's point of view, this module represents the heart of the tool. The part where the engineer can influence the design and manipulate the structure to fulfil its demands. Seen from a designer's point of view, we could call the base geometry the heart of the tool and this module is expected to work in a conservative, safe manner

to facilitate the designer.

Verification: The user's requirements for this tool are conflicting. On one hand, conservative standardized values are desired. On the other hand, the possibility for optimization requires freedom of the factors. If the values provided by the tool are sufficient for the structural design, we will check by comparing it with the features of a regularly used program: Technosoft Raamwerken version 6.07 (build 2046), in short: TS:

Materials: Technosoft offers the freedom to choose either steel, concrete, wood, aluminium or a material added by the user. The choice for steel, concrete and wood are connected to the most common libraries and can be specified for different qualities. The materials are characterized by the value for the Young's modulus, density and the Poisson's ratio. The prototype already offers the possibility to choose different types of material and it's quality as well. This is done with the material selection component provided by Karamba, the values for this tool are set to glass.

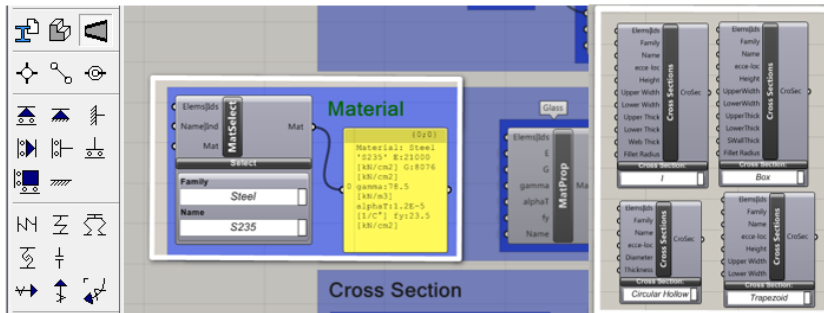


Figure 5.4: TechnoSoft menu (left) and Tool input (right).

Cross section: Technosoft also provides libraries for the most commonly used cross sections and to enter these manually. There is a function to add tapered profiles that morph from one cross section into the next. Technosoft liggers (version 6.22) also offers a function to assign cross section over several sectors of an element. In the tool the use of cross section libraries is limited, for the ease of use it is recommended to extent the selection of cross section by providing libraries for the different common types, that will allow the user to select some predefined cross sections. However, the possibilities to add a cross section are very thorough. Some standard shapes, the I, Box, circular hollow

and trapezoid cross section can be implemented easily and there is a component that can generate and modify all kinds of cross section with the help of 20 variables for different properties; although the surface (A), moment of inertia (I) and Young's modulus (E) of a cross section are already sufficient. The standard value for the tool is a rectangular glass cross section, with the smallest height set to 100 mm: the starting value for the optimization. For future development eccentricity of a cross section could easily be assigned to an element and reinforced cross sections should be added.

Supports: Just like TS, the prototype offers the choice between a hinged, roll or clamped support. In programs like TS, supports are assigned manually. The advantage of a parameterized tool is that this can be done automatically. At the moment, the tool assigns the supports at the cross section of an element with the perimeter. It is also possible to assign the supports manually, however, a understanding of the programming and process is needed. For future developments, extra algorithms should be introduced to offer more choice for the automatic supports placement.

Technosoft offers three more features that are often used; hinges, springs and induced displacements and rotations. These features become more important in the detailed design phase and are therefore not implemented in the prototype. The engineer has the freedom to add hinges and springs manually, which are already implemented in the script. For this verification, we checked if this could lead to problems, when used without proper knowledge, but the analysis stops when the structure becomes Kinematic indeterminate and changes into a mechanism.

In order to check the interactivity of this module, we varied the input of the perimeter: the supports and definition of the elements changed accordingly. The same goes for adjustments in the base geometry. For the interactivity with the loads and analysis we changed the settings of the supports. The simple supported beams have a moment distribution and deflection as can be expected. The same goes for the beams clamped on two sides and the cantilevering beam. For the link with the optimization module, we lowered the yield strength and observed how bigger cross sections were assigned.

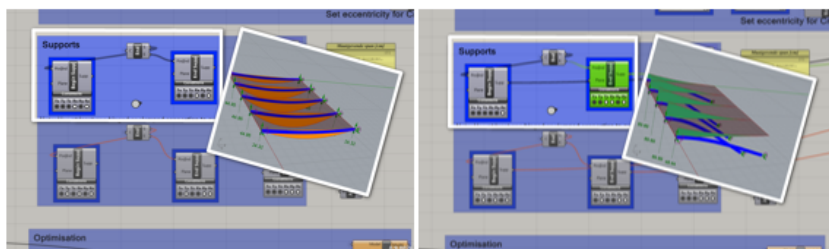


Figure 5.5: Verification of interconnectivity.

Requirements	Test	Remark
REQ-25	Pass	-
REQ-26	Pass	-
REQ-27	Pass	-
REQ-29	Pass	-
REQ-30		Should be extended more for further developments
REQ-31		Future?
REQ-32		Should be elaborated upon
REQ-33	Pass	-

Figure 5.6: Verification test of the structural properties module.

Conclusion: It is safe to conclude that all elements of this module are linked to the other modules. For future developments, it is important to introduce a reset function; to set the variables back to the standard value. In the current version, this is restricted to the parameterized input, like the sliders and value lists, but in the nearby future we recommend standardizing the settings of all important components. This is especially necessary, when used by someone without a structural background, to enhance the safety.

Module load generator

The most important feature of this module, is the interactivity with the parametric structure and the freedom for the user to apply, adjust and remove loads; also to offer the potential for implementation of the Eurocode.

Verification: As for the module for the structural properties, we will make use of other programs to verify the working of the tool. If we compare TS liggers with TS Raamwerken, it becomes clear that, due to the extra dimension, TS Raamwerken has more features concerning the load distribution on a design. Therefore we take this for benchmarking.

TS provides exact assignment of the type of loading, the location of the loading and its properties. The codes of different countries are implemented in the script and can be related to the loads; the user only needs to specify that it concerns a permanent or variable load and if so what kind of load.

Implementation of the Eurocode is time consuming and therefore it was neglected to focus on more important parts of the script. For the first tool setup, we have chosen to use a permanent load of the gravity and loading for the roof plates, which are not accounted for otherwise. We also added a variable wind load and added a load combination with the factor for consequence class 2.

Karamba supports the following types of loads: gravity, point, imperfections, pretension, uniform distributed mesh loads and prescribed displacements at supports. An arbitrary number of these loads and one gravity load may be combined to form a load case of which an arbitrary number may exist.

Requirements	Test	Remark
REQ-34	Pass	-
REQ-35	Pass	Easily extendable
REQ-36	Pass	-
REQ-37		
REQ-38	Pass	-

Figure 5.7: Verification test of the load generator.

Conclusion: Further implementation of the Eurocode is a must.

Module structural optimization

In order to be implemented in the design process, it is important that this module reacts instantly on the changes of the user and can provide instant feedback.

Verification: We see that the optimization module automatically assigns cross sections to elements or parts of elements, based on the load bearing capacity. Furthermore it is possible to set the maximum deflection as a restriction. For the prototype, that focuses on the initial design phase, is chosen to use conservative values with a factor of 0.8 for both the load bearing capacity as well as the governing deflection; this can be changed by the input for “MaxUtil” and “MaxDisp”. It is also possible

to select different parts of the structure to be optimized, this can always be a designers choice.

As the tool only shows the outcome, we give a brief description of the procedure:

1. Assign the first cross section in line.
2. Determine section forces and deflection at “nSamples” points of the beams.
3. Select the first sufficient item on the list.
4. Repeat step 2 and 3 until no other profile is selected at 3, until the iteration has reached the biggest section or until “DispIter” iterations are executed.

A notification appears in the output section whenever the load bearing capacity of the biggest cross section of the list is insufficient.

The optimization list should be ordered from best to worst choice; this can be done either by the height of a profile but also the costs could be leading, or another motivation. The sequence is important, because the tool assumes that the most desired section is placed first. It is possible to use different types of cross sections, so one could also use a combined glass I profile.

Freedom of settings provide choice in the amount of maximum iterations for both the ultimate- and serviceability limit state and the number of samples among the beam where cross section forces and resistance are compared. It's also possible to choose for an elastic or plastic cross section design or to change the material safety factor. It is required to implement the buckling length in the algorithm to provide feasible cross sections.

Figure 5.8 shows a simple optimization example of a cantilevered beam with a uniform distributed load. The optimization module places the higher sections, closer to the support and smaller sections at the end, where the bending moment is reduced to zero. The situation on the right shows that the number of steps between the smallest and biggest section have more than doubled.

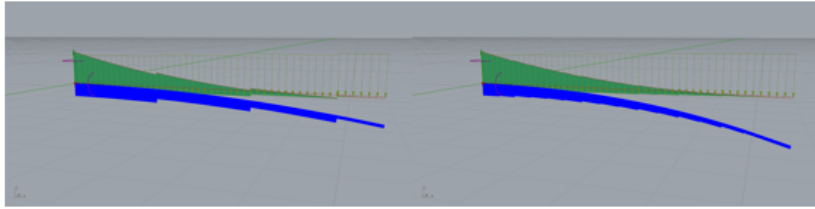


Figure 5.8: Element optimisation.

It is possible to set the element size very small. However, this will lead to big local differences at points with peak stresses. However, the outcome can be used as design help. The user needs to keep in mind that this is the minimum cross section that fulfills the prescribed factors; it is always possible to deviate from this, when the cross section increases.

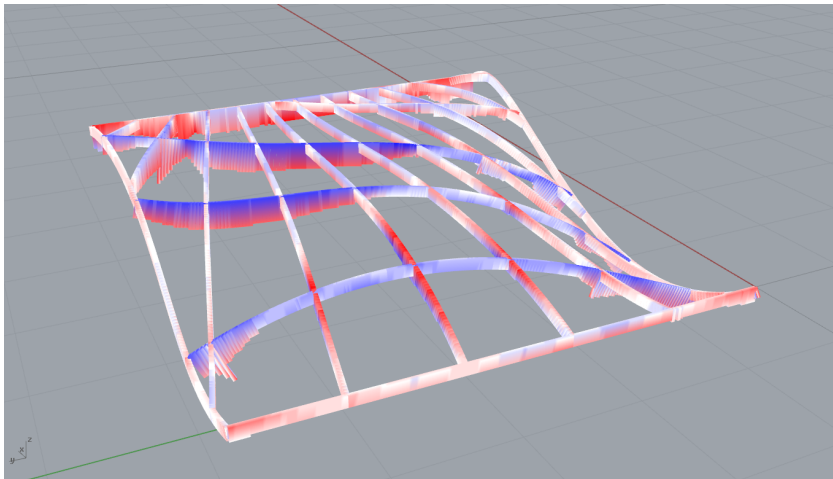


Figure 5.9: Big cross sections assigned near peak stresses.

Requirements	Test	Remark
REQ-39	Pass	-
REQ-40	Pass	-
REQ-41		
REQ-42		
REQ-43		

Figure 5.10: Verification test of the module for optimization.

Conclusion: The output of this module is the optimized cross sections, including warnings when these are insufficient, with notifications for the mass of the structure, the internal energy of each load case and the maximum displacement of each load case at the end- and mid-points of the elements.

Module Analysis:

The most important part of the module is the reliability of the analysis, but also the interconnection with the different modules, and like the other modules, the instant visualization of the outcome.

Just as with the optimization algorithm, the utilization of the elements are determined according to the Eurocode 3 EN 1993-1-1, buckling and lateral buckling are considered. To check the reliability of the analysis elements of Karamba that are implemented in the tool, we look at three common cases that we can compare with several ‘forget me nots’, simple hand calculations. The outcome can be seen in the appendices. In addition to this basic check, the outcome of the tool could be compared with the results of any structural design software. A comparison of these results shows a compelling resemblance with the findings of the algorithmic tool.

Requirements	Test	Remark
REQ-44	Pass	-
REQ-45	Pass	
REQ-46	Pass	-
REQ-47		Should be elaborated upon
REQ-48	Pass	

Figure 5.11: Verification test of the analysis module.

5.4 Conclusions

The prototype fulfils practically all the requirements drafted during the development strategy and thus offers the user an abundance of possibilities. As long as this person is willing to adhere to the supplied options and explore these possibilities.

The tool has the capacity to check extreme scenario's and to prevent mistakes due to malicious design, mistakes with loading or inexperience with a specific material. However, at the same time this becomes a disadvantage, when a user interferes with the original script, without proper knowledge and competence.

More extensive testing, comparison with hand calculations and FEM models is needed, to check extremely complicated designs, to test the boundaries and to see up to which level the tool provides reliable results.

Briefly, the tool shows high potential to function as a design aid on different levels throughout the process. The usability for everyday use should be tested.

Chapter 6

Tool validation and testing

For this chapter, we step into the role of the tester and validate the tool with help of a case study. As described in the previous chapter, this validation focusses on the question; 'Are we building the right product?'.

6.1 Design

In the literature study the potential and limitations of glass as structural material and the importance to provide safety for the design were explored. The tool should facilitate the development of innovative design, exploring the limits and boundaries of structural glass.

Design principles

One of the intended qualities of the tool is evidently the multi applicability; therefore it can be tested for a large amount of scenario's. In consultation with the graduation committee it was decided to apply the tool on a challenging test case, bridging a wide span. The tool will be tested for a cover at the broadest part of the Grote Marktstraat in The Hague. This could function as the entrance of the subway, which is currently quite anonymous. The application of the testcase in the parametric design tool has been translated into the following design principles:

- The roof cover is self-supporting and should provide for its own stability.
- The roof spans at least 30 meters, placed 4 to 20 meters above street level.

- The structure will provide an open space, all supports (columns) are placed in line with the facades.
- Load carrying elements are executed in glass.
- Lateral stability is provided by the glass panes.
- The load transfer to the facades should be spread.
- The cover will highlight the entrance of the subway.
- The project is situated in the urban inner city of the Hague.
- The load on the structure is assumed to be as described in the Eurocode, complemented with the Dutch Annex.
- Structural calculations 6208 en the additional research currently conducted at the TU Delft.
- Focus on the conceptual design process.
- Material properties from literature study, with design strength of 40MPa en a safety factor of 2 for this fictitious design assignment.



Figure 6.1: Scenario of the case study.

Configurational assumptions

The project is located at the city center of the Hague and is meant to highlight the entrance of a subway, which is now quite concealed. The wide street profile poses a challenge to test the extreme possibilities of the tool.



Figure 6.2: Situation of the case study.

6.2 Algorithmic design approach

Boundary conditions

The boundaries are determined by the location. The margins for the location of the supports are shown in the image below, there is the possibility to play with the exact location. We added sliders to the boundary condition points for the vector in z-direction. Later on it will become clear, how this will influence the design. The coordinates of the supports are kept flexible throughout the entire design process.



Figure 6.3: Restricted zone for supports.

Base geometry

For this test case, we generated a large number of alternatives, with straight glass beam elements and both single and double curved glass panes. The minimum height is restricted by the free passage of fire trucks and is set to 4 meters. There is no limit to the maximum height, this is depending on functional, structural and aesthetic considerations.

For this specific situation, we added an algorithm to mesh the base geometry and automatically locate the elements at the facade sides. We want this to be reassigned automatically when the mesh is altered (characterized by its UV partition). To accomplish this, we create a pattern, U+1, V-1, V-1, U+1, that can partition the list of naked points (point locations of vertices not surrounded by faces). These points do not need to coincide with the supports of the structure.

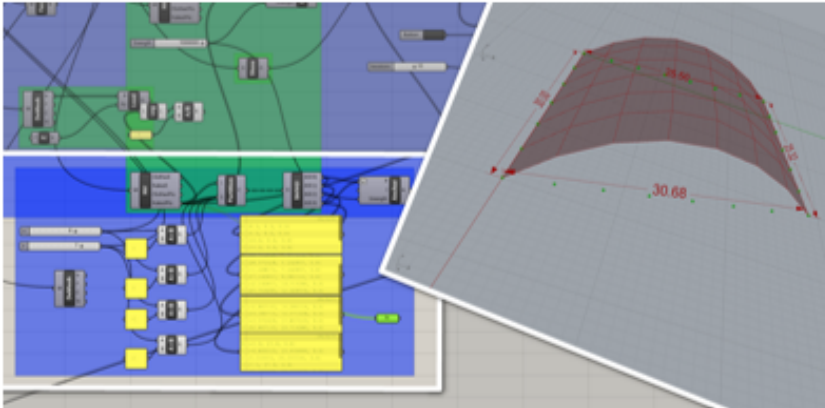


Figure 6.4: Automatic support at the facade.

Structural properties

We can conclude that the initial values of the prototype are sufficient for this tool design.

Supports – As described before, the supports are automatically assigned to all sides of the structure; however, it is only allowed to place columns or supports on two sides of the envelope, coinciding with the two facades. Therefore we need to adjust the script to automatically select the supports corresponding to the ones parallel to the facades in the same manner as the adjusted algorithm of the base geometry module. By default, all beams that intersect

with the facade function as supports. However, for the case study we added the possibility to play with this parameter. It is possible to support all beams, only to corner points or something in between. For this test case, we experimented with the possibilities of the base geometry.

There is a considerable influence on the sequence of design. Discarding supports in the catenary engine will create high deformations and a vault like structure, with relatively low cross sections. If the supports are removed after the form finding process, high beams in the facade are needed to span from support to support. These edge beams could be easily replaced by steel elements. Corresponding supports for the form finding module and support module, generally produce the most efficient structure.

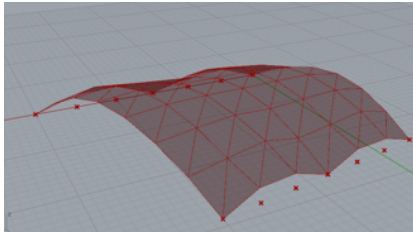


Figure 6.5: Form finding discarding 8 potential supports.

The type of supports appointed in the module of boundary conditions, are simply supported and don't need further action. The choice of type of connections will have consequences for the detailing of these connections, just like other materials and will be discussed later in this chapter.

Material – the assumption for material properties for the test case and that of the tool design arise from the same literature study and are therefore similar. In case of diverging values it is easy to adjust the aforementioned sliders.

Cross section – we use the optimisation module for appointment of the dimensions. Further we use the default settings with rectangular profiles and combinations of the available glass thicknesses as can be seen in the literature review.

Load generator

For glass design the Eurocode turns out to be insufficient, therefore the Dutch NEN 2608 will be leading. However, the NEN-EN 1991-1, the

norms for loading in general, will be used to determine the load cases. The considered loading for the tool is summarized below:

The self weight of the glass panes and the self weight of the beams
 $G = t \text{ [m]} \times 25 \text{ [kN/m}^3\text{]}$

The wind load is determined based on wind zones for canopies with an angle of 30 degrees as worst case scenario. The structural factor $c_{scd} = 1$ for regular, low-rise buildings. C_f = force coefficient for wind zones for canopies with free flow of wind ($\varphi=0$) the overall force is depending on the angle of the roof. With maximum values:

A	+2.2/-3.0
B	+3.2/-3.8
C	+2.4/-3.6
and overall force	+1.2/-1.8

$Z_e = h = 15 \text{ qp}(z_e) = 0.80$
 for Urban regions situated in area II

$F_i = 1 \cdot (-3.8 \text{ V } + 3.2) \cdot 0.8 = -3.04 / 2.56 \text{ kN/m}^2$
 Overall force $F_i = +0.96 / -1.44 \text{ kN/m}^2$

The factor for wind friction: cfr 0.01

The snow load depends on the geometry of the structure. Because snow does not run off like water, accumulation of snow will occur and can cause high loads; $0.7 \cdot 1.6 \cdot 1.0 \cdot 1.0 = 1.12 \text{ kN/m}^2$ is the worst case scenario

As maintenance (1 kN/m^2) is highly unlikely to occur simultaneous with snow loading, we consider the snow load as leading. Composition of the load combinations are stated in the appendix as well and are summarized in the figure below.

Currently, the standard load combinations are in line with the ones in the test case. It should be made possible to use deviant consequence classes and load factors without manually adapting the load combinations. The exact values of the imposed loading is facilitated with num-

ULS	Load case	safety factor
1	self-weight only (G)	$1,35 \cdot G$
2	self-weight plus snow load (Q _{sn})	$1,2 \cdot G + 1,3 \cdot Q_{sn}$
3	self-weight plus wind upwards (Q _{w+})	$1,2 \cdot G + 1,3 \cdot Q_{w+}$
4	self-weight plus wind downwards (Q _{w-})	$0,9 \cdot G + 1,3 \cdot Q_{w-}$
SLS		
5	self-weight only	$1,00 \cdot G$
6	self-weight plus snow load (Q _{sn})	$1,0 \cdot G + 1,0 \cdot Q_{sn}$
7	self-weight plus wind upwards (Q _{w+})	$1,0 \cdot G + 1,0 \cdot Q_{w+}$
8	self-weight plus wind downwards (Q _{w-})	$1,0 \cdot G + 1,0 \cdot Q_{w-}$

Table 6.1: An overview of the load cases.

ber sliders and therefore accessible and simple to adjust for the case study.

Structural optimization

We apply the algorithm as described in Chapter 4 Tool development, for the test case. The specific and standardized boundaries of values are:

- Height – ranging from 100 to 1000 mm divided in 30 intervals.
- Width – the width of the cross section increases non-linear with the height and is rounded up, to combinations of available thicknesses.
- Maximum utility – utility of the elements multiplied with a factor 0.9
- Maximum displacement – the deflection (maximum span \times 0,004) multiplied with a factor 0.9.

Structural analysis

For the analysis of the structure we use the module as described in the tool development and verified in the tool development chapter. We adjust for the height and number of beams until the notification of the outcome report shows that there are no more cross sections that fail the requirements.

In the previous chapter, we concluded that these values are reliable and conservative and therefore it is assumed to be oversized.

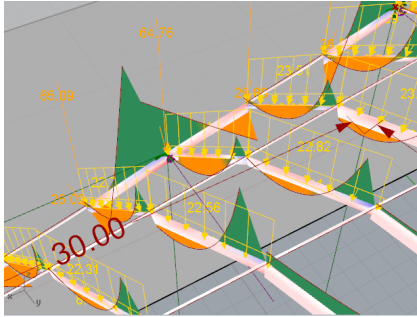


Figure 6.6: An example visualization of the analysis applied to the structure.

As mentioned before, additional calculations concerning buckling and lateral buckling are required and the roof design should also provide for a proper water drainage to avoid water accumulation. In this case the overall shape prevents water accumulation.

Design alternatives

Figure 6.7 illustrates a selection of design alternatives generated in just minutes with the help of the catenary engine.

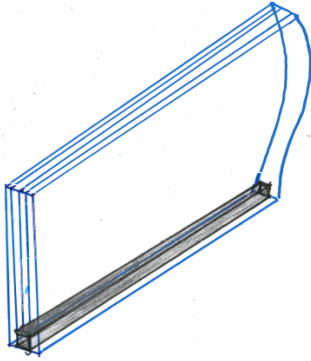


Figure 6.8: Reinforced main beam

Detailing

It is wise to design connections that gradually introduce the loads from one element into the other. Therefore we prefer embedded connections, laminated into the elements. One of the design restrictions is the degrees of freedom of the connections. In the main direction, the elements have no rotational freedom. This will be visible in the connections, which need to be larger than for hinged connections. In the secondary direction, coupling is provided with hinged joints. Stability is provided by the roof planes, connected with a structural sealant.

Furthermore, we consider it practical to use demountable connections for maintenance, replacement and reuse. It is important to keep the realisation and construction approach in mind during the detail design. More deviant elements generate more room for mistakes.

To construct this design, the two smallest outer arches will be placed with temporary supports. Then alternatively the secondary beams and roof planes are placed, followed by the next arch until the entire structure is completed.

Figure 6.9 shows a first design concept for three typical points: the hinged support, continuation of a main beam and a section where four beams come together. The supports are designed in such a way to visually disappear in the faade. The main beam is split into different sections and the secondary beams hang on these elements. The glass is protected with a soft neoprene interlayer between the shoes and the glass.

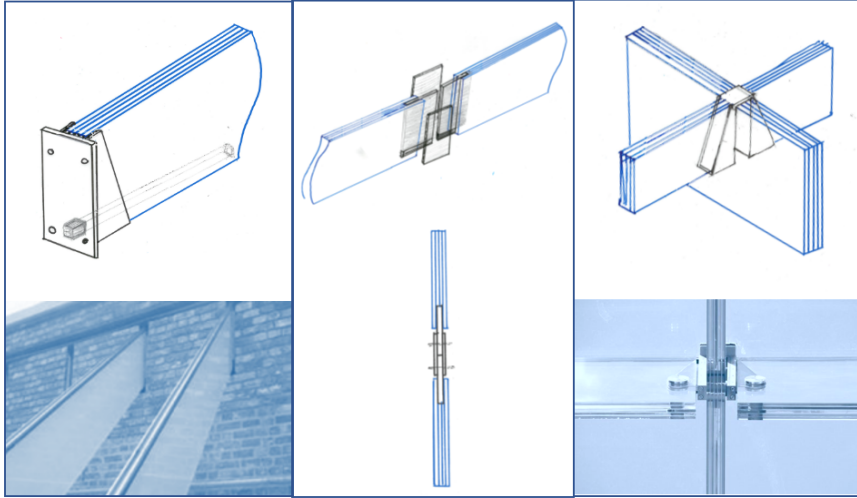


Figure 6.9: Left: supports and detail of the Victoria and Albert museum Middle: continuous beam Right: intersection

6.4 Further development

This research study is a proof of concept, designed to function as base of a much more extensive tool. Since this tool is a prototype, no self-contained program, further development is essential. During the process, it is assumed that extension of the tool is easily done. To be able to evaluate the feasibility of the tool and further development, the adaptability is explored with help of two additional case studies.

Case study 2

Appendix D shows the expansion of the prototype, applied to a second test case: a roof cover for a conservatory at the faculty of Architecture in Delft, covering 50x30 meters. The expansion of the prototype make the addition of mid supports necessary. Besides the supports at the perimeter, that support all intersections with the glass beams, supports at the middle of the roof are required. This is done by adjustment of the structural properties module as shown in Figure 6.10. Four points are introduced that indicate the intended position of the internal beams in the 2-dimensional space. The element 'line to beam' is used to extract the end points of the beams produced during the form finding process described in the appendices. These points are then used as attractors for the supports, so that these are automatically assigned to a junction.

Necessary because the exact location may change due the parametric adjustments. Finally, the output of the additional support component is connected to the support collection component in order to be fully implemented in the tool. The main beam is split into different sections and the secondary beams hangs on these elements. The glass is protected with a soft neoprene interlayer between the shoes and the glass.

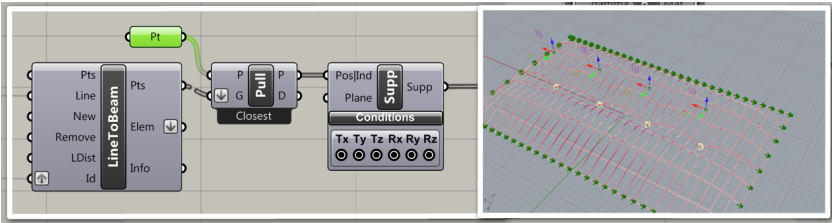


Figure 6.10: Implementation of mid supports.

Case study 3

Appendix D shows the application of the tool for the third case study, used to explore and exploit the functionality of the tool. To check the feasibility of future developments the third test case involves a pavilion, covering a larger surface, executed in timber. Compared to the previous test cases, this case study pushes the boundaries of the supports even further and a construction method is implemented, that allows the pavilion to be built up from one intertwined surface of elements, pushed up into place from under. The module structural properties is used to implement the material properties and cross section for wood. For safety reasons conservative values for the yield strength are used. The material properties are implemented as follow:

Young’s modulus E:	1050	[kN/cm2]
Shear modulus G:	360	[kN/cm2]
Density Gamma:	6	[kN/m3]
Thermal expansion, alphaT:	5.0E-6	[1/C°]
Design strength $f_{m;0;d}$ = :		
$f_{m;0;rep}k_{mod}/\gamma_M = 400,85/1,2:$	28	[N/mm2]

The cross section is adjusted as well. In order to create a more feasible structure, the edge beam will be executed with an increased cross

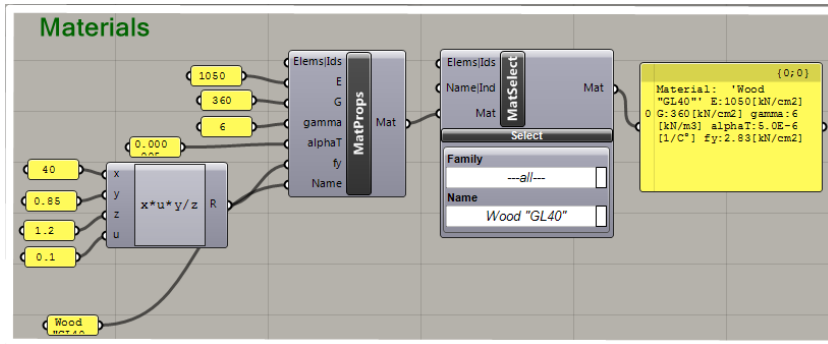


Figure 6.11: Implementation of the mid supports.

section and the inner elements are provided with a possibility for 10 cross sections, connected to the optimization module. The elements are distinguished with a family and name tag.

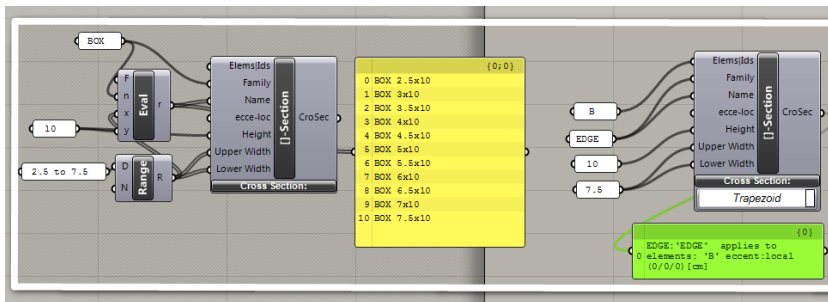


Figure 6.12: Implementation of the cross sections.

The same method as mentioned in the second case study is used to add the mid columns for the pavilion. However, for this occasion, with the flexible timber components, the edge beams will also be part of the form finding. Therefore the supports of the perimeter are locally interrupted to create openings for entrances; this is accomplished by addition of multiple cut patterns, connected with number sliders to vary the number and size of openings in an intuitive manner. The image below shows the support pattern generated with this extension.

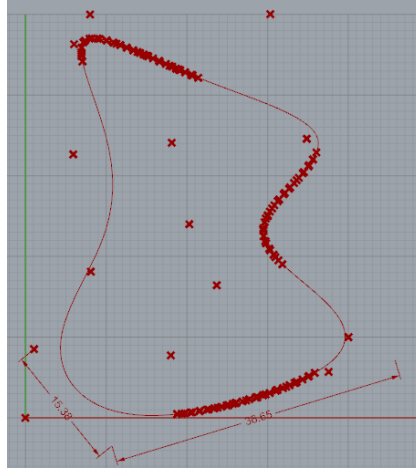


Figure 6.13: Support pattern.

The user can influence the form finding process by adjusting the mesh size and the location of the supports. Another function that is implemented in the prototype is triangulation of the mesh, which offers higher stability and a change of looks. It is possible to use all kinds of grids and patterns, every pattern is connectable to the current module. We prefer the triangulated grid due to its high stability and feasibility regarding construction. The pattern of the elements is defined in the part of the script that can be seen below. The effect of the different patterns is shown in the appendices.

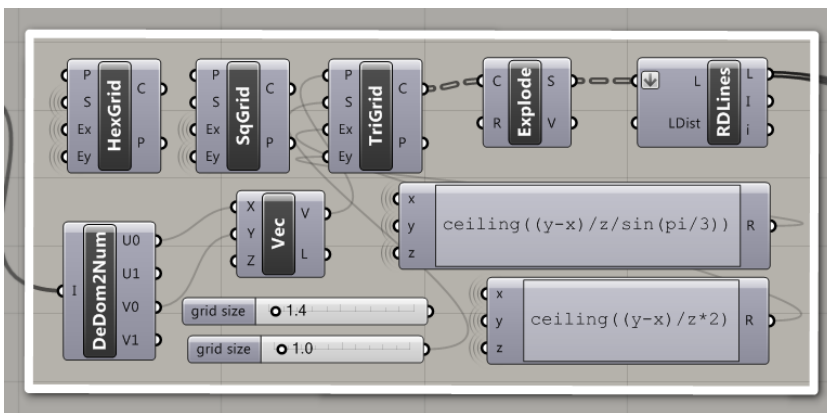


Figure 6.14: Implementation structural pattern in script.

During optimization, the complexity of the structure caused difficulty with finding a sufficient structure. Eventually this was done with intuitively replacing of the supports.

6.5 Conclusions

The tool offers a lot of possibilities for structural glass design and provides awareness of certain pitfalls. One of the most important aspect of glass design, the detailing is not provided by the tool. Therefore, attention should be paid to the implementation of warnings for design consequences and to create awareness of the importance.

The first test case shows that the prototype already contains 'knowledge', that the average architect respectively engineer does not possess. However, it also shows, that much more knowledge regarding structural glass, its possibilities and limitations, should be implemented to extend its functionality. The prototype seems to have potential to be a platform, to explore the opportunities within the limitations in a conservative and safe manner. The knowledge can be implemented, responding to the modular structure, and impediments could be automatically resolved. This can be conceived by a module applying joints with corresponding connections and segmentation, automatically every 6 meters or less.

The third test case showed that with an increased complexity, the generation of a feasible structure becomes more difficult and demands a structural background. The second and third test case show that the modularity of the script contributes to the simplicity of the extension. Although the prototype was effortlessly expanded, it was not a matter of copy pasting of new modules since everything is interconnected. Both the prototype and extension are developed by one person. Therefore, the ease of extension by a person not primarily involved cannot be confirmed; prior knowledge of Grasshopper and the specific script of the prototype seems necessary, due to the complexity of the tool.

Chapter 7

Discussion

This chapter discusses the process, results and application of the algorithmic design prototype.

In 1.5 we stated the main research question, thesis objective and the research objectives:

Thesis objective

The objective of this research is to investigate the possibilities of creating an algorithmic design for structural glass covers, to create an innovative methodology with a broad spectrum of potential applications. The methodology should stimulate the easy use of glass as a structural material. The methodology should contribute to the bridging of the gap between architects and structural engineers and take advantage of the potential of parametric design for architectural and structural purposes.

Research question

In order to achieve the objective of this thesis a research question is formulated:

‘How to develop a prototype for an algorithmic design methodology, considering geometrical control with structural feasibility, intended for structural glass covers in the conceptual design stage, with the opportunity for further expansion?’

Research objectives

Seven sub objectives are composed to answer the research question.

Section 7.1 reflects on the achievements of the prototype, this leads us

to an answer on the first part of the Thesis objective. In order to answer the Research question, Section 7.2 reflects on the research objectives in relation to the conducted research and development; it looks back on the research process and discusses the two main objectives of this thesis: Bridging the gap between architects and structural engineers and the capacity to stimulate the easy use of glass. Section 7.3 summarizes the limitations of the prototype and the intended tool. Chapter 8 formulates conclusions on the Thesis objective.

7.1 Reflection on achievements

A first prototype for the algorithmic design tool has been developed. It offers architects, structural engineers and third parties involved an instrument to create and check (structural) designs of glass roof covers. It involves automatic optimization and assignment of cross sections, loads and it performs structural analysis. The tool provides the user a quick image of the consequences of different possibilities regarding structural feasibility and aesthetic qualities. The tool thus gives more freedom in the artistic design. It is possible to both generate different design alternatives in a short amount of time and perform more extensive analysis in a specific case. The following aspects can be discussed regarding the functionality of the tool:

- Grasshopper offers a very intuitive and visual approach of scripting in either predefined components or individually scripted elements. Grasshopper has a steep learning curve, thus attractive for inexperienced users.
- The tool offers insight in the flow of forces and automatic dimensioning. This allows the (inexperienced) designer to create and explore with a more solid structural foundation.
- The tool assists the decision making in the conceptual stage and can be extended to support decisions throughout the entire design process upon realization.
- The tool can support both architects and engineers as well as other parties like manufacturers, authorities, researchers and students. The tool supports these parties with the possibility to rapidly generate alternatives, to analyze specific designs and to compare design alternatives.
- The tool provides the user a quick image of the consequences of different possibilities regarding structural feasibility and aesthetic

qualities. This possibility to investigate the influence of different design parameters, enables the user to make better informed decisions, throughout the most critical part of the design process.

- The integrated design approach is not limited to the scenario of the case study, as described in chapter 6, but is applicable to several design scenario's and is more widely applicable.
- The tool supports structural optimization; the module for cross section optimisation already functions on all possible structures generated in the prototype.
- Verification showed no significant inconsistencies or large discrepancies from the maximum allowed tolerances.
- One of the most important conditions of the tool is the instant feedback of design choices and is already implemented in all facets of the tool.

And the following aspects, more specific about the application of glass structures:

- The tool offers the user assistance for glass design, by automatically assigning thicknesses in available combinations and assigning material properties.
- The tool provides user awareness for the difference of glass design from more traditional construction methods, with notifications linked to design choices.
- The segment for catenary generation makes it possible to create glass structures with a big span.
- The tool proves to be helpful in the design process for a specific glass roof cover and looks very promising; however there is still need for further enhancement.
- Especially concerning glass design, not all necessary variables are implemented in the prototype. Expansion of the specific glass variables and measures are needed in order to be sufficient for structural glass design. A lot of detailed and extensive completion is necessary as described in the recommendation.

7.2 Reflection on research objectives

During the research-process the seven objectives turned out to be valid to create a sound foundation for the tool and they are stated below:

‘Provide a theoretical framework concerning the structural glass and algorithmic design.’

The results of the research concerning structural glass are summarized in the theoretical framework, Chapter 2, and the results for the research for algorithmic design are implemented in the strategy for tool development, Chapter 3.

‘Define which requirements, concerning functionality, usability, reliability and performance, should be accomplished by the methodology.’

With the knowledge of the literature study, the requirements were drafted, and an important part of the tool development strategy was formed.

‘Define a general outline for the algorithmic method or model taking the requirements into account.’

Substantial time was invested in this flowchart and it turned out to be of good guidance for the developing the general structure.

‘Implement the theoretical framework into the defined general structure.’

The thorough flowchart was quite literally adopted into the Grasshopper script. Slight interaction arose between the flow chart and the development of the script.

‘Validate the outcome of the produced methodology.’
and

‘Investigate the impact of the methodology on the workflow / functioning of engineers and designers by implementation on a case study.’

The case study was performed to investigate the impact of the tool, which gave a good impression of the role that it could have.

‘Determine the required successive steps to obtain a fully functional method/model.’

This objective to determine the successive steps needed to obtain a functional model, is discussed throughout the thesis, as can be seen below

in the recommendation section.

All these research objectives were used to fulfill the main research objective:

‘To develop a prototype for an algorithmic design methodology, considering geometrical control with structural feasibility, intended for structural glass covers in the conceptual design stage’

A first prototype for the algorithmic design tool has been developed. It offers architects, structural engineers and third parties involved an instrument to create and check (structural) designs of glass roof covers. It involves automatic optimization and assignment of cross sections, loads and it performs structural analysis. It is possible to both generate different design alternatives in a short amount of time and perform more extensive analysis in a specific case.

We can now discuss to which extent the thesis objective has been met. The tool proves to be helpful in the design process for a specific glass roof cover and seems to be very promising. However, there are still opportunities for further enhancement (7.3).

The tool provides the user with a quick image of the consequences of different possibilities regarding structural feasibility and aesthetic qualities. The possibility to investigate the influence of different design parameters, enables the user to make better informed decisions, throughout the most critical part of the design process. The tool thus gives more freedom in the artistic design.

Looking back at the original research objective and question, it is noteworthy that the intended algorithmic design method covers several aspects: glass structures and more specific structural glass roofing, the gap between architects and structural engineers and the focus of the initial design phase. This led to a broad spectrum of research and development and less to an in-depth product. The purpose remained unchanged during the process, but the formulation and thus the focus of both the research question, aim and objective fluctuated throughout the project. The focus shifted from the potential of a design tool for double curved glass roof structures to the possibility to apply this on glass roof structures in general, specific in the initial design phase. This change of focus was induced during the literature research, after realizing the power that such a parametric design tool could generate by combining the best of two ambivalent and proud disciplines: architecture and structural engineering, as described in Section 1.3.

Bridging the gap between architects and engineers

Section 3.2 explains that the initial design phase has the potential to bridge the gap between architects and engineers. In this part of the process the design decisions have the largest influence on the entire process and project. In order to estimate if the prototype has the capacity to influence this gap, it should be considered if the prototype is suitable for the initial design phase. The prototype is composed so that it provides an alternative design method. Section 5.3 shows that the prototype enables the user to assign any arbitrary outer perimeter. Furthermore, if the user embraces the new parametric design method, it is possible to generate multiple designs in a matter of seconds and explore these options, as seen in Section 6.2.

The prototype provides the means to generate a structurally feasible design, with automatic assignment of appropriate cross sections, without a structural background requisite. The tool also assists in generating efficient structures with appealing appearances; for example by making use of the hanging model base geometry, qualified to generate a wide variety of design.

However, the third case study, as described in Section 6.4, shows the difficulty to explore the boundaries of the design possibilities, especially without a structural background. As soon as a span or cantilever becomes too large, the prototype will not be able to automatically provide a cross section that suffices the maximum allowable stress and deflection (but will give a warning). The same goes for a complex structure, with elements intertwined, large openings, asymmetric form or irregular set-up. When it is necessary to manually optimize the position of internal beams or slight changes of the design, that might lead to significant improvement of the structural feasibility, a structural background or at least a sense of load distribution is essential.

As mentioned in Section 1.3, the structural engineer is often accused of lacking imagination. The prototype facilitates already an wide range of design options, thus empowering the engineer, even without a very great imagination, to easily create eye-catching structures. A more experienced designer can refine this as desired.

Facilitate structural glass design

The literature study in Chapter 2 emphasizes on the fragility of glass and the importance of careful design of the elements and detailing, to

avoid local peak stresses. One of the objectives of this tool is to provide a platform to collect the available knowledge regarding glass structures and to implement this in a design method.

At the moment the material properties of glass are implemented in the prototype, a glass design is linked to an interactive structural analysis and optimisation module, after automatic allocation of loads. The limitations due to the manufacturing process have not yet been fully implemented. The tool takes the different thicknesses of the glass panes into account and the size limitation influencing the grid size. Also a reduce factor in stiffness of laminated beams, by applying an equivalent thickness, is added to the prototype. The module for catenary structures is introduced due to the high compression capacity of the material. The tool provides user awareness for the difference of glass design from more traditional construction methods, with notifications linked to design choices.

Further development

The prototype is merely the base of a much larger ambition, therefore the continuation of the tool is important. In 6.2 the ability to expand the prototype was explored. The modularity facilitates this process, however it is not a matter of copy pasting this into the script. Because of the complicated and interactive features of the tool, a specific extension needs to be interconnected with different parts of the script.

7.3 Reflection on limitations

This section discusses the limitations of the current prototype:

- The tool is developed in Grasshopper, a plug-in in Rhino, with the help of extensions. Therefore calculation and visualization are bundled into one single model. The downside is that the paid service of Karamba is needed for a fully functioning tool. Therefore it is not possible at the moment to use a free trial.
- The prototype offers the possibility to create the base for an efficient glass structure. However, the most important aspect, the detailing, is not implemented in the prototype. Connections have a large impact, both from a structural as an aesthetic point of view. The tool can be used to automatically intersect beams, larger than 6 meters or any other value, depending on the type of

beam and automatically assign connections. An addition that automatically assigns standardized details, corresponding to the mechanical joints assigned by the user, would be useful.

- Due to the adoption of restricted design conditions regarding structural glass, the prototype is solely fit for the very first beginning of the design process and not yet for the general initial design phase. Not all the necessary variables are implemented in the prototype. Expansion of the specific glass variables and additional safety measures are needed in order to be sufficient for structural glass design, as described in the recommendation.
- It should be noted that the outcome of such a tool always needs a second opinion from a specialist. It should function as a design assistant to avoid unnecessary repetition of work, but can never fully replace the function of an engineer. Sanity checks and comparison between the input of the tool and the physical project related situation is always needed. One cannot trust the output of any structural design tool blindly. However, this tool can be used to bring the many possibilities of glass design to the attention.
- The verification of the structural analysis module is merely executed on the basis of simple 'rule of thumb' formulas. For more complex situations a comprehensive check with a FEM program should be performed.
- The optimisation algorithm of the prototype is limited to size optimisation. Shape and topology optimization could be considered as well as multi-objective optimization.
- Merely components provided by the used packages were implemented in the script; this can lead to devious algorithms. Customization of the components will lead to more efficient scripts and quicker results, when the algorithm gets more complex.
- Currently, the script is not protected from adjustments by the user. To avoid accidental alterations of the script, certain components should be protected from changes. At the moment it is possible to return to the standardized values for the parameters without losing the manual input, however this function could be extended.
- Currently, the Eurocode is not fully implemented in the algorithm.
- Addition and development of the algorithm demands knowledge of both Grasshopper and the composition of the script of the tool.

- Software packages suitable for the initial design phase are not widely available. This prototype anticipates on this and seems to be a potential solution for several design problems. As shown in Chapter 6, the tool provides a wide range of design options, but it cannot respond spontaneously to the fluctuating demands of the user and therefore the tool will always be one step behind.

Chapter 8

Conclusions

The motivation to promote the use of glass as a promising structural element, along with the ambition to combine the best of both worlds of architecture and engineering and to close the gap between them, led to the development of an algorithmic design methodology for structural glass covers in the conceptual design phase.

Bridging the gap between architects and engineers

As discussed in previous chapter, based on the conducted case study described in Chapter 6, it can be concluded that the prototype has potential to be used as a design aid for the conceptual design phase. However it needs further expansion to facilitate a broader functionality with other form finding techniques.

The tool provides quick analysis and an optimization of design options. However, the case study also revealed that with an increase of complexity of the design, the urge for structural background increases as well. Therefore, it may be concluded that the prototype is partly suited for designers and architects without structural background and makes a limited contribution to bridging the aforementioned gap.

Because of the ease of generating diverse and appealing design variants, as tested in Chapter 6, it may be concluded that the engineer is provided with a design aid to put himself in the architect's position. Hence the tool is contributing to bridging the gap.

Facilitate structural glass design

The research conducted in Chapter 6 shows that the prototype has potential as a design aid for the use of structural glass, with the implementation of thicknesses, material properties, restrictions due to manufacturing and efficient configurations. The tool can contribute to the ease of designing with structural glass, but needs further expansion of the available knowledge and research regarding structural glass.

From the case study the conclusion is drawn, that the prototype does not provide sufficient guidance regarding safety measures, needed for designing with structural glass.

Further development

From the quick expansion of the prototype during the validation, it can be concluded that the tool can also be used for other materials and configurations. However, as this was done by the developer of the prototype, it is only safe to say that it is possible to expand the script. Further research should confirm, that this can also be done with ease by outsiders. As discussed in the previous chapter, it can be concluded that although the complex nature of the script might hinder further exploration, the modularity contributes greatly to the ease of expansion.

Concluding: The prototype has fulfilled the research- and thesis- objectives, with some limitations. For further development more investigation and tests are needed; this will be elaborated upon in the Recommendations.

Chapter 9

Recommendations

This chapter provides recommendations for continuation of the research and further development, needed to make the tool ready for use. The chapter is divided into two sections, providing recommendation regarding usability, described in Section 9.1 and functionality, shown in Section 9.2.

9.1 Usability

- The main focus, from an architectural point of view, is minimal interference in the creative process. We attempt to accomplish this with instant visual feedback of design decisions and consequences. Currently, the representation from an architects point of view is limited to the experience gained as a bachelor of architecture. To increase the benefits and use of this integral tool, further research should be conducted to map the demands and desires of designers and subsequently to conduct user tests for the tool.
- The user would benefit from better guidance in the parametric design process. In this way the full potential can be utilized. User tests should be executed to establish where the tool needs better explanation.
- It is recommended to complement more standardized values to facilitate the design process; this is necessary for the creative design process and also to implement the calculation and analytic process.
- The manual input should be kept to a minimum, when more modules are added to the tool in order to give maximum freedom to

the creative process.

- Research has been done for interoperability between different software (5.3) and the focus on usability for Building Information Modelling (BIM) to a certain extent. We can use this knowledge to enhance the tool, so that it will become part of the commonly used software.
- Further studies and testing of the user interface, visualization and database management is recommended.
- Restructuring and cleaning of the Grasshopper script is necessary.
- Assemblage of component combinations into single components is recommended.
- It is important to introduce a reset function, to set the variables back to the standard value.
- Research for better understanding of underlying algorithms and the development of alternative components is recommended. The outcome of the analysis is verified and checked, but still consists of elements that work like a black box. For example, the component that uses the first order theory for small deflections already considers buckling, but the exact underlying algorithm remains an enigma.
- More extensive verification and validation of the tool is desirable to enhance the quality of the tool.
- Further research should be conducted to facilitate the ease of expansion.

9.2 Functionality

- To amplify the capacity for bridging the differences between architects and engineers, the tool needs further expansion. This could be done with the help of other formfinding techniques.
- To be multi deployable, it is recommended to expand the functionality to entire structures for all types of geometries with addition of columns, walls and floors. Extension of several modules will be inevitable.

- In order to be useful throughout the entire structural design process, it is important that the calculation and analytic section is extended as well. Buckling, lateral buckling, vibrations, analysis for plate structures and stability should complement the tool.
- As mentioned before, standardization is the key. Load combinations, loading, material factors and other elements of the Euro-code (& national appendix) should be implemented to limit the manual input.
- For the entire design process it is necessary that there is more attention to the detailing of the structure. The effect of certain choices, moment resisting or hinged connections should be explained and examples be provided.
- Other modules that should be investigated are: other construction materials, costs, climate control and building physics, wind simulation, optimization for a given set of requirements and fire engineering.
- The modular set up contributes to the flexibility and continuity of the tool; this creates a lot of development potential, which should be discovered and utilized. This gives a perspective for use in the education process. Thus students can make additions to the tool as part of their study.

Recommendation regarding structural glass

- With glass design, aspects concerning safety, detailing, production, transport and costs need extra attention. This should be integrated in the outcome report.
- It is recommended to complement specific aspects of the glass design like connections, details and consequences.
- Design restrictions arising from the manufacture, transport and construction process should be further implemented. An example is automatic division of glass beams into segments of 6 meters and allocation of connections in between.
- Connect the design decisions to accompanying costs. to provide insight in the price and possibly contribute to a more economical design. Add also notifications, when expensive methods are addressed; this could amplify the cost reduction; for example when using double curved glass panes, panes with a large curvature, large components or expensive surface treatments.

- The tool has the potential to be coupled with an extensive database, functioning as an online encyclopedia.

Bibliography

- Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C. (2014), *Shell structures for architecture: form finding and optimization*, Routledge. 58
- Balkow, D. & Schittich, C. (1999), *Glass construction manual*, Birkhauser. 20
- Barou, L. (2016), 'Transparent restoration', *Master's Thesis, TU Delft* . 20
- Beck, K. (2003), *Test-driven development: by example*, Addison-Wesley Professional. 46
- Blanchard, B. S., Fabrycky, W. J. & Fabrycky, W. J. (1990), *Systems engineering and analysis*, Vol. 4, Prentice Hall Englewood Cliffs, NJ. 27
- Burry, M. (1996), 'Parametric design and the sagrada familia', *Architectural research quarterly* 1(04), 70–81. 32
- Copeland, L. (2004), *A practitioner's guide to software test design*, Artech House. 80
- Cross, N. & Roozenburg, N. (1992), 'Modelling the design process in engineering and in architecture', *Journal of Engineering Design* 3(4), 325–337. 14
- Davis, D., Burry, J. & Burry, M. (2011), Untangling parametric schemata: enhancing collaboration through modular programming, in 'Proceedings of the 14th international conference on Computer Aided Architectural Design, University of Liege, Liege'. 32, 52
- Frederick, M. & Kuprenas, J. (2013), *101 Things I Learned® in Engineering School*, Grand Central Publishing. 28
- Haldimann, M., Luible, A. & Overend, M. (2008), *Structural use of glass*, Vol. 10, Iabse. 20, 21, 22, 23, 24

- IEEE (1998), 'Ieee recommended practice for software requirements specifications', *IEEE Std 830-1998* pp. 1–40. 28, 79
- Larsen, O. P. & Tyas, A. (2003), *Conceptual structural design: bridging the gap between architects and engineers*, Thomas Telford. 28
- Mall, R. (2009), *Fundamentals of software engineering*, PHI Learning Pvt. Ltd. 32
- Schipper, R. (2011), large transparent glass surfaces. TU Delft. 21, 23, 24
- Sharp, J. A. (1992), *Data flow computing: theory and practice*, Intellect Books. 32
- Tedeschi, A. (2014), *AAD Algorithms-aided design: Parametric strategies using Grasshopper*, Edizioni Le Pensur. 58, 59
- Veer, F. (2007), 'The strength of glass, a nontransparent value', *HERON-ENGLISH EDITION*- 52(1/2), 87. 20
- Weller, B., Unnewehr, S., Tasche, S. & Härth, K. (2009), *Glass in building: principles, applications, examples*, Walter de Gruyter. 19
- Woodbury, R. (2010), *Elements of parametric design*, Taylor and Francis. 32
- Wurm, J. (2007), *Glass structures: design and construction of self-supporting skins*, Walter de Gruyter. 24
- Xie, Y., Felicetti, P., Tang, J. & Burry, M. (2005), 'Form finding for complex structures using evolutionary structural optimization method', *Design Studies* 26(1), 55–72. 48
- Zaal, T. (2014), *Integrated design and engineering: as a business improvement process*, Maj Engineering Publishing. 28

Appendix A

Requirement tests

Table A.1: Requirement tasks for phase 1.

Phase 1 Outline	
Module	Tests
Boundary conditions	Provide user with the possibility to adjust length and width of rectangle. Show the outline, always orient the shortest span in y-direction. Shows dimensions connected to the outline in Rhino interface. Show shortest span in Rhino interface.
Base geometry	Create beams with standardized center to center distance (h.o.h.) Provide freedom of controlling either the number of beams or the distance between. Show center to center distance in Rhino interface. Convert elements to beams detectable by Karamba 1.2.2.
Structural properties	Assign basic material properties of glass to the beams. Assign hinged support at end of all beams. Assign a random cross section to the beam elements.
Load generator	Assign a dead load and one mesh load (corresponding to the perimeter) projected on the structure.
Optimization	-
Structural analysis	Perform structural analysis of the input generated so far, limited to UC of stresses and deflection. Provide options for conditions. Visualize the stresses and deflection in Rhino interface. Show cross sections, loads, moment distribution and deflection in Rhino interface.
Report	-

Table A.2: Requirement tasks for phase 2.

Phase 2 Optimization	
Module	Tests
Boundary conditions	-
Base geometry	Allow user to add secondary beams and define h.o.h.
Structural properties	-
Load generator	-
Optimization	Optimize element based on the stresses and deflection of these elements. Apply height as variable for this optimization. Change the thickness non-linear with the height and increase with steps linked to glass production. Enable manual adjustment of the width distribution. Attach name tags to cross sections. Link the optimization to parameterized input. Provide real-time feedback. Define maximum height.
Structural analysis	-
Report	-

Table A.3: Requirement tasks for phase 3.

Phase 3 Form finding	
Module	Tests
Boundary conditions	-
Base geometry	Add catenary form finding to the design process. Implement a particle spring engine to induce catenary principle. Give possibility to determine maximal height. Make form finding process interactive with the parameterized input. Provide form finding process with instant feedback.
Structural properties	-
Load generator	-
Optimization	Notify when elements do not satisfy the conditions.
Structural analysis	-
Report	-

Table A.4: Requirement tasks for phase 4.

Phase 4 Extension	
Module	Tests
Boundary conditions	Add a 4-point polygon, not limited to 2d plane but extend to 3 rd dimension. Provide a geometry flexible and based on more points.
Base geometry	Add triangulation of the structure. Provide freedom of base surface geometry. Provide projection of beams on any given surface.
Structural properties	Provide user freedom to adjust types and location of supports. Provide user choice of material and possibility for adjustments.
Load generator	-
Optimization	-
Structural analysis	-
Report	-

Table A.5: Requirement tasks for future developments.

Future developments	
Module	Tests
Boundary conditions	-
Base geometry	Provide design options to generate a geometry using only flat panels.
Structural properties	-
Load generator	Load combinations concerning the Eurocode. Enable asymmetric loading.
Optimization	Make increase of thickness dependable on available glass thickness combinations. Optimize costs, material use and other design variables.
Structural analysis	Implement a stability check and additional analysis methods. Provide the envelope of the structural outcome.
Visualization	-
Report	-

Appendix B

Structural verification of the prototype

To check the reliability of Karamba, three common cases are considered that can be compared with several 'forget me not' formula's, simple hand calculations. In addition to this basic check, a comparison with the results of any structural design software could be compared to the. A comparison of the results shows a compelling resemblance with the findings of the algorithmic tool

Test case 1 – Cantilevered beam:



General info

$$l = 10 \text{ [m]}$$

$$F = 10 \text{ [kN]}$$

$$b = 0,02 \text{ [m]}$$

$$h = 0,5 \text{ [m]}$$

$$E = 70000 \left[\frac{\text{N}}{\text{mm}^2} \right]$$

$$f_y = 40 \left[\frac{\text{N}}{\text{mm}^2} \right]$$

$$W_{\text{rectangle}} = \frac{1}{6}bh^2 = \frac{1}{6}0,02 \cdot 0,5^2 = 8,33 \text{ E} - 4 [\text{m}^3]$$

$$I_{\text{rectangle}} = \frac{1}{12}bh^3 = \frac{1}{12}0,02 \cdot 0,5^3 = 2,08 \text{ E} - 4 [\text{m}^4]$$

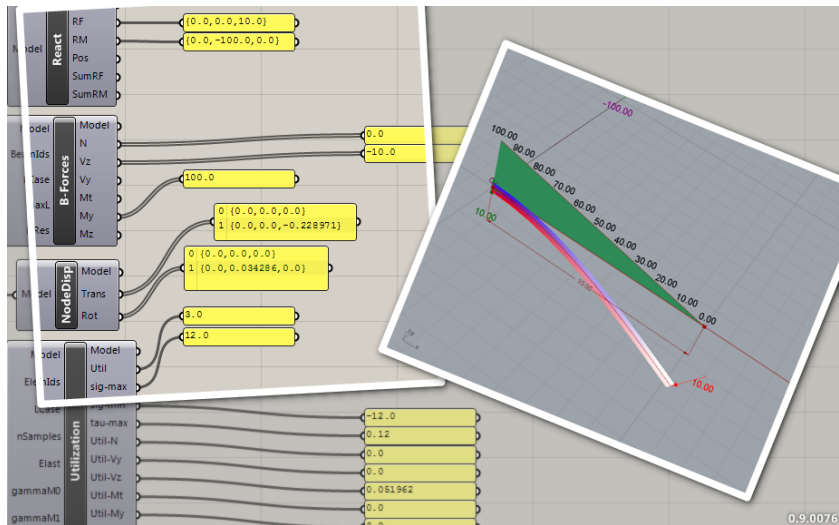
$$M_{\text{max}} = Fl = 10 \cdot 10 = 100 [\text{kNm}]$$

$$\sigma_{\text{max}} = \frac{M_{\text{max}}}{W} = \frac{100}{8,33\text{E} - 4} \cdot 10^{-3} = 120 \left[\frac{\text{N}}{\text{mm}^2} \right]$$

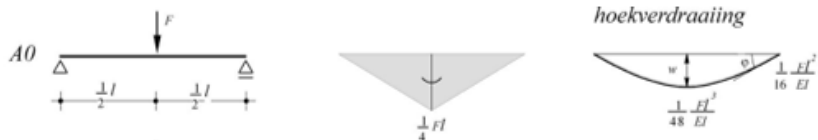
$$u = \frac{1}{3} \frac{Fl^3}{EI} = \frac{1}{3} \frac{10 \cdot 10^3}{2,7\text{E}7 \cdot 2,08\text{E} - 4} = 0,2289 [\text{m}]$$

$$\varphi = \frac{1}{2} \frac{Fl^2}{EI} = \frac{1}{2} \frac{10 \cdot 10^2}{2,7\text{E}7 \cdot 2,08\text{E} - 4} = 0,0343 [\text{rad}]$$

$$UC = \frac{120}{40} = 3 > 1 \text{ would not suffice}$$



Test case 2 – Simple supported beam:



General info

$$l = 15 \text{ [m]}$$

$$F = 15 \text{ [kN]}$$

$$b = 0,02 \text{ [m]}$$

$$h = 0,5 \text{ [m]}$$

$$E = 70000 \left[\frac{\text{N}}{\text{mm}^2} \right]$$

$$f_y = 40 \left[\frac{\text{N}}{\text{mm}^2} \right]$$

$$W_{\text{rectangle}} = \frac{1}{6} b h^2 = \frac{1}{6} 0,02 \cdot 0,5^2 = 8,33 \text{ E} - 4 \text{ [m}^3\text{]}$$

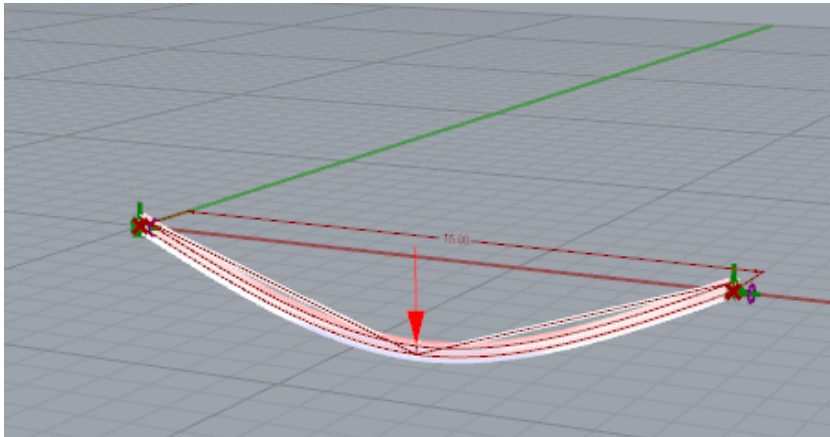
$$I_{\text{rectangle}} = \frac{1}{12} b h^3 = \frac{1}{12} 0,02 \cdot 0,5^3 = 2,08 \text{ E} - 4 \text{ [m}^4\text{]}$$

$$M_{\text{max}} = \frac{1}{4} F l = \frac{1}{4} 15 \cdot 15 = 56,25 \text{ [kNm]}$$

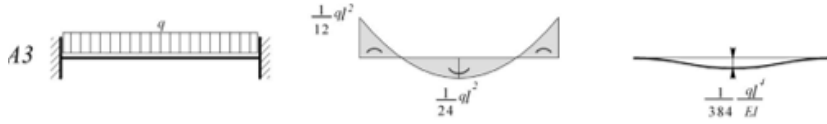
$$\sigma_{\text{max}} = \frac{M_{\text{max}}}{W} = \frac{56,25}{8,33 \text{ E} - 4} \cdot 10^{-3} = 67,5 \left[\frac{\text{N}}{\text{mm}^2} \right]$$

$$u = \frac{1}{48} \frac{F l^3}{E I} = \frac{1}{48} \frac{15 \cdot 15^3}{7 \text{ E} 7 \cdot 2,08 \text{ E} - 4} = 0,07243 \text{ [m]}$$

$$UC = \frac{56,25}{40} = 1,4 > 1 \text{ would not suffice}$$



Test case 3 – Double clamped beam:



General info

$$l = 10 \text{ [m]}$$

$$q = 5 \text{ [kN/m]}$$

$$b = 0,02 \text{ [m]}$$

$$h = 0,5 \text{ [m]}$$

$$E = 70000 \left[\frac{\text{N}}{\text{mm}^2} \right]$$

$$f_y = 40 \left[\frac{\text{N}}{\text{mm}^2} \right]$$

$$W_{\text{rectangle}} = \frac{1}{6} b h^2 = \frac{1}{6} 0,02 \cdot 0,5^2 = 8,33 \text{ E } - 4 \text{ [m}^3\text{]}$$

$$I_{\text{rectangle}} = \frac{1}{12} b h^3 = \frac{1}{12} 0,02 \cdot 0,5^3 = 2,08 \text{ E } - 4 \text{ [m}^4\text{]}$$

$$M_{\text{support}} = \frac{1}{12} q \cdot l^2 = \frac{1}{12} 5 \cdot 10^2 = 41,667 \text{ [kNm]}$$

$$M_{\text{field}} = \frac{1}{24} q \cdot l^2 = \frac{1}{24} 5 \cdot 10^2 = 20,833 \text{ [kNm]}$$

$$\sigma_{\text{max}} = \frac{M_{\text{max}}}{W} = \frac{41,667}{8,33 \text{ E } - 4} \cdot 10^{-3} = 50 \left[\frac{\text{N}}{\text{mm}^2} \right]$$

$$u = \frac{1}{384} \frac{q l^4}{EI} = \frac{1}{384 \cdot 7 \text{ E } 7 \cdot 2,08 \text{ E } - 4} 5 \cdot 10^4 = 0,0089 \text{ [m]}$$

$$UC = \frac{50}{40} = 1,25 > 1 \text{ would not suffice}$$

Appendix C

Case Study 1

This appendix shows an overview of different design possibilities. The variables are the height of the structure, location of the supports, distance between the beams and overall curvature of the structure. Below we show the top and side views and a perspective view.

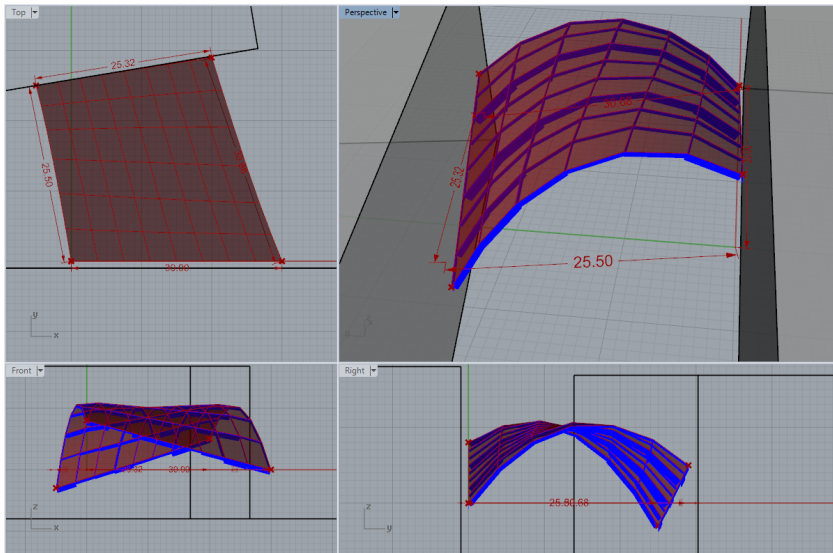


Figure C.1: Design Alternative

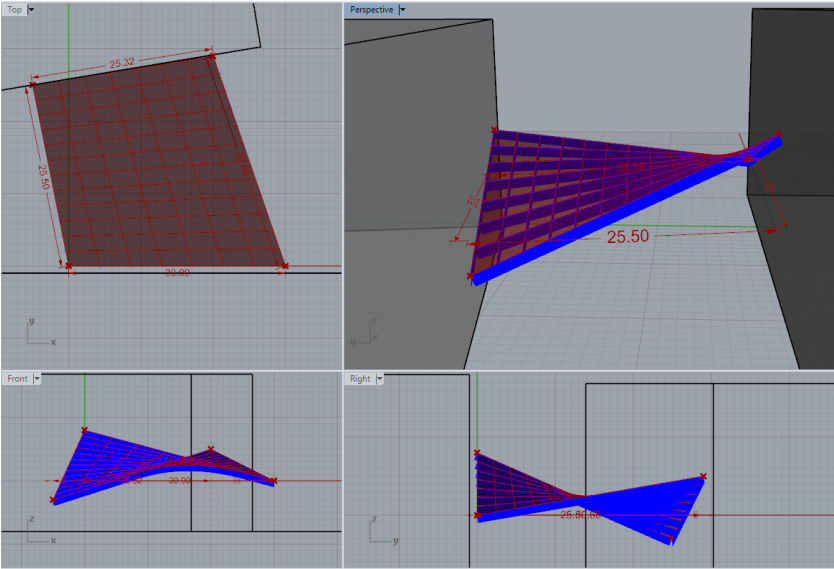


Figure C.2: Design Alternative

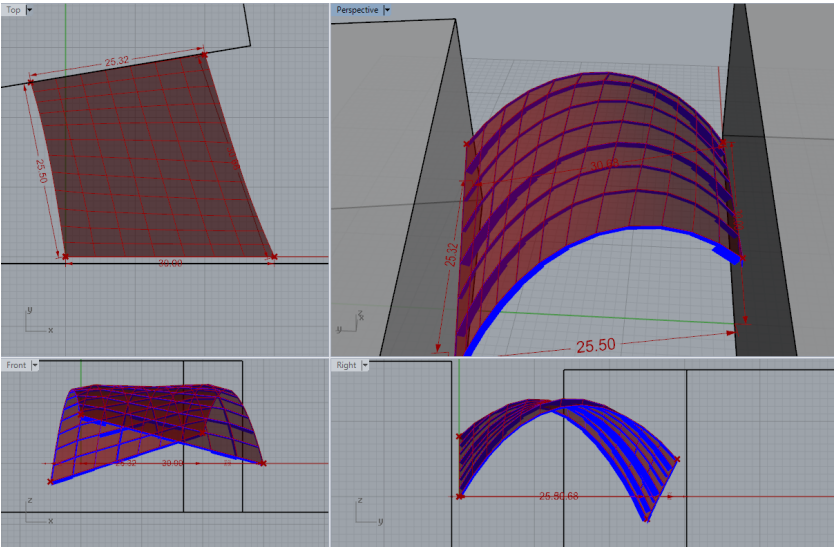


Figure C.3: Design Alternative

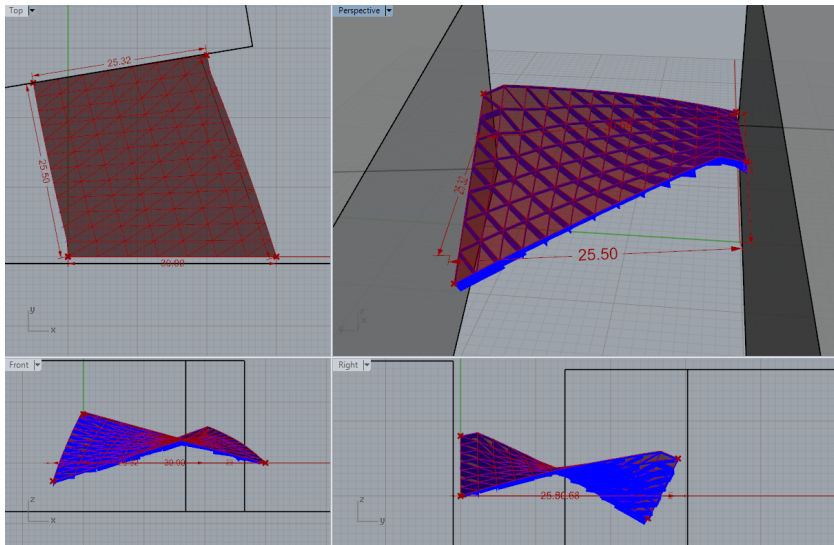


Figure C.4: Design Alternative

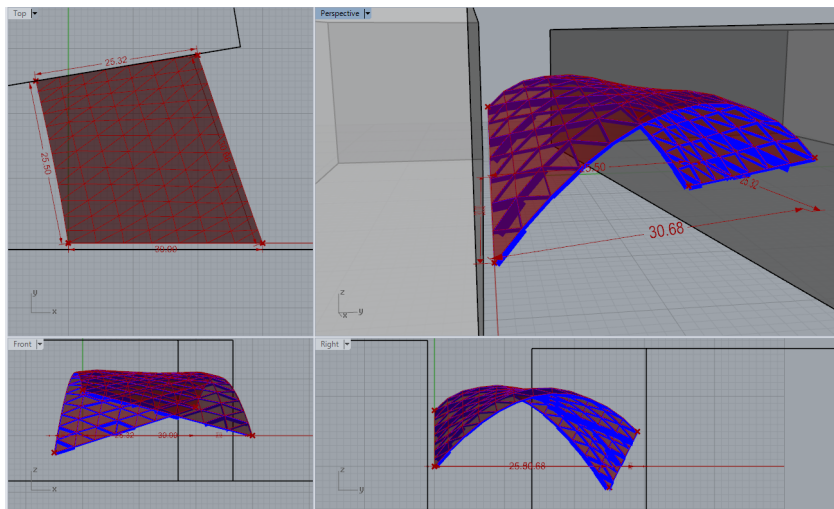


Figure C.5: Design Alternative

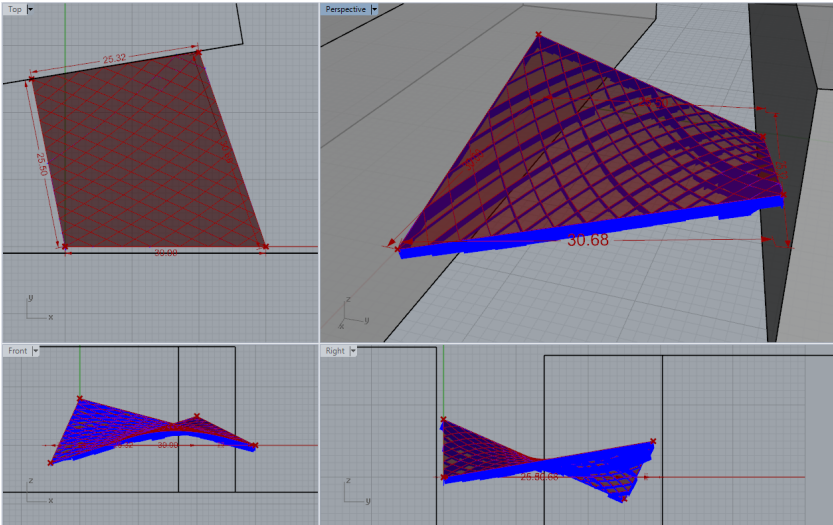


Figure C.6: Design Alternative

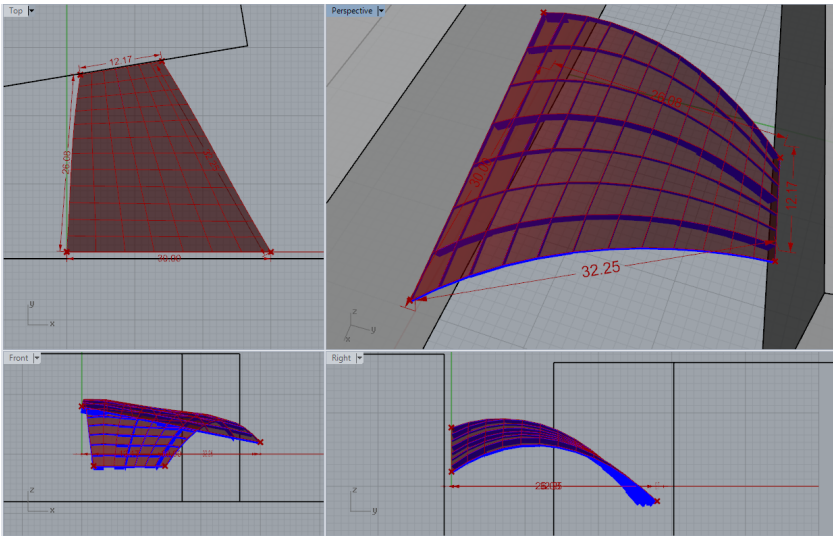


Figure C.7: Design Alternative

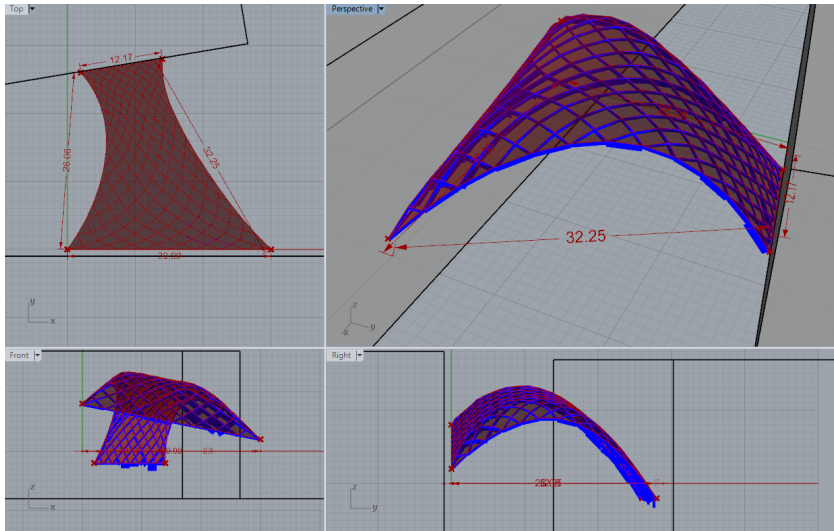


Figure C.8: Design Alternative

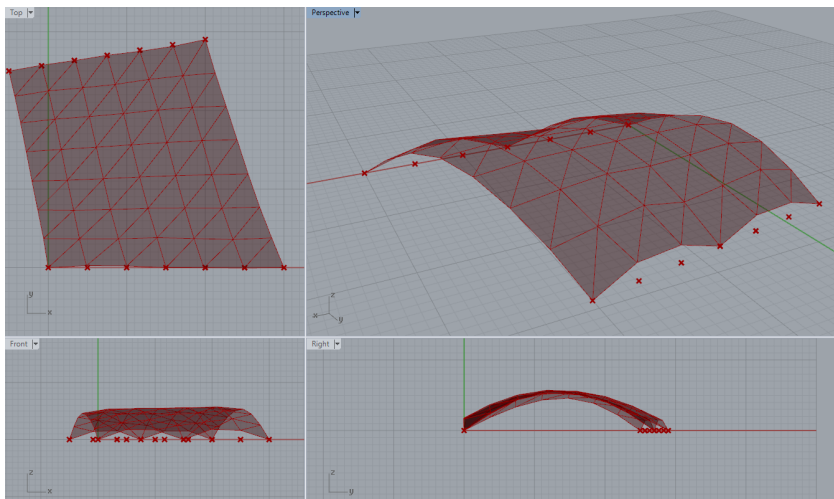


Figure C.9: Design Alternative

Appendix D

Case Study2

The second case study consists of a central laboratory for building technology at the faculty of Architecture. The structure will essentially be a full glass structure distinctive for its transparent characteristics. For this assignment, it is important to minimise the use of components other than glass. This is obviously to enlarge the main function of the material; transparency. However, for certain functions the characteristics of glass are not optimal. This will become clear, for example, in the section of the connections. We will minimize the use of other materials, considering the added functional value against the loss of transparency.

The boundary conditions for the design are described with the following requirements:

- The building is built of glass and doesn't obstruct the view of the building.
- Other materials are minimized and only used where functionally necessary.
- The enclosed area should be as large as possible
- A straight corridor should be created from the main entrance to the road.
- A minimal offset of 3 meters between the facade and glass structure is mandatory
- The glass building should accommodate 550m² of laboratory space
- A loading place for the truck should be available at the front of the building.



- All supplies should be able to get into the structure and be moved through the corridor.

Three designs will be created and chosen for further development.

Conceptual design phase - Orientation

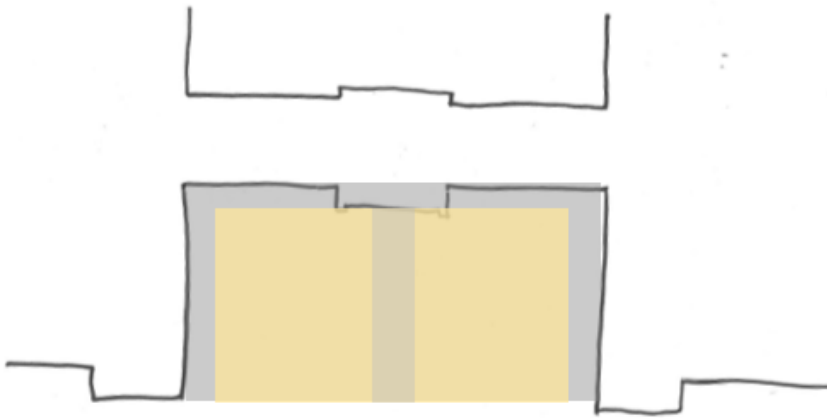
The use of glass as a structural element slowly finds its way in today's construction industry. With its high compression strength glass is very applicable. However, in order to work with glass it is important to get familiar with the material, due to its still unknown and to a certain extent unpredictable nature.

For this assignment, a system is designed with use of hanging models to enlarge the compression in the structure and allow bigger spans. This can be applied in several assemblies, we will discover several possibilities during this research. Important starting points for this design were:

- Maximal transparency
- Structure and architecture intertwined
- Efficient structural system, keeping properties of glass in mind

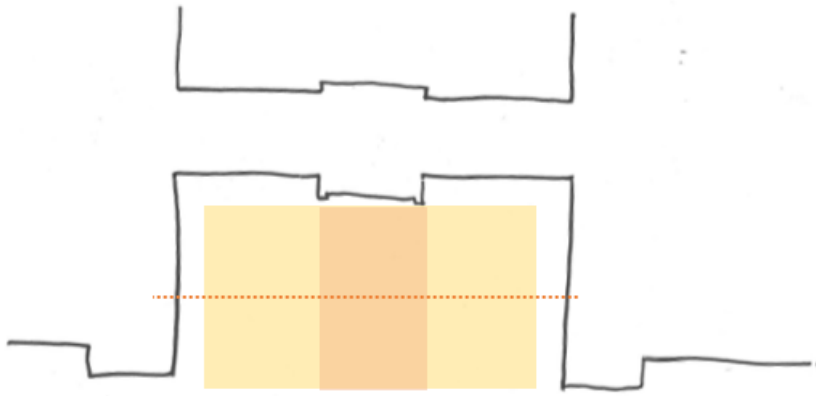
- Detailing, keeping properties of glass in mind
- Repetition of elements

The requirements for the boundaries have a considerable influence on the design concept. A corridor in the centre of the plan, to the entrance, prevents columns to be placed in the middle. The offset of three meter from the existing facade, in combination with maximizing of the surface, dictates the outer shape. However, we have a lot of freedom for the shape of the roof surface. In this report, we will focus on this part of the design. The figure below shows the boundary conditions.



The restrictions led to a subdivision of three parts, the center and the side parts. The internal columns will be placed on the borderlines. This will result in an unobstructed view when entering or leaving the main building. For the feasibility, the design will be mirrored along the x axis.

To minimize the tensile stresses, we revert to the hanging model, that is subjected to gravity. Karamba provides the possibility to simulate the behaviour of hanging models, with help of the “Analyze large deformation” component. The geometric non-linearity is handled by an incremental approach: The external loads get applied in steps. After each step, the model geometry updates to the deflected state. The more and the smaller the steps, the better the approximation of geometric non-linearity. The purely incremental method however incurs an unavoidable drift from the exact solution. For form finding this error should be negligible in most cases.



The user can influence the form finding process by adjusting the mesh size and the location of the supports. Another function that is implemented in the prototype is triangulation of the mesh, which offers higher stability and a change of looks. It is possible to use all kinds of grids and patterns, every pattern is connectable to the current module.

Design variants

There are different possibilities for the hanging model. The first variant shows a difference in height between the internal and external supports. The second shows equal heights of the internal and corner supports. The third option shows all supports at one level. The first and third option ensure the possibility to use a standardized solution for the connection with the facade elements. From a structural point of view, the first option is least efficient and the third most efficient. We will elaborate on this third option, because this means obstruction of the view on the building is minimized. In the next chapters, we will discuss the principles for connections and assemblage. In the chapter of structural analysis, we will discuss the final dimensions of the elements. Optimization might also lead to a change of the grid size.

Overall dimensions

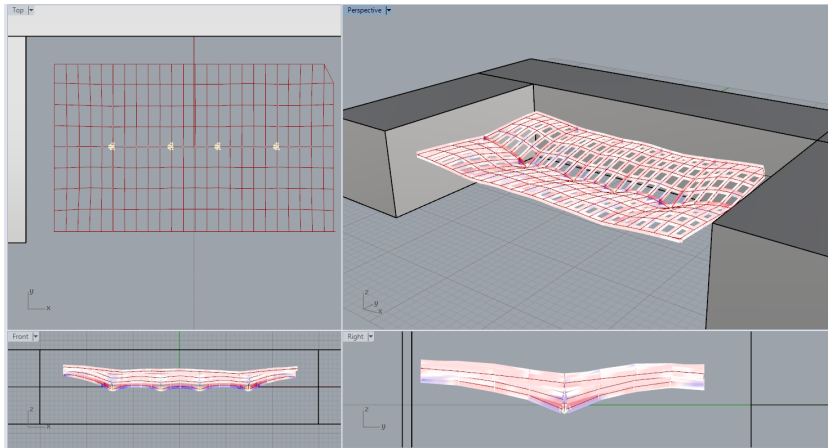


Figure D.1: Variant 1

Sizing and manufacturing of elements

Due to the size limitations of the lamination furnace we limit the maximum component size to 5.4 meters. Bigger sizes are possible, but we want to reduce the costs. The main beams are split in three parts. The secondary beams are placed as a whole in between.

The central main beams spanning in the x direction have a height differing from 340mm to 610mm. We use the governing height and apply it to all sections of this middle beam. The beam is built from 4 layers of 19mm sheets and a sacrificial layer at each side of 6 mm. 6-19-19-19-19-6, in total 88mm and 76 mm that is accounted for in the calculation.

The beams that are placed perpendicular to the main beam have a governing height of 490mm. The beam is built from 4 layers of 16mm sheets and a sacrificial layer at each side of 6 mm. 6-16-16-16-16-6, in total 76mm and 64 mm that's accounted for in the calculation. For the secondary beams, we use a height of 100mm, built from 3 panels of 12 mm. And as mentioned before, we use 12-12-12-12-6 for the roof panes.

Construction

An important aspect of the feasibility of the glass structure is the construction process. We will focus on the structure itself and assume

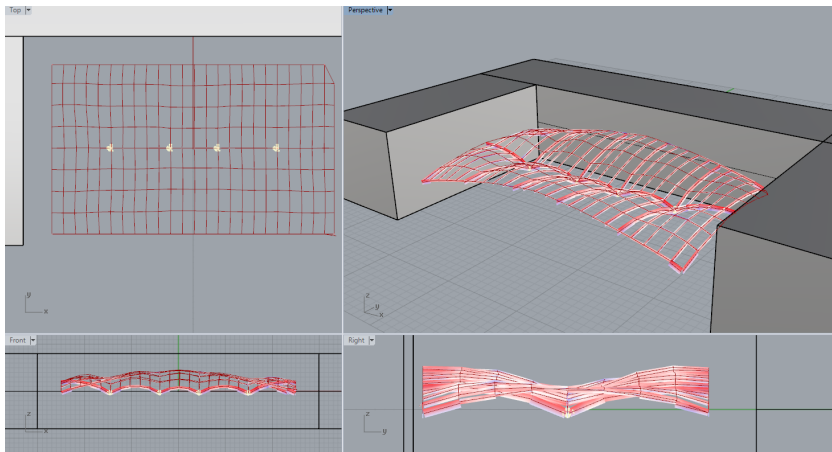


Figure D.2: Variant 2

the foundation is designed properly to transfer the loads and house the pipes for the water drainage. The image on p.149 summarizes the assembly sequence. First the glass brick columns in the corner of the plot will be placed. The facade elements with the glass fins are placed between the outer columns. The mid main beam is placed on top of the columns before the perpendicular beams are placed. The straight shape of the brick columns and glass fins are efficiently shaped for the rectangular container and are therefore transported in one piece and hoisted in place. The arched main beams are not and are transported in three pieces and assembled on site before hoisted. the edge beams are placed and subsequently the roof panes are mounted to the beams and provide stability of the whole. This process is repeated until the structure is completed. It is important that the different connections are accessible during construction, but also for maintenance and repair. The middle part near the street is assembled last, for every part to be easily accessible during construction.

Connections

The connections are a critical aspect for glass design, it is important to prevent high local stresses in the material. Naturally, it is necessary to connect the different elements of the design. One of the most important aspects of these connections is to reduce the visual impact of the non-glass elements, while ensuring the desired strength and mechanical

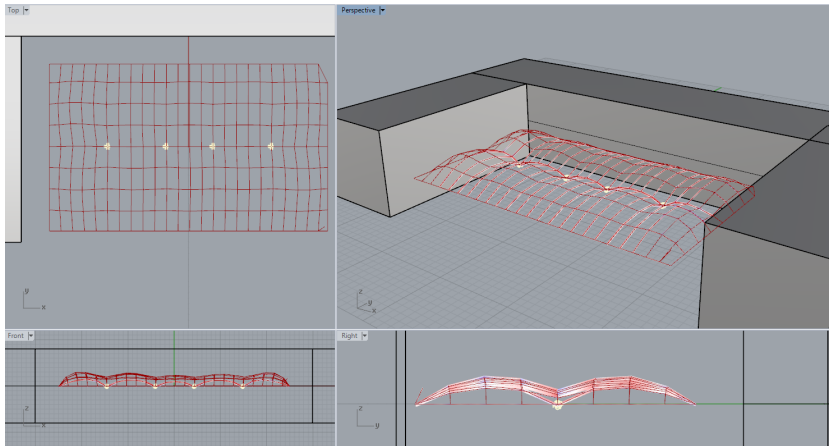


Figure D.3: Variant 3

performance.

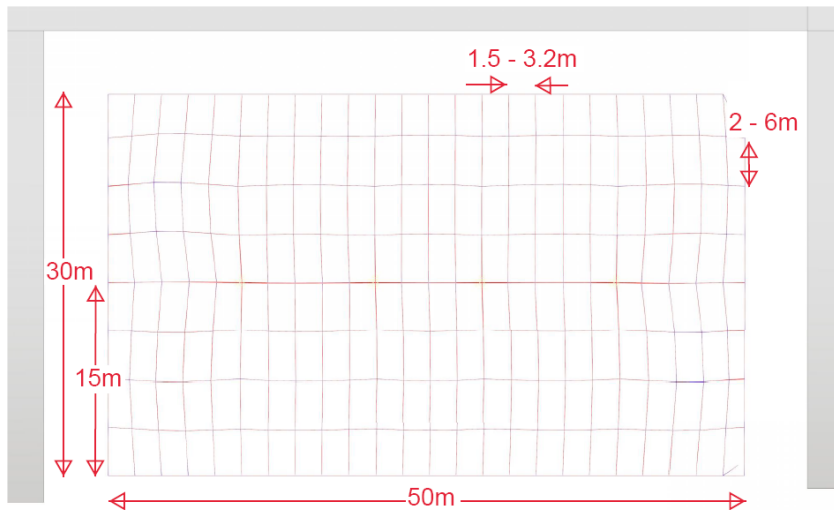
There are different types of connections using holes, adhesives, and metal inserts. Understanding of the material is important. Holes and bolts which are in direct contact should be avoided, to prevent local high stress concentration. The stresses should be introduced over a bigger area. Adhesives are a good alternative, but the right thickness is needed for optimum strength. The surfaces also should match and need to be clean (so no dust during construction).

We will look at the connection of:

- Edge beam to edge beam
- Mesh beam to edge beam
- Column to edge beam
- Beams to facade

Main beams

Due to cost efficiency, the main beams are split in three parts, the connection between the elements needs to take up the occurring bending moment. One of the design restrictions is the degrees of freedom of the connections. In the main direction, the elements have no rotational freedom. This will be visible in the connection. To transfer the loading, the metal insert is placed between the glass panels. The middle panels are set back to create enough glued connection to transfer this loading.



An extra measurement will be the reinforcement of the main beams. In case of complete failure the reinforcement will keep up the entire structure.

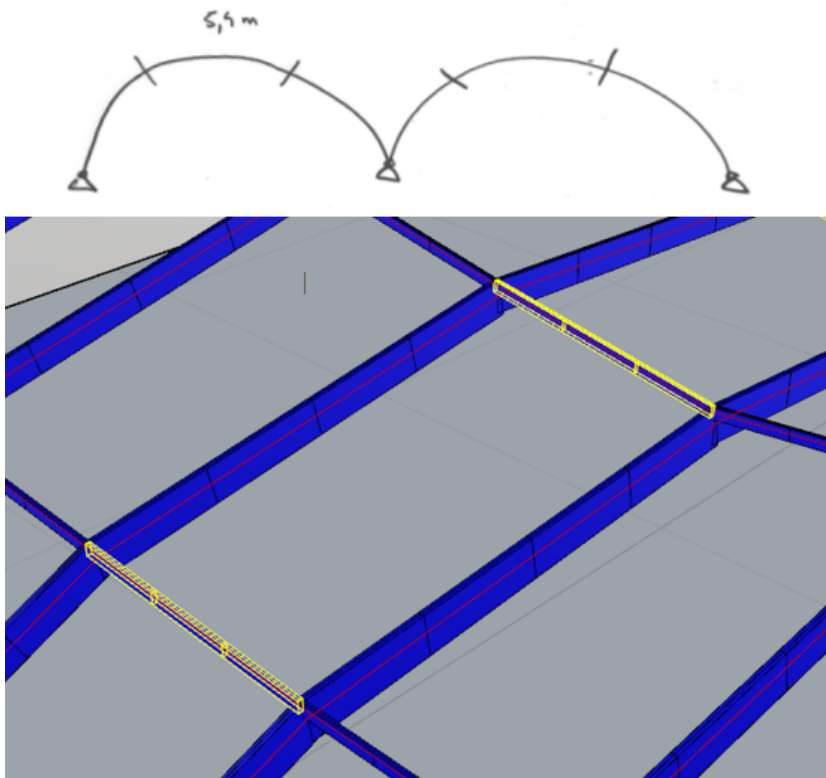
Secondary beams

In the secondary direction, coupling is provided with hinged joints. Stability is provided by the roof planes, connected with a structural sealant. The connection principle for the secondary beams is showed in image below. However, due to the visual impact of the steel elements, we decided to eliminate the secondary beams by limiting the span between the main beams.

Roof structure to columns

For the columns and the connection to the roof we are inspired by the principle used in the thesis of Vania Gatsiou, shown in the figure below.

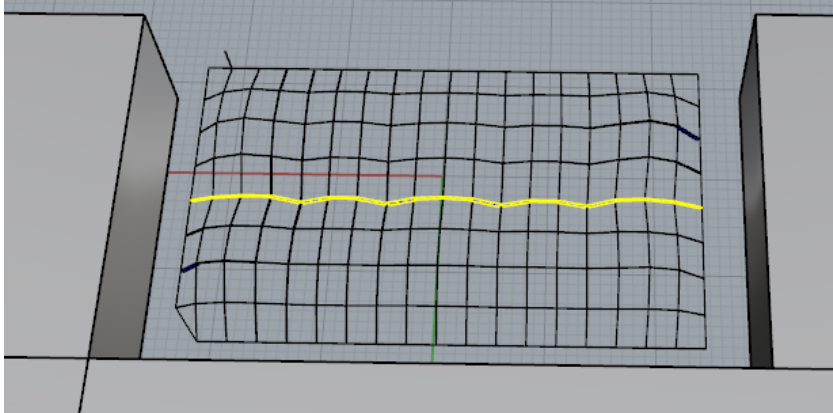
However, we will make some adjustments to deal with an important aspect of the load distribution, water accumulation. The shape of the roof make this an important aspect for the detailing, therefore we would like to implement water drainage to the columns of the structure. The moment distribution in the supports of the column is 105 kNm and the the vertical reaction force is 400 kN. We design the connection to



withstand these loads.

For the opening of the water drainage we use NEN-EN 1991-1-3 art. 7.2. We need at least a free centre of 170x170mm with bricks of 100mm we'll need columns of 400x400mm.

The type of the connection between the glass beams and the glass brick column in the roof uses the glass bricks mass in order to stabilize the beams and avoid the local torsional buckling. The beams are connected with the use of steel elements which are glued at the top of the column. After adding the bolts in each element, customized glass bricks are glued at the four sides in order to cover the whole length of the beams. They can be easily removed in case of maintenance or replacement of the elements. An addition to the design is the opening of the top and different configuration of the steel elements, which allow water to pass and prevent water accumulation.



Roof to facade

The focus of this report is the roof structure, therefore we will only discuss the principle of the facade. This consists of glass fins, with vertical glazing in between. The image below shows a hinged connection between the glass fins and the roof beams. These supports are placed at every intersection and therefore only need to withstand relatively small forces. If needed a secondary fin will be placed in between to prevent the fins from buckling.

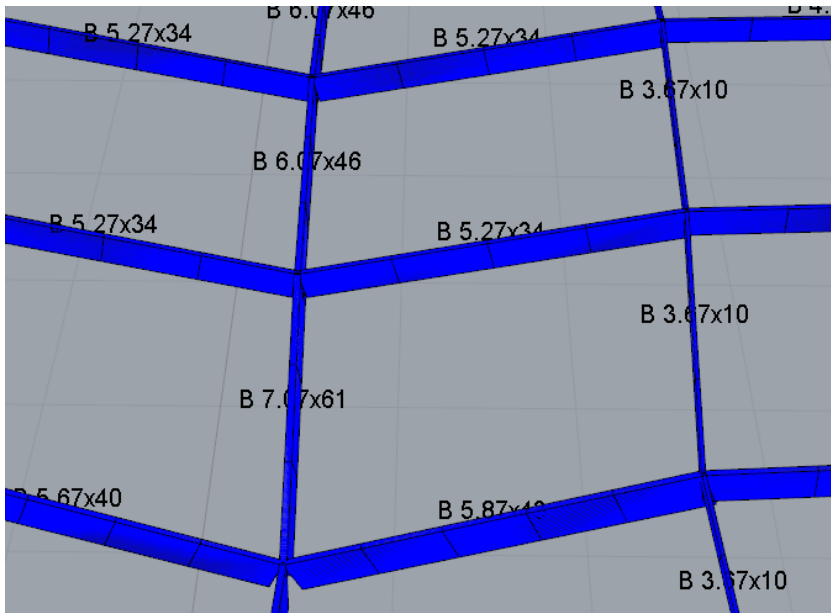
Risk Analysis and Safety

For the overall safety of the structure we want to minimize the risk. The risk is defined as follows:

$$\text{Risk} = \text{consequence} * \text{probability}$$

The higher the probability the lower the risk should be and vice versa.

We determine the consequence of a failure in terms of injuries, ranging from first aid to catastrophic consequences with many deaths. And the probability depends on the possibility of a scenario to happen and the exposure of the structural element. First, we consider different scenarios that might occur, during the structure's lifetime, subsequently we search for methods to reduce the consequences and probability of the scenario, to reduce the risk.



Scenario's:

Vandalism: Delft is city with a lot of students, known for their drinking and becoming reckless. The building is close to the city center. Therefore a slightly increased risk for vandalism

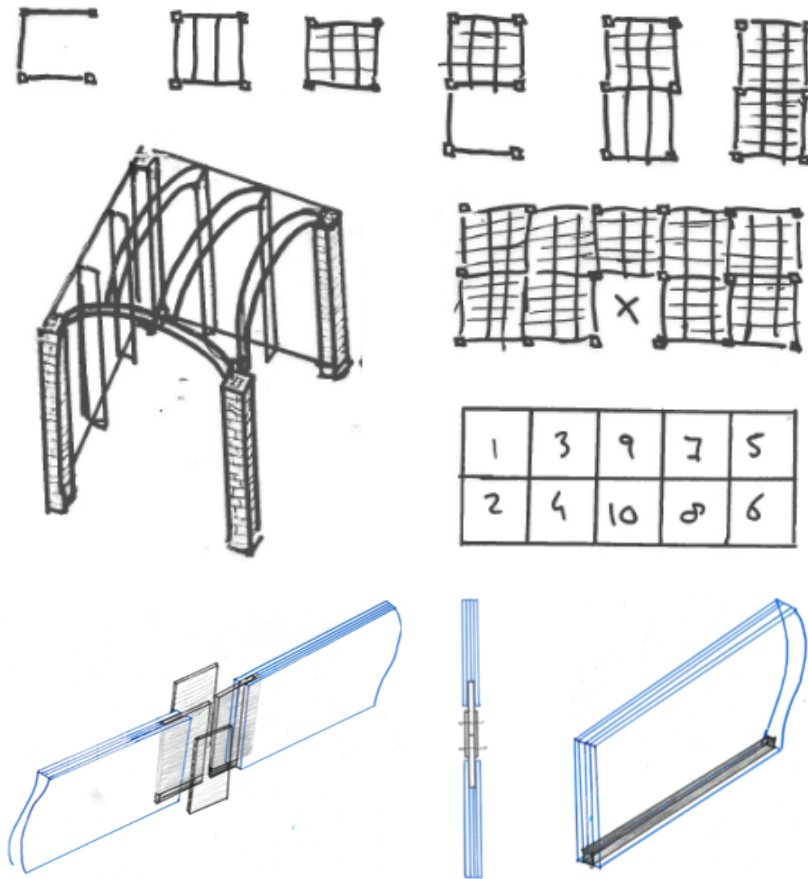
Collision: The structure is exposed to the Julianalaan on one side of the building. Therefore there's a risk of a vehicle crashing into the structure. The threat can also come from inside the structure, for example when a forklift truck accidentally hits one of the columns.

Fire: A fire with an extreme high temperature, reducing the bond of the lamination of the main structural elements and eventually complete failure.

Reducing the consequences:

Sacrificial layer: effective for vandalism. When a rock hits the outside of an element, this layer will take the hit.

Second load path: we increase the redundancy with alternating the load path of the structure. This will be done with reinforcement in the main beams. Even if a complete column fails, and causes failure of the main

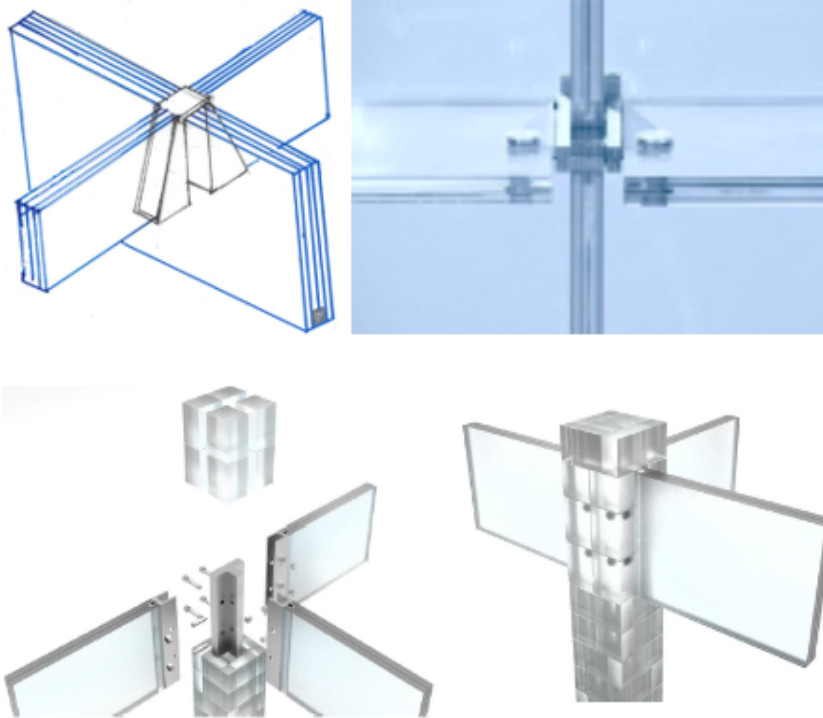


beams, the roof will hang between the next columns.

Thermal pre-stressing: this will not prevent failure when this fire happens, but provides extra time for the visitors to leave before the elements collapse. This will reduce the consequence to material damage.

Structural analysis

For the analysis, we'll apply a load generator. The most important feature of this module, is the interactivity with the parametric structure and the freedom for the user to apply, adjust and remove loads. This is a big advantage compared to the use of GSA. For the load distribution, we distinguish the permanent and variable loads and consider combinations of the two to check the serviceability and ultimate state. the



values that were applied are described below:

Permanent load

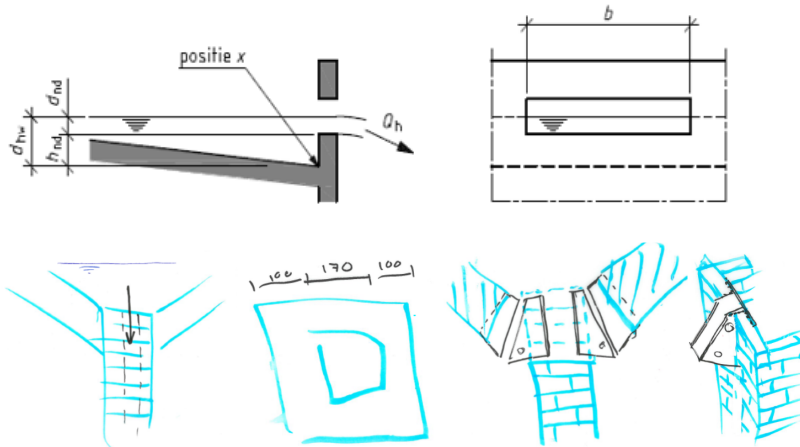
The self-weight of the structure and dead load of the facade elements or cladding subjected by gravity. These are implemented in the script and linked to the elements. Density of Glass 25 kN/m³

The dead load depends on the cross section of the elements and therefore interactive in the script.

Roof panes: For the roof panes, we will use a conservative value. We use a protective layer on the top side of the glass panes, due an increased risk of damage. When checking the panes for maximum allowable stress and deflection ($u_{max} = 0,004 \cdot l$) with help of an algorithm for shells and supports on two sides, we see that we need panes of 40mm + 6mm, deflection is governing in this case.

We cannot assume that the entire cross section works as a solid cross section, therefore we take 85% due the lamination. Instead of 40mm we will use 47mm as minimum. We use 12 12 12 12 6 for the roof panes.

$u_c =$ 5,00E-05 m/s		$b = d_{nd} + \text{minimaal vereiste ophoging EC1:}$ 30 mm									
		toegepast ophoging: 50 mm									
nr	clak	opp [m ²]	dnd [mm]	bnd [mm]	Qh [m ³ /s]	b ben. [mm]	dhw [mm]	aantal [-]	b [mm]	b [mm]	b totaal [mm]
1		247	120	50	0,0124	174	170	1	170	170	170



Variable load

wind load: The wind pressure is estimated 1 kPa, so 1000 N/m.

$F_{wind} = C_s \times C_d \times C_f \times q_p(Z_e)$

Compression wind load: $Q_d = 1,0 \times 1,0 \times 0,7 \times 1 = 0,7 \text{ kN/m} = 700 \text{ N/m}$

Tension wind load: $Q_s = 1,0 \times 1,0 \times 0,25 \times 1 = 0,25 \text{ kN/m} = 250 \text{ N/m}$

We consider the wind load as both horizontal and vertical loading. For further analysis, we recommend taking asymmetric loading into account as well. For now, we decided to focus on the optimization and neglect this effect.

Snow load: Snow loads are determined according NEN-EN 1991-1-3:

$Q_s = 1,2 \times 1,0 \times 1,0 \times 0,7 = 0,8 \text{ kN/m}$

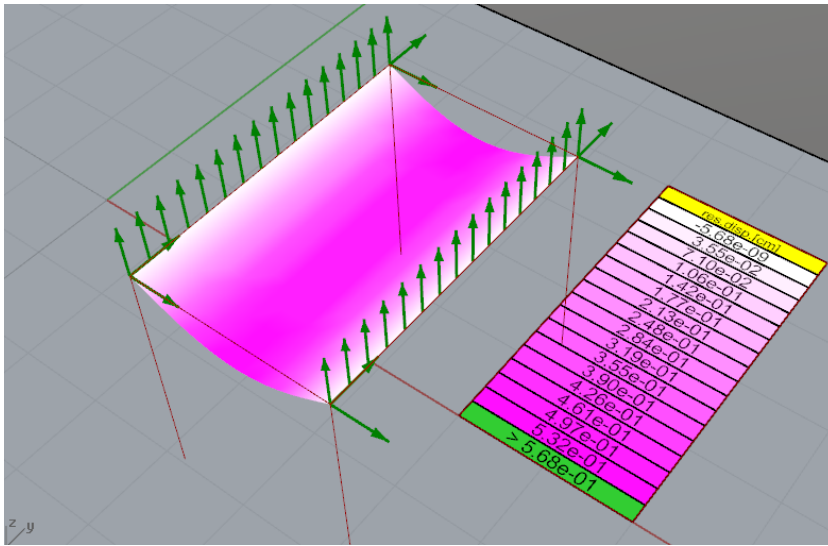
Maintenance: For this conceptual design phase, we'll ignore the point load of 1kN, that's more favourable than the wind load.

Load combinations

We will allow for the following load combinations; the grey combinations have not been taken into account:

Final Structural Analysis

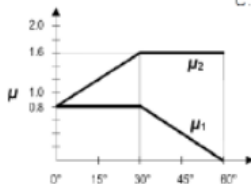
The structural analysis is performed with the analysis module in Karamba and shows, as mentioned before, instant results. All optimization



determining the snow loading

$$s = \mu_i \cdot C_s \cdot C_t \cdot s_k$$

- s = Characteristic value of the snow load on the ground [kN/m²] (0,7 kN/m² in the Netherlands)
- μ_i = Shape coefficient, dependent on roof angle α
- C_s = Exposure coefficient (1,0 for most cases)
- C_t = Thermal coefficient (1,0 for most cases)



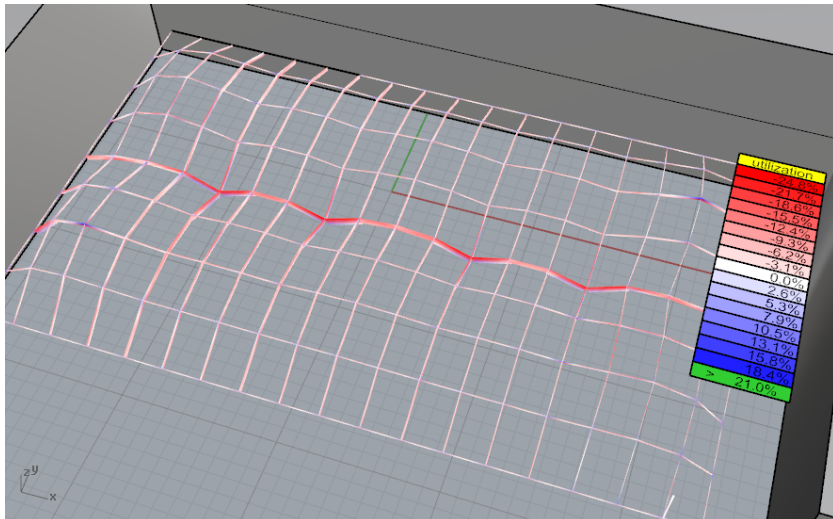
The shape coefficient can be determined with the shown diagram. Note that μ_2 is only applicable where snow accumulation can occur. For complex roof shapes and more information, see NEN-EN 1991-1-3

models are calculated with the load combinations as described before. For the final analysis, we'll discuss the outcome of the optimized structure for the critical combination, ULS LC3 and SLS LC 7. We will briefly discuss the outcome. We will use the roof panes as an extra load of the structure ($0,054 \cdot 25 = 1.35$ kN/m²). We will use this for the dimensioning of the structural elements, keeping in mind the maximum stress should be smaller than 40 N/mm² and the deflection smaller than 0,004 times the span.

For the analysis of the structural glass we considered an equivalent thickness of 85% of the laminated panes, to take into account a reduce of the shear interaction.

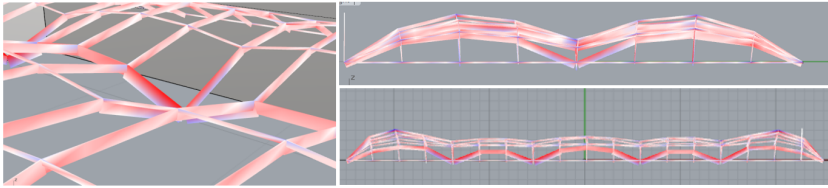
ULS	Load case	safety factor
1	self-weight only (G)	$1,35 \cdot G$
2	self-weight plus snowload (Qsn)	$1,2 \cdot G + 1,5 \cdot Q_{sn}$
3	self-weight plus wind upwards (Qw+)	$1,2 \cdot G + 1,5 \cdot Q_{w+}$
4	self-weight plus wind downwards (Qw-)	$0,9 \cdot G + 1,5 \cdot Q_{w-}$
SLS		
5	self-weight only	$1,00 \cdot G$
6	self-weight plus snow load (Qsn)	$1,0 \cdot G + 1,0 \cdot Q_{sn}$
7	self-weight plus wind upwards (Qw+)	$1,0 \cdot G + 1,0 \cdot Q_{w+}$
8	self-weight plus wind downwards (Qw-)	$1,0 \cdot G + 1,0 \cdot Q_{w-}$

Stresses

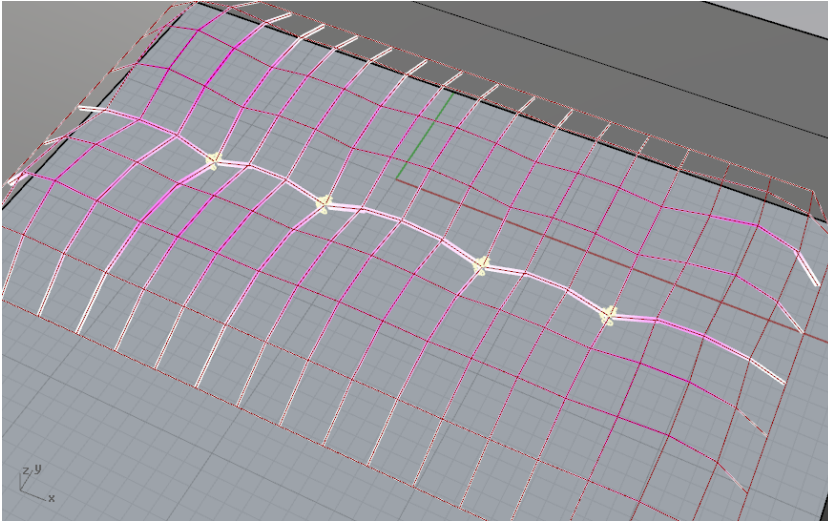


In the figure above, we see that, due to the form finding, the tensile stresses are reduced. The highest stress can be found in the central main beams spanning in the x direction, and the lowest stresses are found in the secondary beams of the grid. During detailing of the connections, It is decided to eliminate the secondary beams. This is possible everywhere except one place near the edge. Therefore, the beams near the edges (on both sides) will remain.

The secondary beams are not necessary, but used to decrease the deflection and stresses of the roof panes and build in an extra safety factor.



Deflection



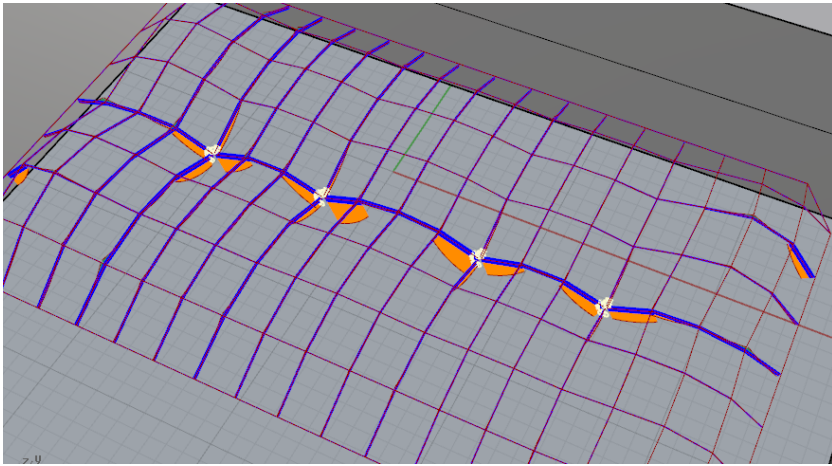
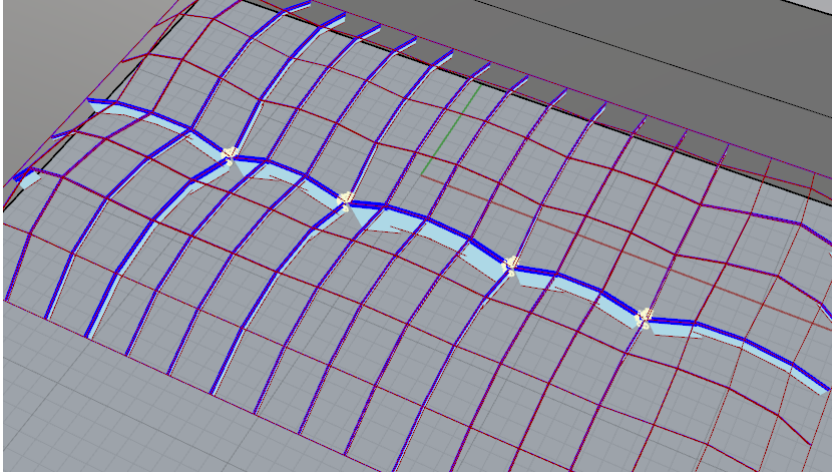
As expected the biggest deflection occur at the middle of the span. And all happen to be below the maximum allowable deflection. UC most deflected element ; 1, the requirements are met.

Axial forces

As expected the axial forces are mainly compression forces, normative at the supports.

Bending moment

The bending moment is limited and the positive bending moment occurs also at the parts with the biggest span. Highest moment are found near the inner supports. Figure below shows bending moment:



Conclusions and discussion

During the design, we noticed that the connections between the secondary beams cause a big decrease of transparency. It was decided to reduce this effect and eliminate most of the secondary beams. Also, during the design we tried to develop a structure with solely flat panels, this did not lead to a satisfying design and therefore we dropped the percentage of flat panels to about 80%. A further optimization of

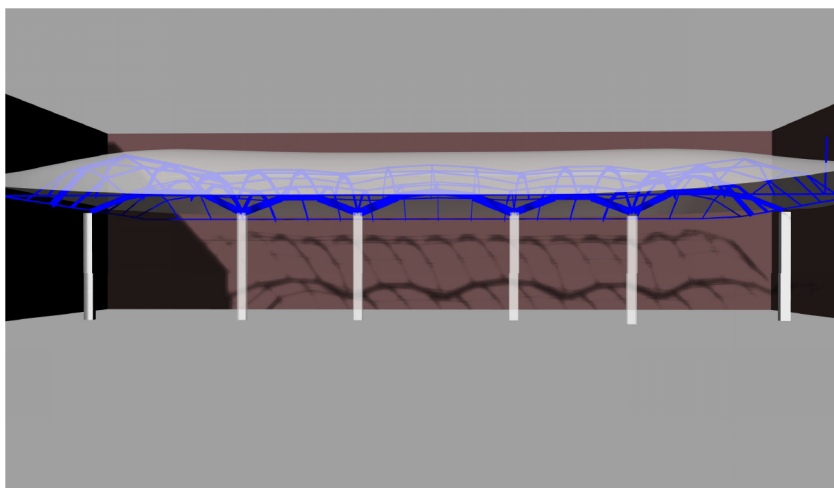
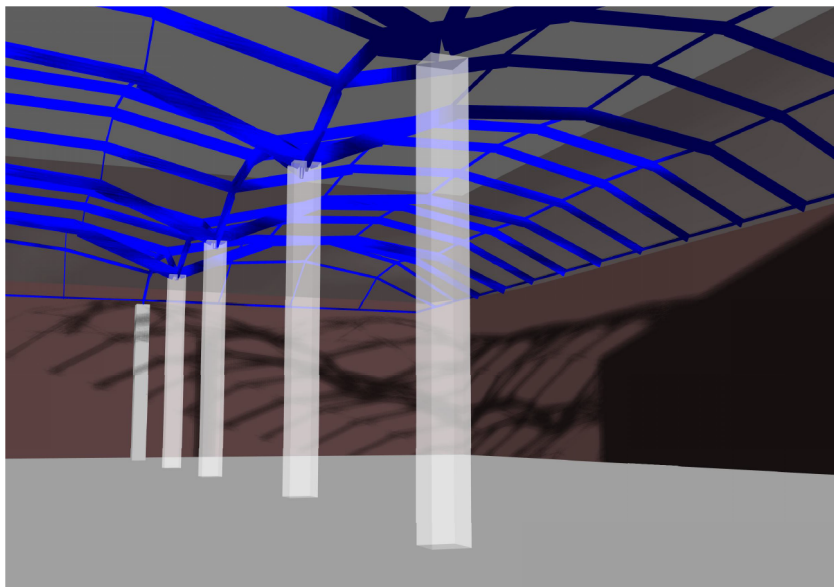
shapes in compression, but with solely flat panels is recommended. The solution can also lie in the fixation of the panes to the underlying structure. If these can bear the differences, without losing transparency. This could even lead to an interesting 3d arrangement.

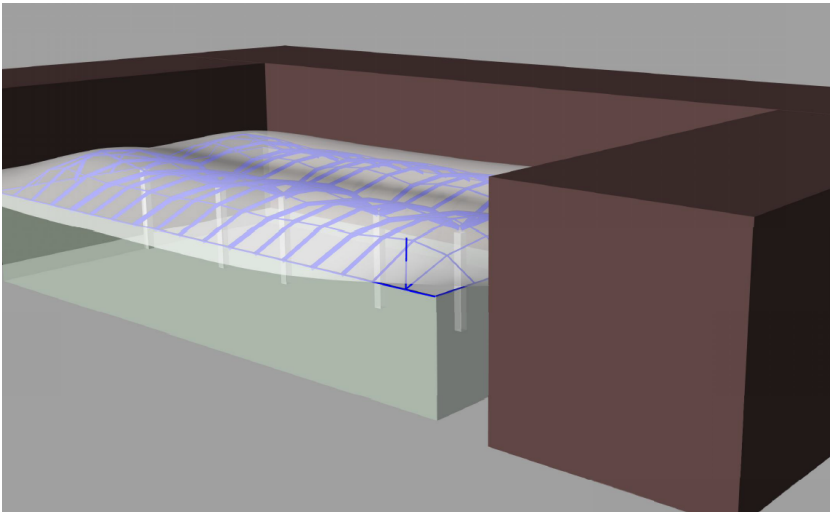
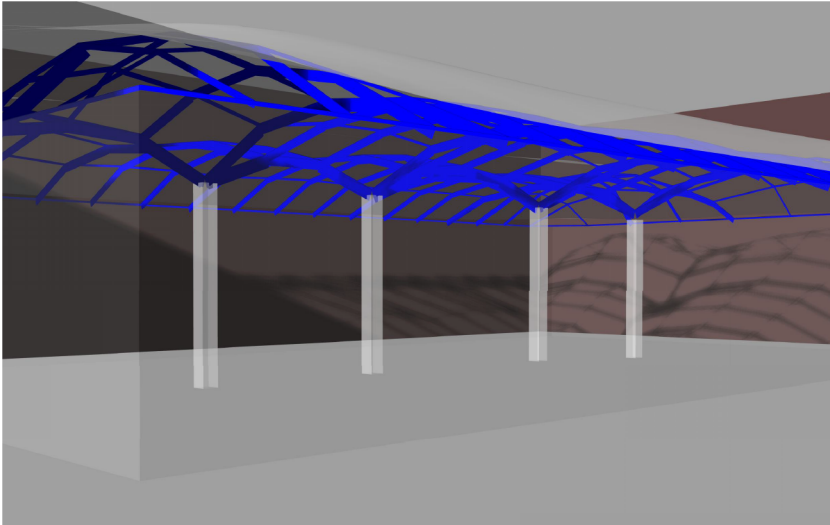
The choice to support the edges on a small interval, without using an edge beam, resulted in a load bearing facade with glass fins. For future development, this might be reconsidered. Depending on the execution the facade with fins can be almost as transparent as without, but they draw quite some attention. As can be seen in the image below.

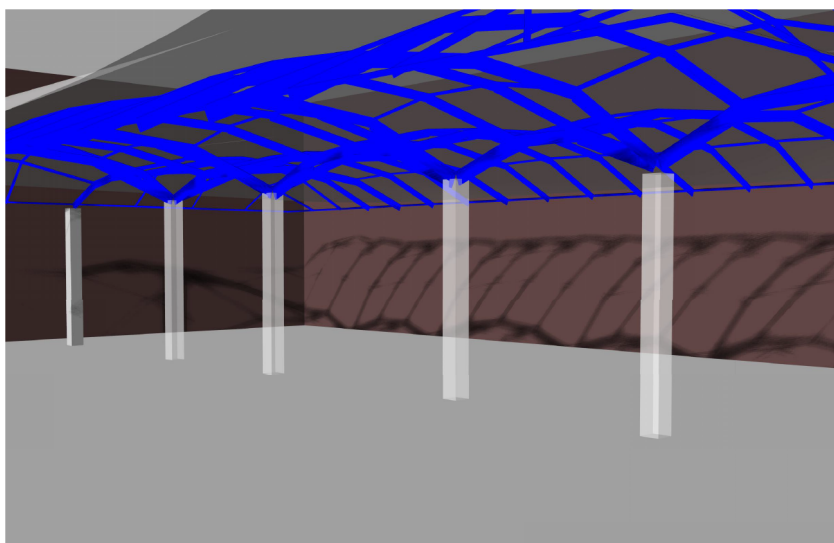
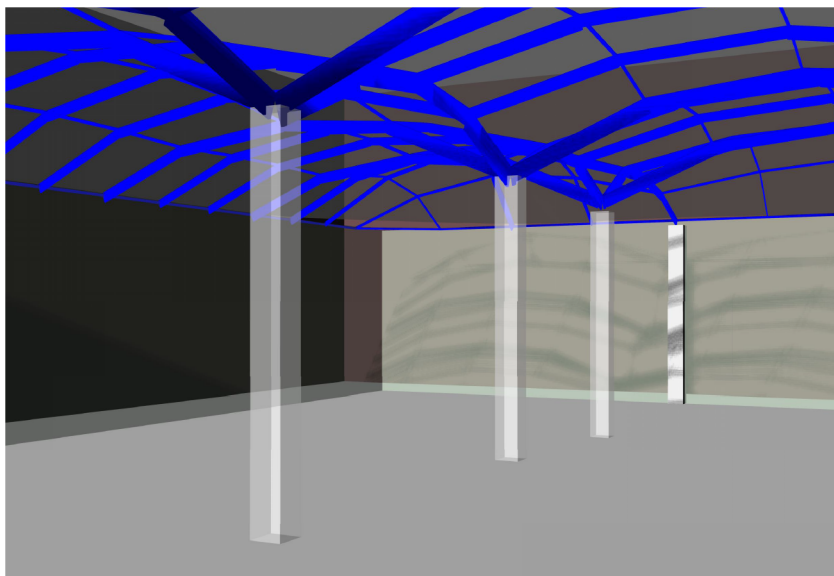


It is very important to perform a detailed finite element analysis on the connections of the design, to check the local peak stresses. Due the parameterized design method, It is possible to use the same method for different projects or with change of requirements. It is recommended to extend the functionality of the design tool, in order to fully utilize its potential.

Visualizations







Appendix E

Case Study 3

The third case study explores the use of a new material; timber. It concerns a design for the great exhibition, also known as the Crystal Palace exhibition. The exhibition was the first universal Exposition and many more followed, used to expose the most recent developments and state of the art products of a countries involvement and to draw attention to urgent matters.



The aim of this assignment is to provide a design for a high-profile pavilion for the Expo 2020 in Dubai. With help of the structural principles discussed during the course, using generative and algorithmic design. The main goal is to translate the requirements into a design which needs to be converted to a logical and buildable structure.

The many pavilions build for previous editions function as inspiration for the design. For this assignment, we stepped into the role of both the designer and the structural engineer. The first part provides a brief overview of the structural design considerations and concludes with a choice of concept. The second part represents the structural optimization and analysis of the chosen design. The report is concluded with a representation of the final design for the Dutch pavilion of the World Expo 2020 in Dubai.

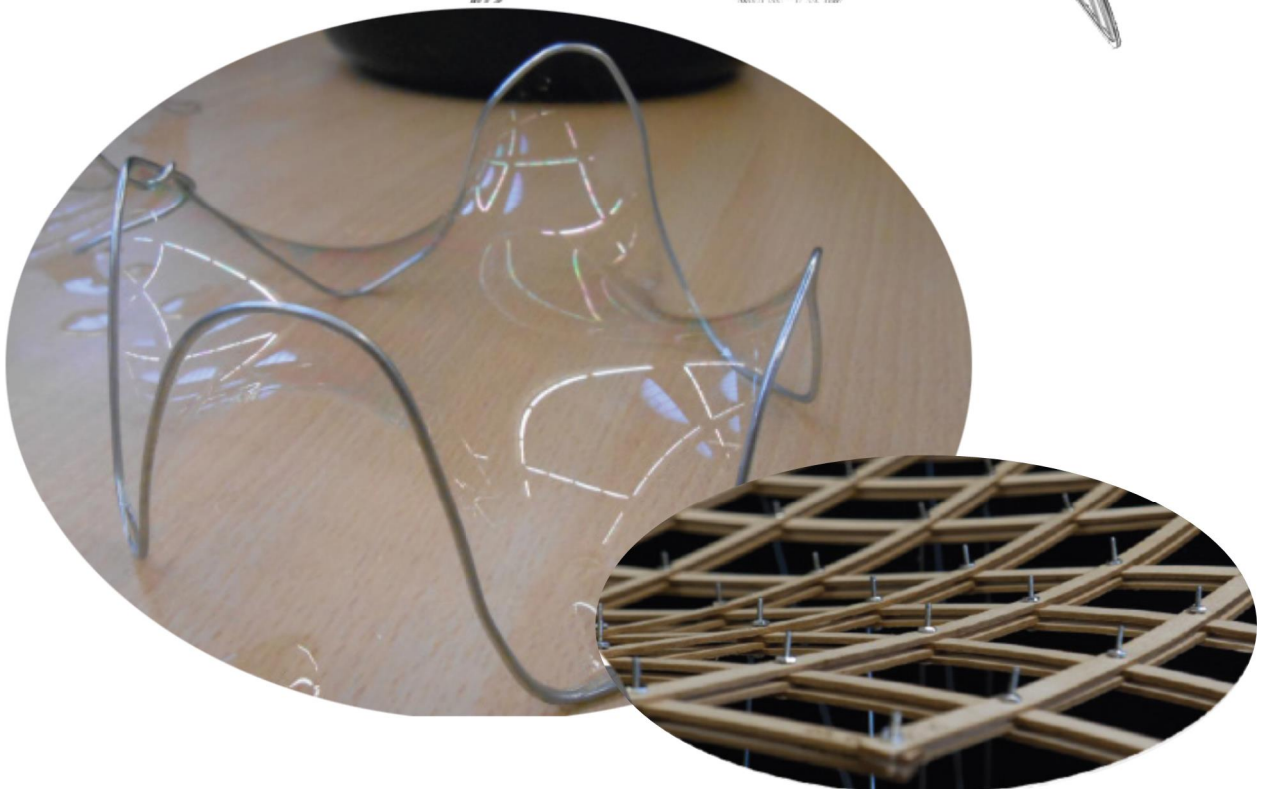
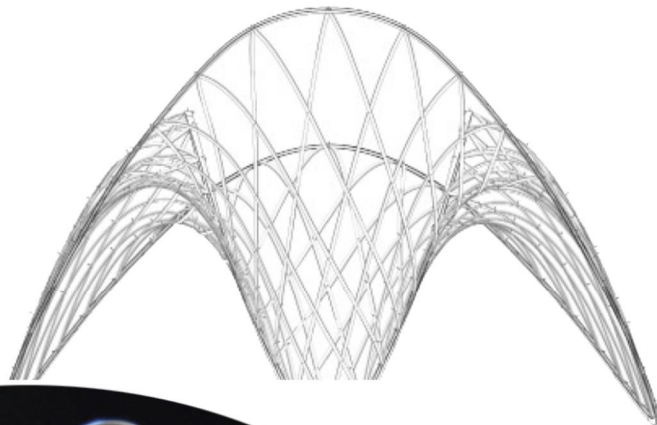
Conceptual design phase - Orientation

For this concept, we explore a different kind of special structure, the grid shell. This structural principle is suitable for two very interesting types of form finding, which can be explored with several types of form finding; the minimal surface (generated with soap films) and the funicular surface (generated with the hanging model). We would like to further invest the fascination for the hanging models, used by many engineers and architects, like Gaudi.

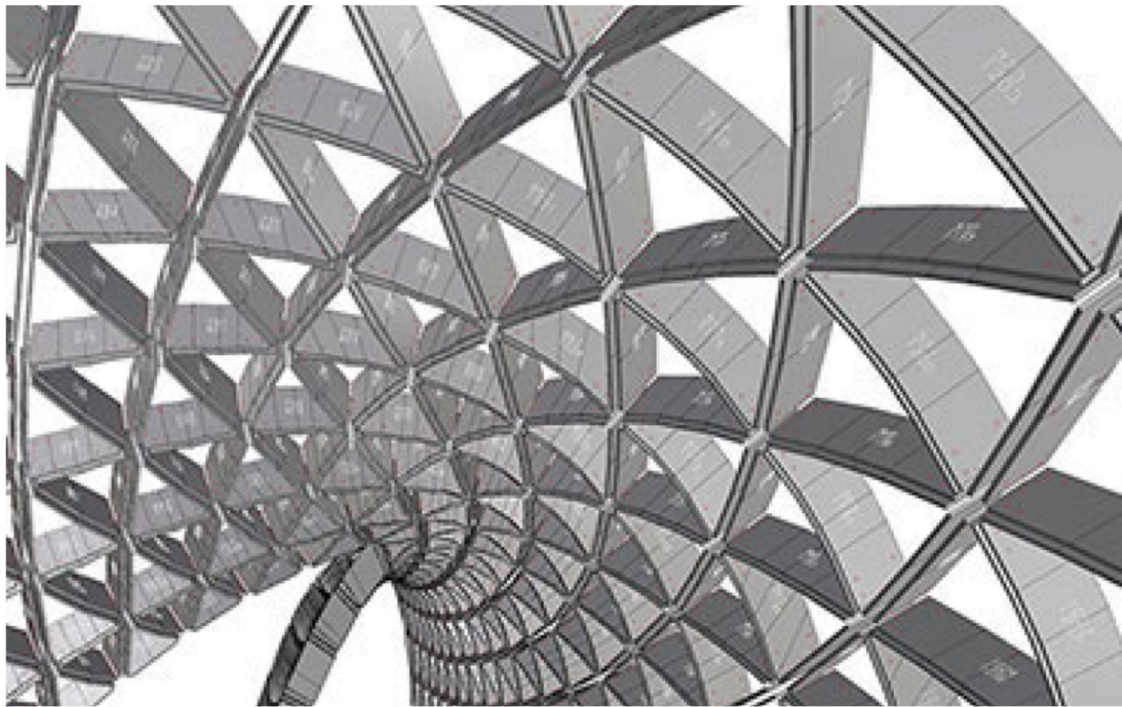
A very impressive reference is the Multihalle of Mannheim, both in its size and method. The shape is generated with a hanging model. The model is subjected to the gravity and fully under tension, when reversed the model is fully under compression when no other loads are applied.

Different distributions or patterns are possible, grid, triangulation or a fluctuating distribution.

- + Interesting structure
- + Suitable for parametric design
- + Several materials possible
- + Efficient load distribution
- + No additional supports needed
- Labor intensive
- Relatively high costs
- Long erecting time
- Limited supports: risk for collapse.



We want to use the funicular surface to function as base of the model and project the beams on this surface. This procedure provides a lot of design freedom and is very suitable for parametric and algorithmic design.



The requirements are discussed in the introduction and will be further discussed in the next part.

Design variants

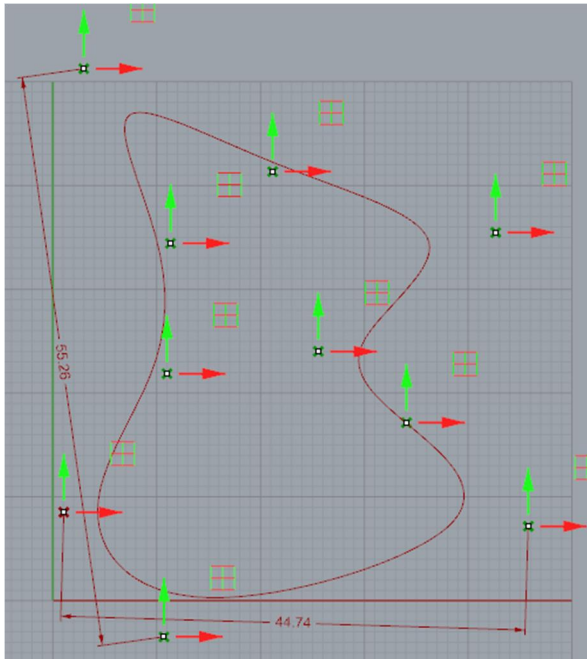
The first module is scripted in a way to find the potential shape and consist of two parts; the boundary conditions and form finding.

Boundary conditions:

For the boundary conditions, we set the different requirements, stated in the chapter before. The most important aspect is the visitors flow. The building needs to be big enough to house all visitors, but we also want a routing in the pavilion, to guide this flow. The third requirement is the openings for the visitors flow, these need to be big enough to avoid congestion; we set the entrance and exit to be at least 10 meters wide.

The perimeter of the pavilion is defined by a NURBS curve and its control points as can be seen in the figure. These points can be controlled from both the Grasshopper script as Rhino's interface. Due to the algorithmic design, it is possible to change the boundaries throughout the entire design and analysis process, and therefore for the optimization of the structure.

The scenery routing, to guide the users, will be generated with strategically placing of circular columns inside the pavilion. This also creates a waving surface, with local high peaks, to simulate the wavy and round character of the sea, and a more interesting structure.



Form finding

The first optimization already starts at the generation of the shape. To minimize the tensile stresses, we revert to the hanging model, that is subjected to gravity. Karamba provides the possibility to simulate the behavior of hanging models, with help of the “Analyze large deformation” component. The geometric non-linearity is handled by an incremental approach: The external loads get applied in steps. After each step, the model geometry updates to the deflected state. The more and the smaller the steps, the better the approximation of geometric non-linearity. The purely incremental method however incurs an unavoidable drift from the exact solution. For form finding this error should be negligible in most cases.

In the image below you see the parts that are used for supports. It's visible that not the entire boundary is supported to increase the irregular and wavy character of the design.

Structural properties

We also parameterize the structural properties in the script.

Supports:

The supports for the form finding coincide with the supports for the structure. Therefore, we can use the same output as described above

Material:

We implemented the material properties for wood. This is set by default, but can be adjusted by the user. In case people want to compare different material, this is a possibility. Optimization between materials is something that could be a future addition.

For safety we use conservative values for the yield strength. We use the following values:

Young's modulus	E:1050[kN/cm ²]
Shear modulus	G:360[kN/cm ²]
Density Gamma:	6[kN/m ³]
Thermal expansion, alphaT:	5.0E-6[1/C°]
fm;0;d = fm;0;repkmod/gM = 40·0,85/1,2:	28 N/mm ²

Cross section

We use one cross section for the edge beam and the possibility for 10 cross sections for the inner elements, which we will use for optimization.

Assemblage

Eventually all elements are converted to Karamba elements and all properties are assembled in one component.

Load generator:

For the analysis, we'll apply a load generator. The most important feature of this module, is the interactivity with the parametric structure and the freedom for the user to apply, adjust and remove loads. This is a big advantage compared to the use of GSA.

For the load distribution, we distinguish the permanent and variable loads and consider combinations of the two to check the serviceability and ultimate state. The values that were applied are described below:

Permanent Load:

dead load

The self-weight of the structure and dead load of the façade elements or cladding subjected by gravity. These are implemented in the script and linked to the elements.

Density of timber 600 kg/m³

Density of façade fabric: 0,4 kg/m²

The dead load depends on the cross section of the elements and therefore interactive in the script.

Variable load:

wind load:

The wind pressure is estimated 1 kPa, so 1000 N/m.

$F_{wind} = C_s \times C_d \times C_f \times q_p(Z_e)$

Compression wind load: $Q_d = 1,0 \times 1,0 \times 0,7 \times 1 = 0,7 \text{ kN/m} = 700 \text{ N/m}$

Tension wind load: $Q_s = 1,0 \times 1,0 \times 0,25 \times 1 = 0,25 \text{ kN/m} = 250 \text{ N/m}$

We consider the wind load as both horizontal and vertical loading. For further analysis, we recommend taking asymmetric loading into account as well. For now, we decided to focus on the optimization and neglect this effect.

Snow load: since the change of snow in Dubai in the period of January to June is highly unlikely we don't consider this as a governing load and don't take the load into account.

Maintenance: For this conceptual design phase, we will ignore the point load of 1kN, that is more favorable than the wind load.

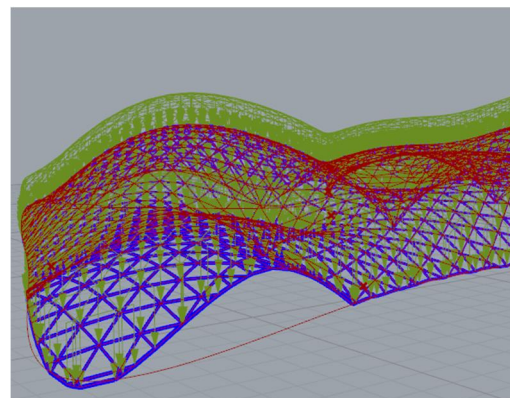
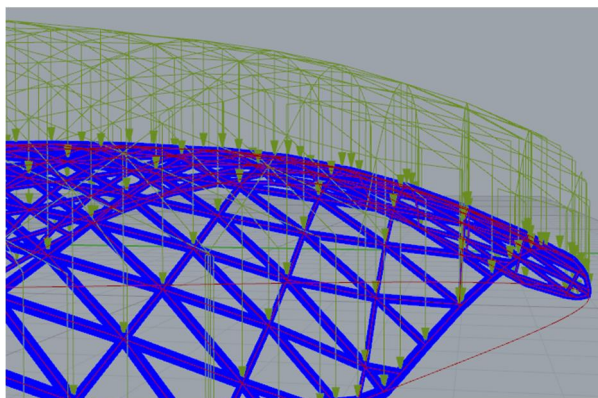
Load combinations:

We will allow for the following load combinations, the grey combinations have not been taken into account:

ULS	Load case	safety factor
1	self-weight only (G)	1,35*G
2	self-weight plus snowload (Qsn)	1,2*G+1,5*Qsn
3	self-weight plus wind upwards (Qw+)	1,2*G+1,5*Qw+
4	self-weight plus wind downwards (Qw-)	0,9*G+1,5*Qw-
SLS		
5	self-weight only	1,00*G
6	self-weight plus snowload (Qsn)	1,0*G+1,0*Qsn
7	self-weight plus wind upwards (Qw+)	1,0*G+1,0*Qw+
8	self-weight plus wind downwards (Qw-)	1,0*G+1,0*Qw-

We will derive the governing forces, moments and stresses from the most critical Ultimate Limit state combination, and the deflection from the most critical Serviceability Limit State.

In the figure below we show how the loads are assigned to the structure. The variable load is projected onto the structure and redistributed to the grid and the dead load is connected to the elements and their material properties.



Structural optimization

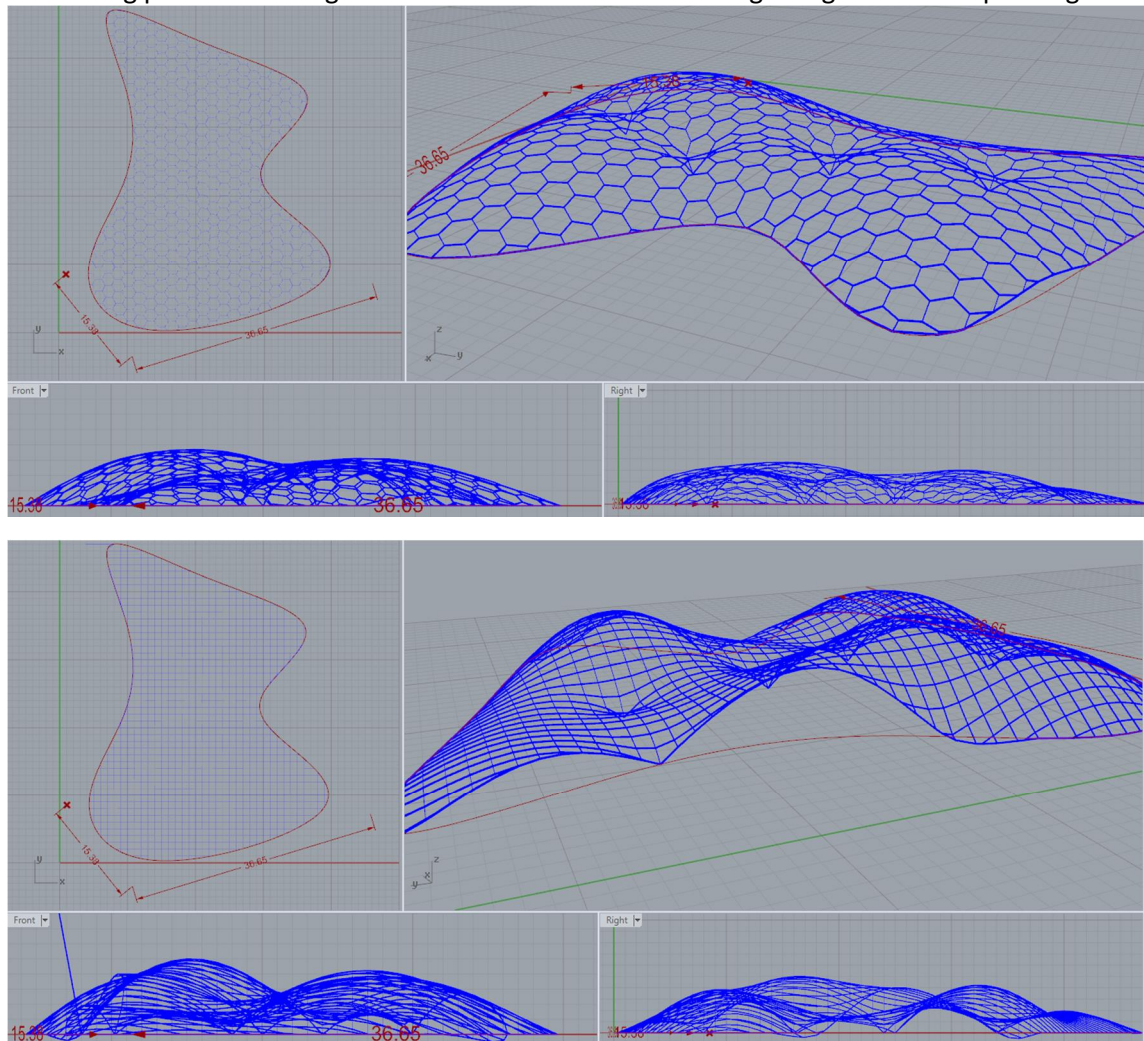
For the optimization, we make use of different options. The first step is done in the form finding, with the help of the hanging model. This will generate one extra optimization parameter, the overall height. The boundary conditions also provide the possibility to optimize the structure: with the location of the inner columns and outer supports. The grid pattern and size is also used as optimization variable and had a considerable influence on the structural feasibility.

Cross section

For the optimization of the cross section we use an algorithm that assigns the smallest cross section that suffices the maximum stress and deflection. The list of available sections is described in the section before. We optimize all internal elements. The outcome shows if all elements meet the requirements. The optimization component is connected to the structural analysis and material properties.

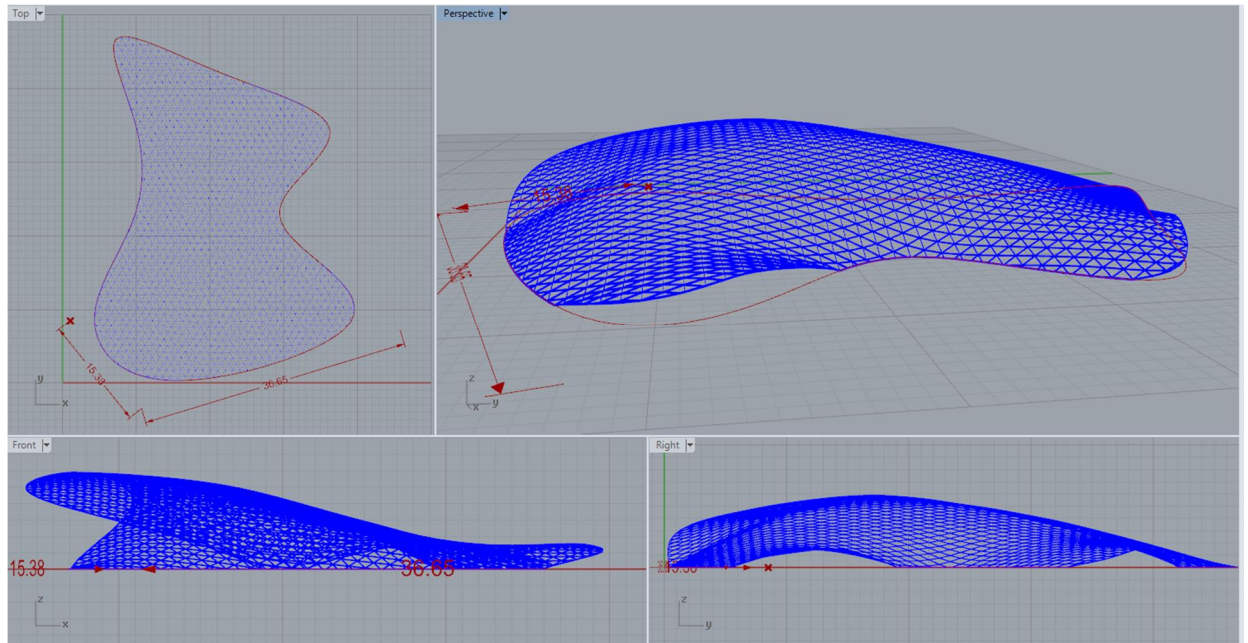
Further optimization

We used the maximum deflection and stresses as leading for the optimization, keeping the architectural value in mind. First, we looked at the effect of the different patterns. The triangulated grid showed the smallest deformations, as expected. Compared to the hexagonal grid, it's much easier to be erected on site, therefore we choose this pattern over possible more interesting patterns. The figures below show the use of a hexagonal grid and 2d squared grid.

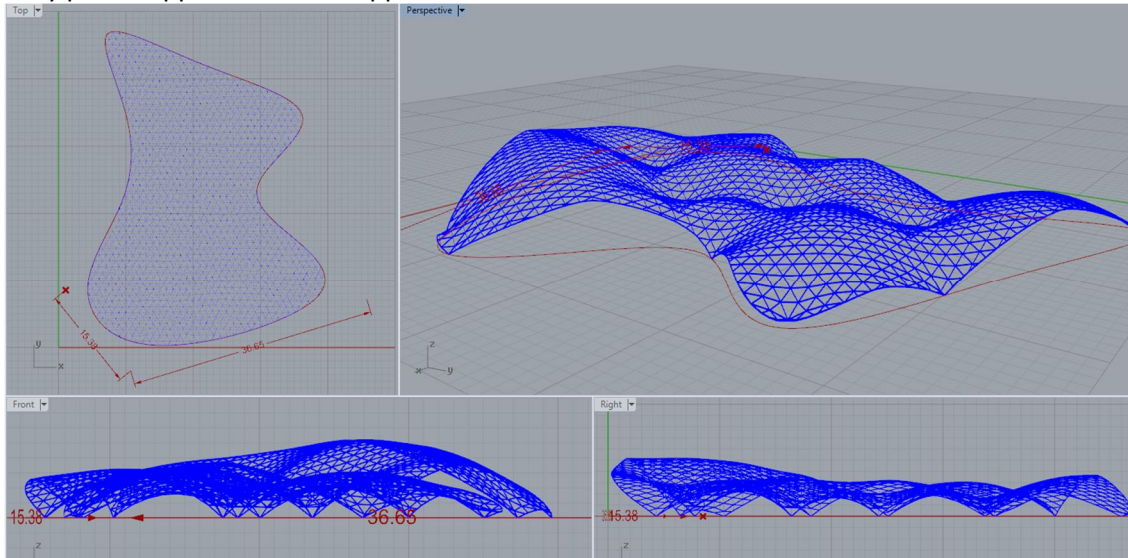


The next step, was the placing of the columns. We tried different options, a combination of internal, external, line and point supports. For both structural as architectural point of view we decided to use a combination of internal point supports and external line and point supports. This would limit the span and at the same time create an irregular wavy shape.

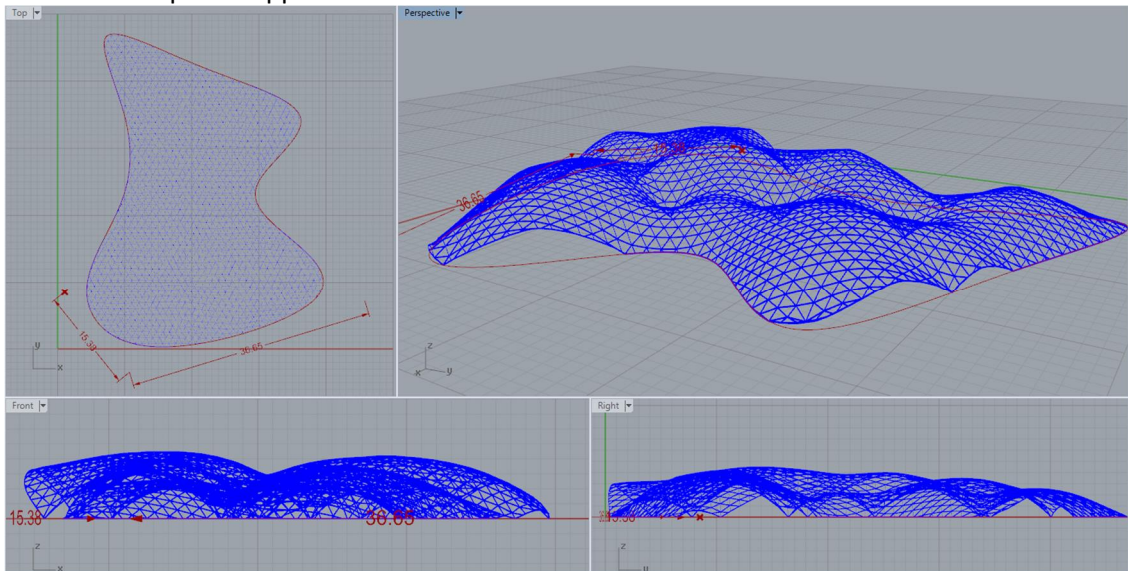
Optimization of supports: No internal columns, only line supports:



Only point supports: no line supports:



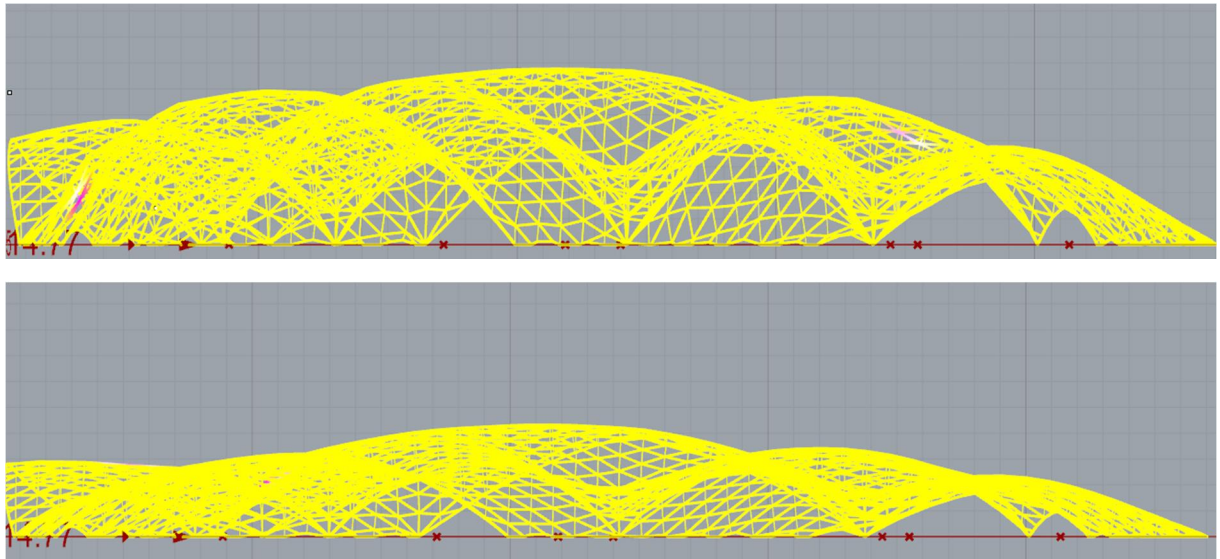
Both line and point supports:



Subsequently, we adjusted the overall height of the structure. We noticed that this parameter has no linear effect on the optimization. Due to the complex shape the height has a different effect on each part and there for a change in height has a positive effect on a certain part at the expense of an adjacent part. We vary between 6 and 20 meters and find an optimum at 8 meters.

Another optimization was done with the mesh size. We reduced the size of the grid to one meter, to reduce the maximum deflection of the structure.

The last part of the optimization consists of refinement of the height, exact location of the supports, grid size, the curves of the perimeter. This was necessary to create the final and feasible design as described in the next part. The figure below show the effect in change of height.

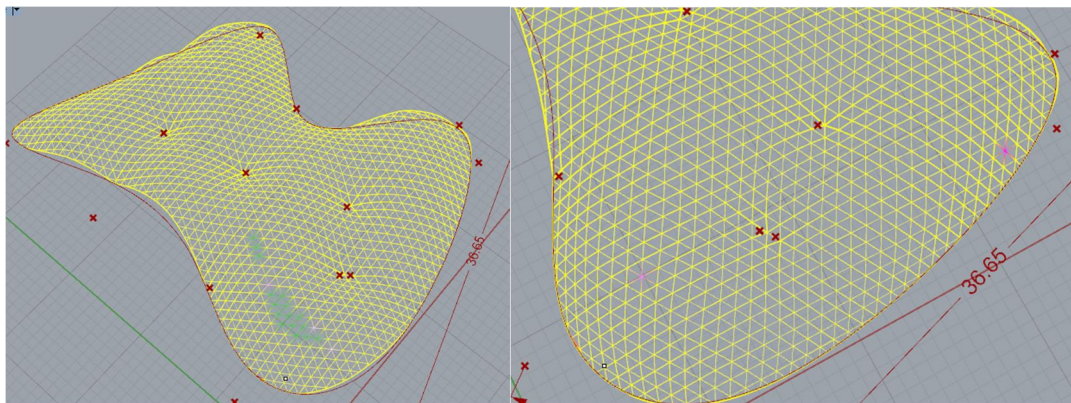


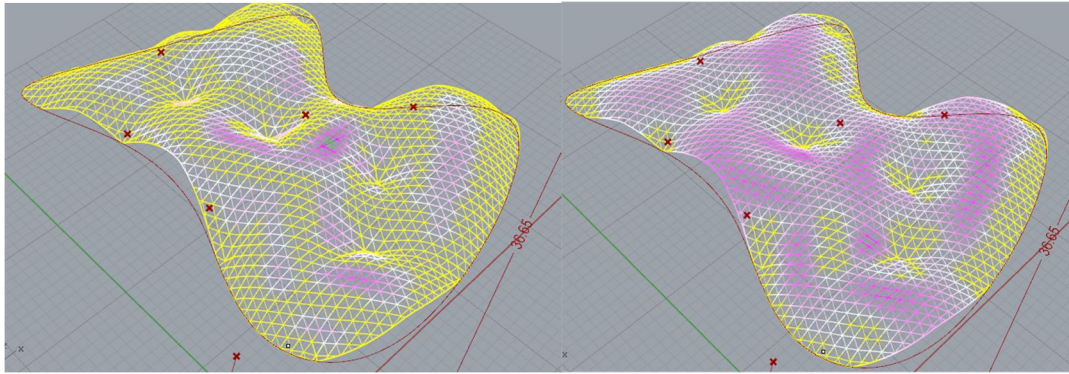
Find the optimum of the structure concerning the material use. When using a denser mesh, the cross section need to be bigger, there's an optimum to be found. This is depending on different parameters, including not only the mesh size, but also the pattern and cross section shape.

Final optimization

We tweaked the different buttons in order to keep optimizing and reducing the overall material. At a certain moment, it became difficult find an optimum. This is caused by the interconnection of the entire structure.

The four images below show the process to find an optimal shape. At first the maximum displacement exceeded above the entrance. Then at just one node above the entrance, but also near the exit. After reshaping the boundaries, we moved the critical deflection to the center of the structure. After changing the height, boundaries and inner columns, we generated a structure that didn't exceed the maximum deflection nor stresses.





Conclusion:

It is clear that the integral tool provides instant visual and numerical feedback, therefore the impact of a design decision is very transparent. However, with a complicated design as this project, a lot of components are intertwined and therefore the optimization can become quite random, since the relationships are not always directly visible. We can conclude that the optimization part was a bit more complicated than expected, because of the interactivity of the complex shape.

While changing these settings, we received instant visual feedback of the structural analysis, which made it possible to design while structurally optimizing the structure. However, an adjustment to resolve a local problem, shifts the problem to a point elsewhere in the structure. This made it difficult to resolve. As this will also be the case when using external finite element methods, we can conclude that the integral process has an advantage as opposed to the more traditional verification.

Structural analysis

The structural analysis is performed with the analysis module in Karamba and shows, as mentioned before, instant results. All optimization models are calculated with the load combinations as described before. For the final analysis we'll discuss the outcome of the optimized structure for the critical combination, ULS LC3 and SLS LC 7. We will briefly discuss the outcome.

Axial force:

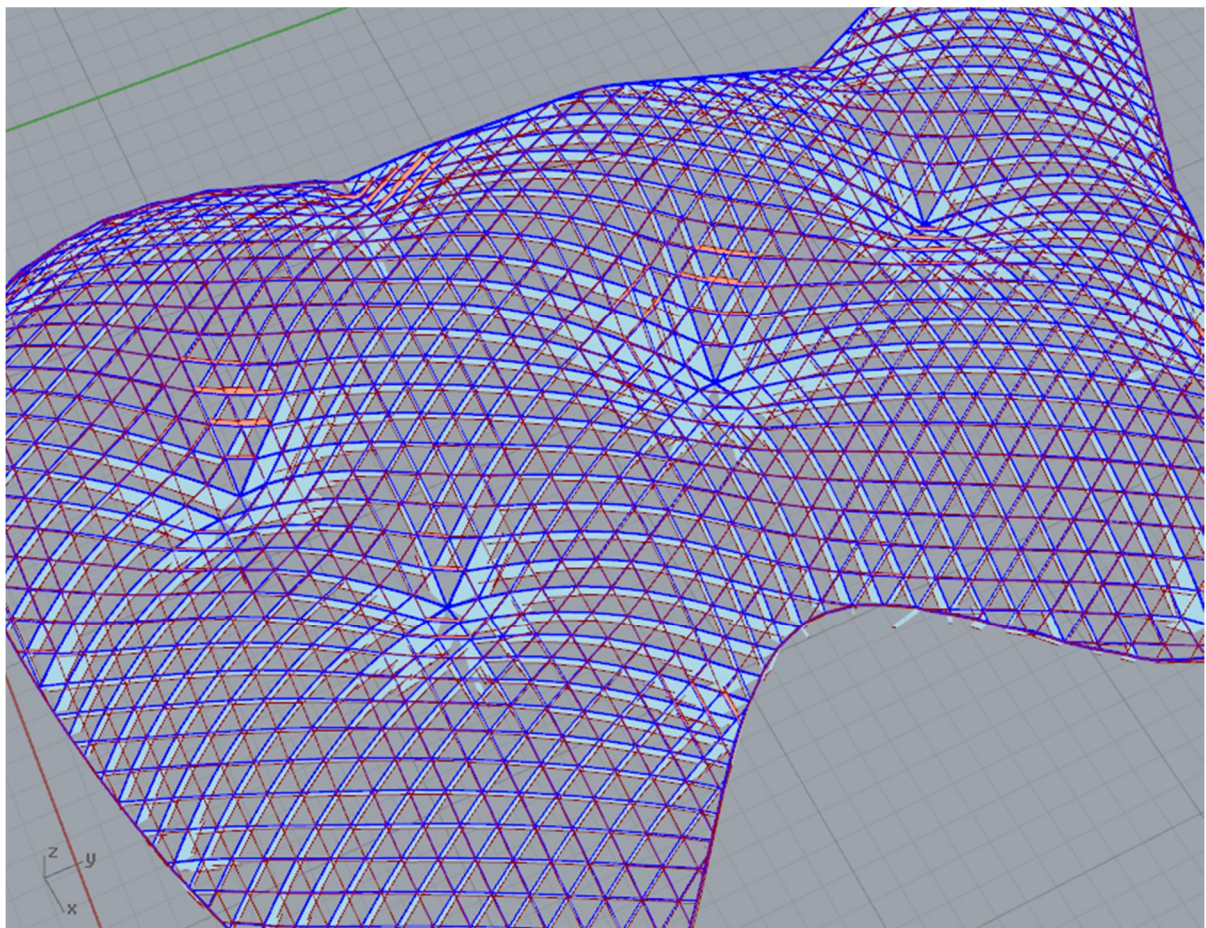
Compression 190 kN

Tension 20 kN

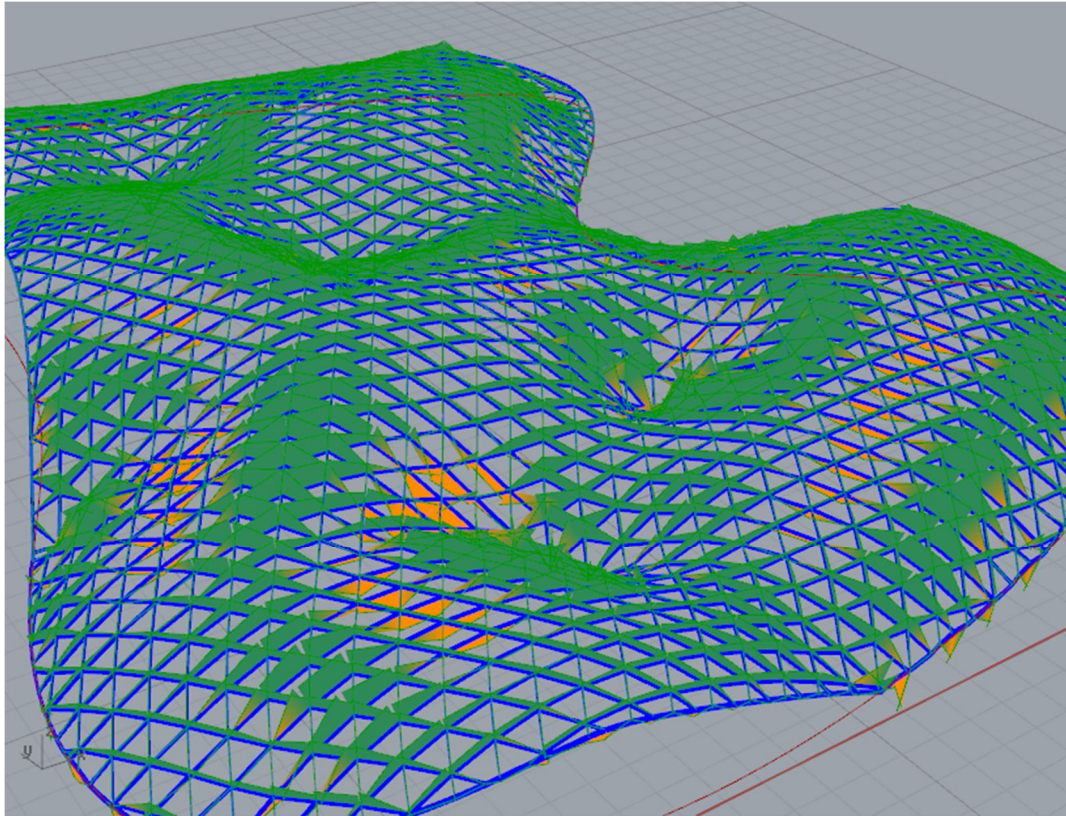
Shear force: 9 kN

Moment: +0,7 kNm

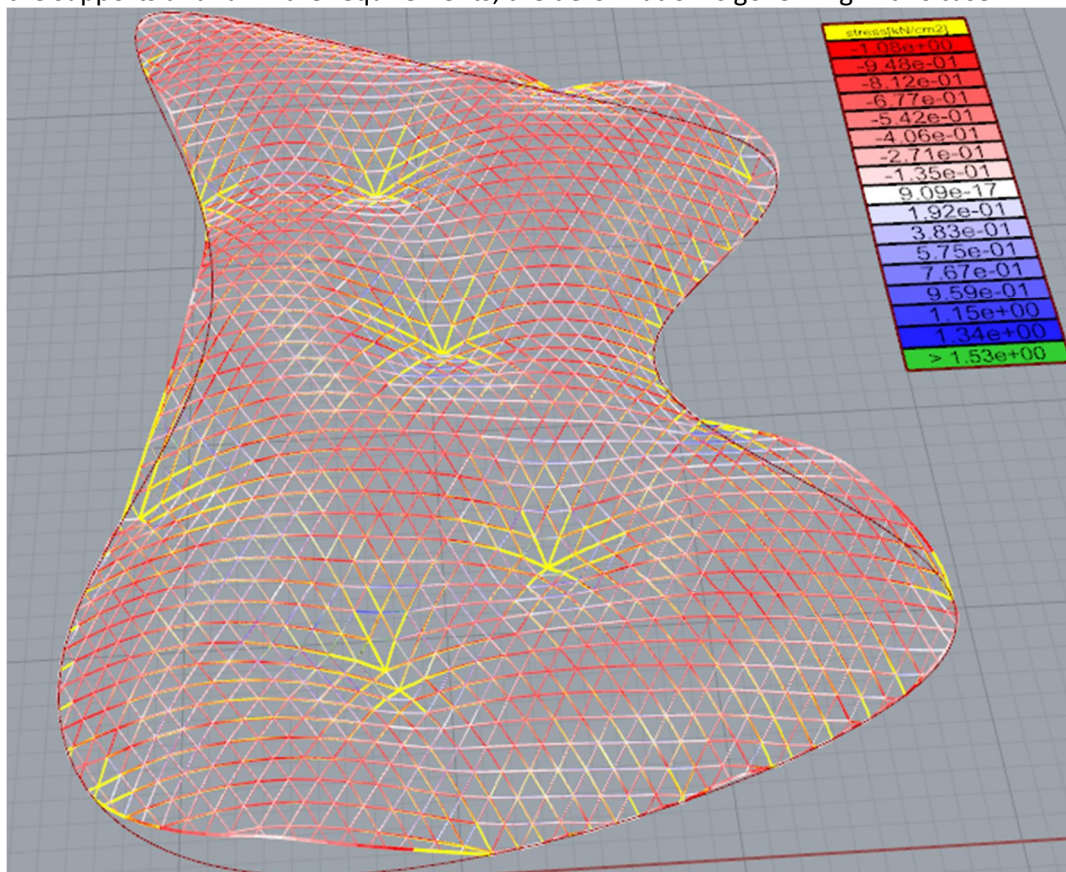
As expected the axial forces are mainly compression forces, normative at the supports. We see some tensile members at parts with larger spans, this could be expected. Figure below shows axial forces.



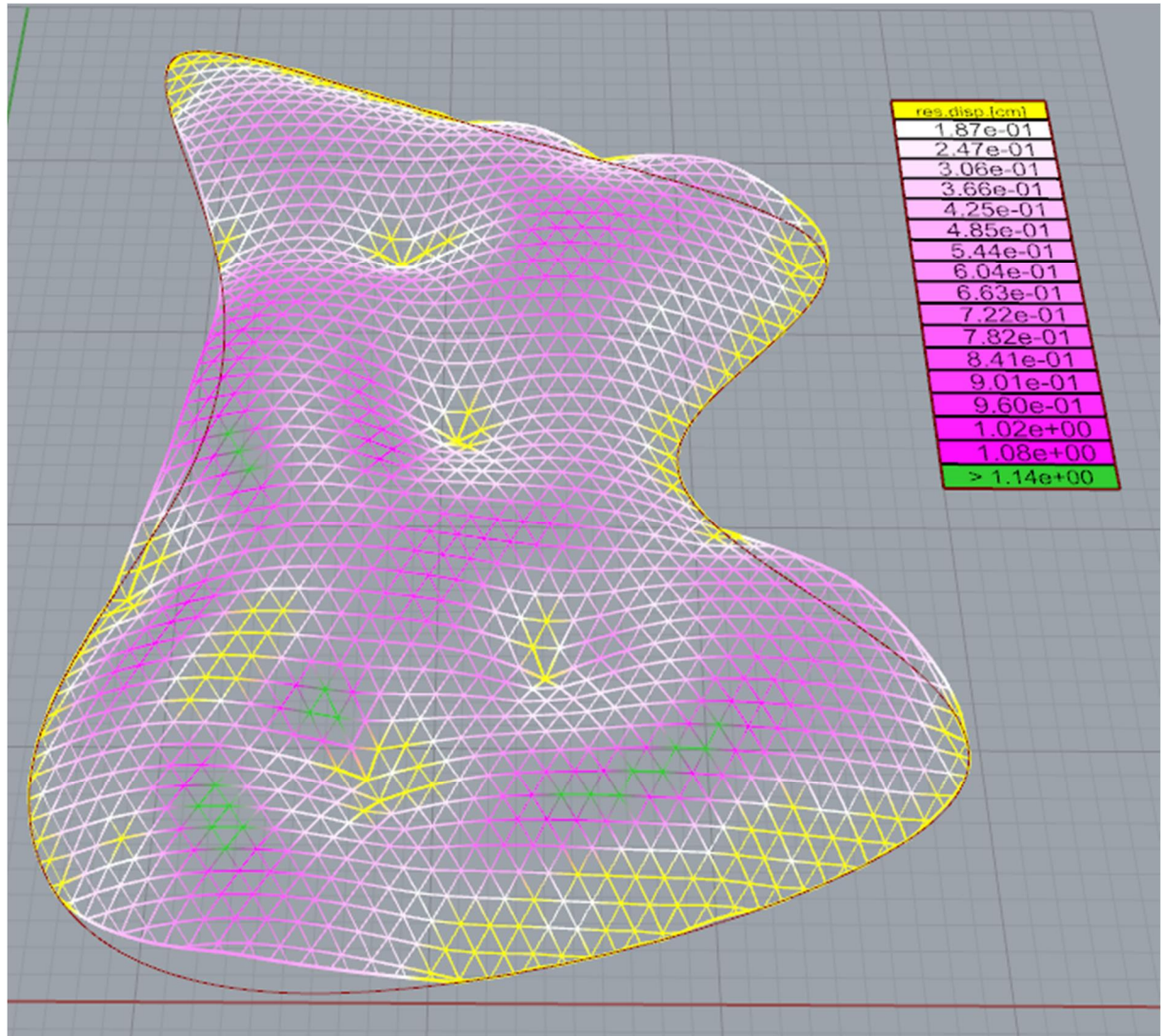
The bending moment is limited and the positive bending moment occurs near the entrance and exit, also the parts with the biggest span . Figure below shows bending moment.



In the figure below, we see that, due to the form finding, the lowest stresses can be found near the supports and fulfill the requirements; the deformation is governing in this case.



Deflection:



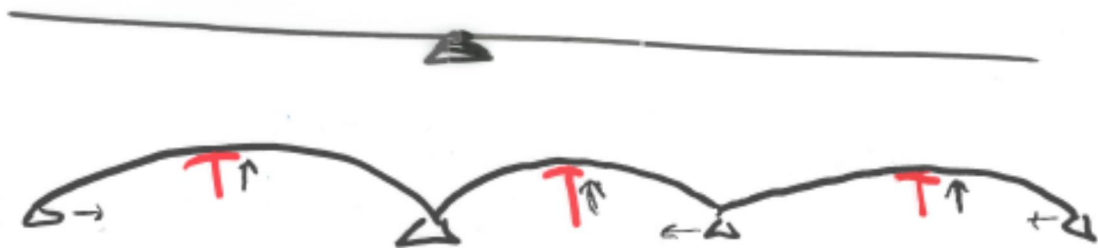
Biggest deformation: 13 mm

Span: $35/250 = 14$ mm

UC most deflected element < 1 , the requirements are met.

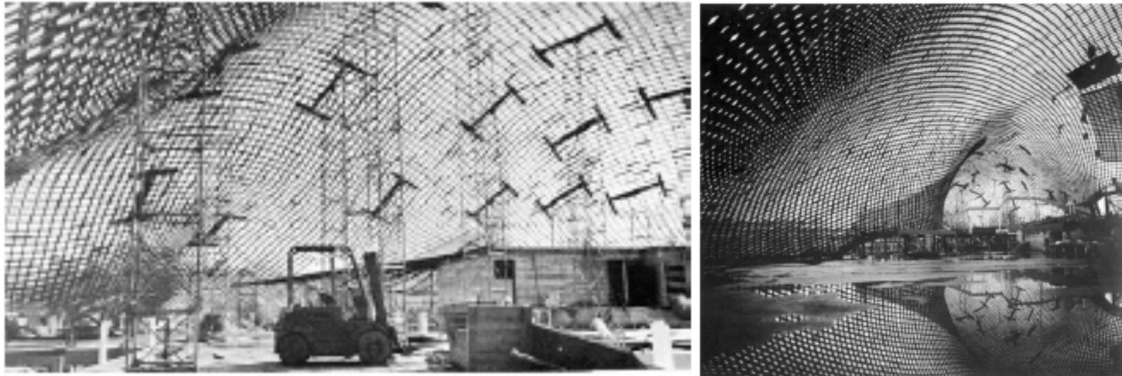
Detailing and erection

We designed rotational connections, to make it possible to erect the structure as a hole. One middle support will be fixed, while the others are pushed towards this center point. Lastly the side supports are pushed into place. This process is performed with jacks, from under the grid, that push the structure upward.





Erection of the Multihalle in Mannheim, with help of fork lifts pushed into place.



The erection scheme is shown below:



Detailing

To allow this kind of erection method, the connections need to be able to rotate, since the timber elements need to be able to shift, relative to each other.

The timber elements are connected in 3 different layers to each other. To provide the rotational freedom for construction we'll use the pinned principle in combination with shear blocks.

The supports are hinged connections that offer the freedom to rotate freely from horizontal, needed at erection, to the angle needed for the structure. As soon as the supports are pushed into place, they'll be fixed to the concrete underground.

The fabric will be placed at a small offset of the structure on a rails.

