NTNU
Norwegian University of
Science and Technology

The Digital Workflow of Parametric Structural Design

Developing Grid Shells in a Nordic Climate

Marie Eliassen
Åshild Huseby

Civil and Environmental Engineering
Submission date: June 2018
Supervisor: Anders Rönnquist, KT
Co-supervisor: Marcin Luczkowski, KT

Norwegian University of Science and Technology
Department of Structural Engineering

Department of Structural Engineering
Faculty of Engineering Science and Technology
NTNU- Norwegian University of Science and Technology

ACCESSIBILITY

MASTER THESIS 2018

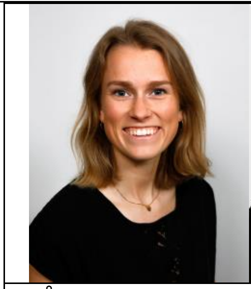
SUBJECT AREA: Conceptual Structural Design	DATE:05.06.2018	NO. OF PAGES: 201

TITLE:
The Digital Workflow of Parametric Structural Design
Developing Grid Shells in a Nordic Climate

Digital Arbeidsflyt i Parametrisk Design av Konstruksjoner
Utforming av Gitterskall i et Nordisk Klima

BY:

Åshild Huseby
Marie Eliassen


Åshild Huseby

Marie Eliassen

SUMMARY:
Unusual shapes are becoming more and more prevalent in the cityscape, however they often rely on large and complex configurations of material. New and empowered digital tools - parametric modelling software - allow for continuous structural optimisation parallel with the design process. This enhances the synergy between architects and engineers and complements the search for alternative solutions with a better use of resources. Shell structures provide the opportunity of combining artistic shapes and low use of material due to their efficient load bearing capacity. The efficiency of the shell is dependent on its shape and gets challenged when exerted to external asymmetric loading pushing it out of its ideal shape. Until now, previous research papers and articles have provided information about form finding of shell structures exposed to symmetric pressure while omitting asymmetric loading. This leaves an important question of how to cope with such load situations? In this thesis, the objective thus became to come up with a parametric design procedure for a timber grid shell where the structure was robust enough to handle the asymmetric pressure from climate induced loads. The grid shell in question is to be a cabin built by the students at the Norwegian University of Science and Technology (NTNU) for the use of NTNU students in the Norwegian mountains. Through the studies in this thesis it was found that Eurocode (EC) provides limited information about design loads for shell structures, and that it can only be adopted to a certain extent. By comparing the EC loads to a load configuration-algorithm built in Grasshopper, it was discovered how EC did in fact not give the worst-case load scenarios. However, the load distributions obtained from the algorithm showed some contradictions with regards to the guidelines in EC and were thus not accounted for in the design. It was illustrated how the shape is crucial for the efficiency and structural behaviour of the shell, and how double curvature gives a more robust structure which is less prone to deflect. Optimisation of the cabin's shape was executed through minimising displacements from external loads. Initially the shape showed substantially larger displacements for the Drifted (asymmetric) snow load case from EC compared to the Undrifted (symmetric) case, and the optimised shape displayed an increase in the initial double curvature. The optimisation approach led to an improved structural performance and a smaller gap between the structural response for the Undrifted and Drifted load cases.

RESPONSIBLE TEACHER: Anders Rønnquist

SUPERVISOR(S): Anders Rønnquist and Marcin Luczkowski

CARRIED OUT AT: Department of Structural Engineering, NTNU, Trondheim

Abstract

Unusual shapes are becoming more and more prevalent in the cityscape, however they often rely on large and complex configurations of material. New and empowered digital tools - parametric modelling software - allow for continuous structural optimisation parallel with the design process. This enhances the synergy between architects and engineers and complements the search for alternative solutions with a better use of resources. Shell structures provide the opportunity of combining artistic shapes and low use of material due to their efficient load bearing capacity. The efficiency of the shell is dependent on its shape and gets challenged when exerted to external asymmetric loading pushing it out of its ideal shape. Until now, previous research papers and articles have provided information about form finding of shell structures exposed to symmetric pressure while omitting asymmetric loading. This leaves an important question of how to cope with such load situations? In this thesis, the objective thus became to come up with a parametric design procedure for a timber grid shell where the structure was robust enough to handle the asymmetric pressure from climate induced loads. The grid shell in question is to be a cabin built by the students at the Norwegian University of Science and Technology (NTNU) for the use of NTNU students in the Norwegian mountains. Through the studies in this thesis it was found that Eurocode (EC) provides limited information about design loads for shell structures, and that it can only be adopted to a certain extent. By comparing the EC loads to a load configuration-algorithm built in Grasshopper, it was discovered how EC did in fact not give the worst-case load scenarios. However, the load distributions obtained from the algorithm showed some contradictions with regards to the guidelines in EC and were thus not accounted for in the design. It was illustrated how the shape is crucial for the efficiency and structural behaviour of the shell, and how double curvature gives a more robust structure which is less prone to deflect. Optimisation of the cabin's shape was executed through minimising displacements from external loads. Initially the shape showed substantially larger displacements for the *Drifted* (asymmetric) snow load case from EC compared to the *Undrifted* (symmetric) case, and the optimised shape displayed an increase in the initial double curvature. The optimisation approach led to an improved structural performance and a smaller gap between the structural response for the *Undrifted* and *Drifted* load cases.

Sammendrag

Uvanlige og arkitektoniske former blir mer og mer vanlig å se i bybildet, men de består ofte av kompliserte løsninger og overflødig materialbruk. Digitalisering og framskritt innen ny teknologi som parametrisk modellering, gjør at optimaliseringsprosessen og konstruksjonsanalyse kan kombineres og utføres parallelt med designprosessen. Dette legger til rette for et bedre samarbeid mellom arkitekter og ingeniører, og kan bidra positivt mot et slutt-design med mer effektivt materialbruk. Skallkonstruksjoner er en type konstruksjoner som tillater kombinasjon av artistiske former med lavt materialforbruk på grunn av skallets effektive bæreevne. Skallets effektive oppførsel er svært avhengig av formen og blir svekket når asymmetriske laster skyver formen ut av sin stabile konfigurasjon. Det har blitt funnet flere forskningsartikler som beskriver metoder for hvordan skallform kan og bør utformes når det utsettes for symmetriske laster. Det gjenstår derimot å finne ut av hvordan asymmetriske laster skal håndteres, og hvordan disse påvirker skallet. I denne masteroppgaven ble dermed målet å bygge opp en framgangsmåte for parametrisk design og dimensjonering av et gitterskall i tre som skal være robust nok til å håndtere asymmetriske laster fra snø og vind. Gitterskallet skal være en fjellhytte lokalisert i Indre Fosen Kommune og skal bygges og brukes av studenter ved Norges teknisk-naturvitenskapelige universitet (NTNU). Gjennom de utførte studiene i oppgaven ble det funnet at Eurokoden (EC) gir begrenset informasjon om dimensjonerende laster for skallkonstruksjoner, og at den gitte informasjonen bare kan anvendes til en viss grad. En algoritme for lastkonfigurasjoner ble etablert i Grasshopper, og lastene viste seg å være mer kritiske for gitterskallet enn de gitt av EC. Disse lastkonfigurasjonene ble imidlertid antatt å være lite sannsynlige, og ble dermed ikke benyttet i videre dimensjoneringsprosedyrer. Det ble illustrert hvordan formen påvirker skallets oppførsel, og hvordan dobbeltkrumme former fører til stivere og mer robuste konstruksjoner. Formen på hytten ble optimalisert basert på å minske forskyvninger. Den initielle formen viste seg å respondere betydelig verre på snølasten fra snødriver (asymmetrisk) enn på jevnt fordelt snø (symmetrisk). Den optimaliserte formen viste tydeligere dobbelkurvatur, som førte til en bedre lastbæringsevne og minsket forskjellen på konstruksjonens påvirkning av snølast fra snødriver og jevnt fordelt snø.

Preface

This master thesis was written as a concluding part of our Master of Science Degree at the Norwegian University of Science and Technology (NTNU), department of structural engineering. We have been engaged in the research and writing of this thesis from January to June of 2018.

During our years at NTNU we have been exposed to a lot of structural design exercises and theoretical problems. The tasks has usually involved designing a structure with a predetermined shape. The last semester before initiating our master thesis, we had a course involving parametric design with the software Dynamo. Both of us were quickly fascinated by this type of design method where visual programming was used to create a model that could easily adapt to the change of parameters. It allowed us to participate in the optimisation of the structure during the design process. In the same course we were introduced to form finding methods and grid shell structures. The grid shell caught both of our attention, and we decided to move away from the topic of concrete structures, which was our original plan, to conceptual structural design (ingeniørarkitektur).

We would like to begin with expressing our sincere gratitude to our advisors, Professor Anders Rønnquist and Marcin Luczkowski for helping us formulate our research question and guiding us through this semester. A special thanks to Marcin Luczkowski for responding to our many question at all hours, and Anders Rønnquist for all the inspiring conversations and discussions. We would also like to thank Steinar Hillersøy Dyvik for his assistance with the parametric software and valuable architectural input.

Thank you to Helle Stam Faugstad and Øyvind Sunnvoll Rognes for all the helpful discussions.

Last but not least a great thanks to each other, for motivating and supporting each other when motivation and support was undoubtedly needed.

Marie Eliassen

Marie Eliassen

Åshild Huseby

Åshild Huseby

Introduction

Advances in computer technology, as well as fabrication and construction methods have opened the door for designing structures of complex and extraordinary geometry. Elegant, thin-shell structures can be aesthetically pleasing and can provide more material-efficient designs [1, p.1].

The shape of the shell is of great importance as the load-bearing efficiency derives from it. If the shape deviates from its stable geometry the structural advantages can be challenged. Deformation due to actions on the structure can in worst-case lead to collapse of the building. The aim thus becomes to pursue a design robust enough to withstand all plausible load combinations. The shape of the shell requires careful consideration and the design should be continuously modified until the solution is satisfactory. *Parametric modelling* provides this type of flexible workflow and is hence an efficient method to adopt for the design of shell structures. In addition it enhances the synergy between architects and engineers by enabling structural optimisation parallel with the design process.

A shell structure can either be continuous or reticulated, where the latter is a shell structure divided into a grid of smaller elements. This characteristic inherits its name of a 'grid shell', where the material used is typically timber. The Mannheim Multihalle, the Downland and the Pods Sports Academy grid shell roofs are just a few examples of State of The Art (SoTA) shell structures. A common denominator for the structures mentioned above are that they are built in low lying regions, where the affect of snow loads are less significant. Previous studies have shown that shells are very efficient when it comes to tackling symmetric loads, however challenges appear when the shell is exposed to environmental loads which tend to cause asymmetric pressure.

Norway is a country in Northern Europe where large areas are being seasonally exposed to high snow and wind loads. Accordingly, the structural capacity of buildings (roofs) depends on the ability to handle strong winds and snow accumulations on roofs [2, p.283]. The challenge is how to design a light-weight grid shell structure, so dependent on its shape, in conditions where asymmetric pressure caused by environmental actions possibly can lead to large deformations.

In this thesis we will investigate how the Eurocode deals with snow and wind actions on shell structures. A parametric study of simple geometric shell shapes is carried out to get a deeper understanding of how the shape effects the structural behaviour of the shell. In Chapter 6 a case study will be performed with the intent of designing a grid shell roof for a cabin planned to be built in the mountains of the Indre Fosen region near Trondheim. The cabin will be built as a part of the cabins managed by the sports association of NTNU (NTNUI). The focus will be on establishing a method of design where parametric modelling and optimisation will be central. The structural design is done in close cooperation with the PhD candidates Marcin Luczkowski (structural engineer) and Steinar Hillersøy Dyvik (architect). In addition, we have been working in parallel on this case with two co-students at NTNU; Helle Stam Faugstad and Øyvind Sunnvoll Rognes.

Contents

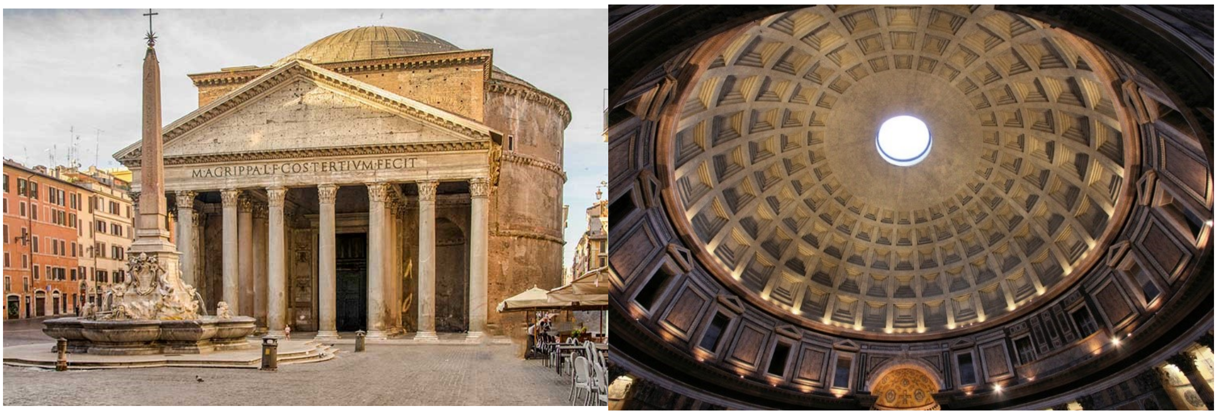
1 | Background

1.1 Architects and engineers

Historical structures

Throughout the history of time, astonishing structures have been raised, and are shown to have had a great impact and influence on the structures of newer age. The greatness of these structures does not only lie within the incredible architecture, but also the complexity of the structure compared to in what age they were built. One of the famous historic buildings, which represents a major step forward in structural engineering is the Pantheon in Rome (see Fig. 1.1). The Pantheon was constructed as early as in the 120s, and it is famous for its characteristic dome which spans about 44 meters [3]. Similar structures such as the Florence Catheral (1434) and Saint Peter's in Rome (1590) have since been constructed, where the dome has shown to be a central eye-catcher seemingly repeated. A common denominator for these structures is that the same person was in charge of the design and construction. In fact it is an anachronism to use the words "engineer" and "architect" about the designers in the ages before the 1450s.

Many of the constructors were artists and mathematicians, using the concepts of geometry and physics to figure out the shape of the structures. Michelangelo for instance, known for his artistry and poetry, worked on the construction of the dome of the St. Peter's Basilica in the 1500s[1, p.36].



(a) The Roman "temple of gods"[4]

(b) The dome of the Pantheon [5]

Figure 1.1: The Pantheon of Rome

Technological era

The roles of architects and engineers have changed since Michelangelo's time, from having one Master Builder in charge of the both the artistic and technical part of the design, to having a clear distinction between the two disciplines [6].

Today, the continuing specialization of their disciplines has caused a *growing gap* in the understanding between architects and engineers. The architect and the engineer, working on the same project, will use different measures and be concerned with different objectives while heading for their goal [7].

Development of *new technology* has made it possible to enhance the synergy between the architect and engineer, permitting the design to become more efficient and complex. An example of this is *parametric design*, which is a flexible tool that allows for effortless changes to the design without deleting and redrawing. It is highly beneficial compared to the traditional CAD-software where the geometry is more time consuming to change. The model is generated in an environment where the geometry is parametrically defined and assigned properties that are either fixed (constrains) or variables (parameters). The designer can modify the parameters, and the model will adjust accordingly [8]. This opportunity of freely alternating between and comparing options allows for a more dynamic design.

In traditional methods of design, the structural engineer would optimise the structure mainly after the design is finalised [7]. This work flow leads to a distinction between the architect and engineer. By combining parametric design and structural analysis, the structural principles can be considered parallel with designing the structure's geometry. This enables the structural engineer to contribute in the design process by performing structural optimisation from an early stage [7, p.626]. As a result, this could advance the constructability and thus reduce time and cost without compromising the structural performance and architectural shape. It must however be mentioned that parametric design opens the door for a better cooperation, but do not ensure it. It is up to the architect and engineer to make sure their priorities are shared and that they have a common understanding for the perfect structure to be realized [1, (Foreword, Shigeru Ban)].

1.2 Beauty of shell structures

The roles of the architect and engineer are weighted differently depending on the size and function of the building. In shell structures, where the shape derives directly from their flow of forces, the design process is highly reliant on the engineer [1]. The shape is what give rise to the distinctive characteristic of shell structures, namely the efficient load-bearing behaviour and lightness. With the use of shells as structural elements large span to thickness ratios can be achieved. Thus, the advantages are sustainable structures where the design is efficient, light-weight and hence also economic. The thin-shell structures will have minimal bending resistance, assigning the responsibility to the designer to create a shape where the shell works primarily in membrane action. *Form findings techniques* are commonly used by engineers and architects to achieve such shapes. These techniques are based upon the search of optimised shapes with regards to satisfying required constraints and objectives. An example is the use of Thrust Network Analysis (TNA) on masonry structures in which finds a shape where the forces within the structure are solely working in compression. A second is the Force Density Method (FDM) which has proven to be an invaluable approach to generate equilibrium solutions. Heinz Isler, a Swiss structural engineer was a master of form finding and famous for his thin concrete shells. Figure 1.2 illustrates one of his many works on thin concrete shell structures where the shape is based on the hanging chain principle (will be further discussed in Sec. 2.4).

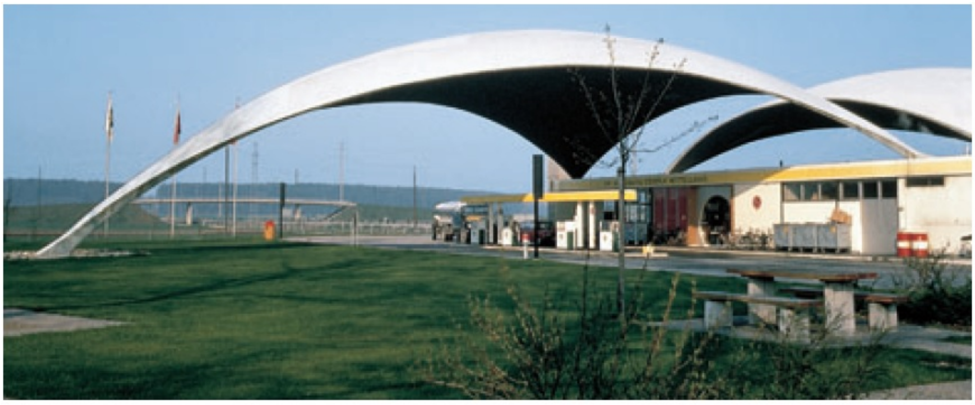


Figure 1.2: BP Service Station, Deitingen, 1968 [9, Fig. 20.11]

1.3 Inspiring shell structures

Mannheim Mulithalle

The timber lattice roof of Mannheim Multihalle is an example of a grid shell many times mentioned in literature of structural engineering. The structure was made as a part of a garden exhibition in Mannheim, Germany in 1975 [10], and was for that reason designed as a temporary structure. The architects of the Mannheim shells wanted the structure to assimilate the surrounding landscape, and its "hilly" form. To reach this unusual shape, they contacted Frei Otto, the famous architect and Professor at Stuttgart University. Frei Otto was known for the use of hanging chain models [10], which used the principle of the oldest form-finding method published by Robert Hooke [1, p.7]. As mentioned in Section 1.2, form-findings methods are important tools used for deriving an efficient shape. The unique shape could hence be realized by deciding on the boundaries of the structure and hanging the chains under gravity-only load (see Fig. 1.3).

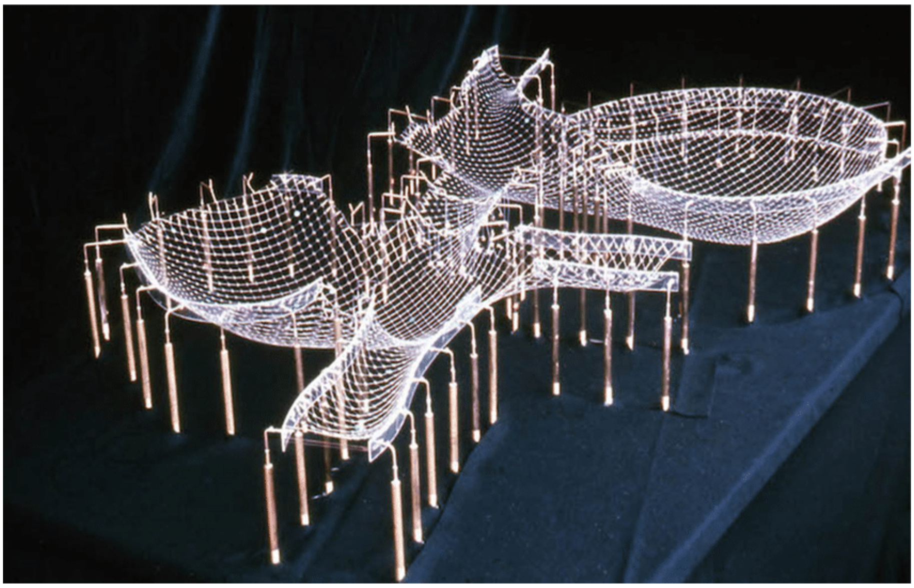


Figure 1.3: The final hanging chain model of Mannheim [11, Fig. 2]

If the self-weight would be the only force acting on the structure, the shell could be very thin, as the internal forces would only be in compression. However, with the presence of external actions the structure was found to need a double-layered lattice grid with shear blocks and diagonal ties to provide the required stiffness to withstand buckling and asymmetrical loading (see Fig. 1.4). The double layers also enabled the flexibility to bend the timber as it was going to be constructed by laying the grid out flat, and raising it by the nodes. The grid shell is hence a kinematic grid shell as it is actively bent into its final position.

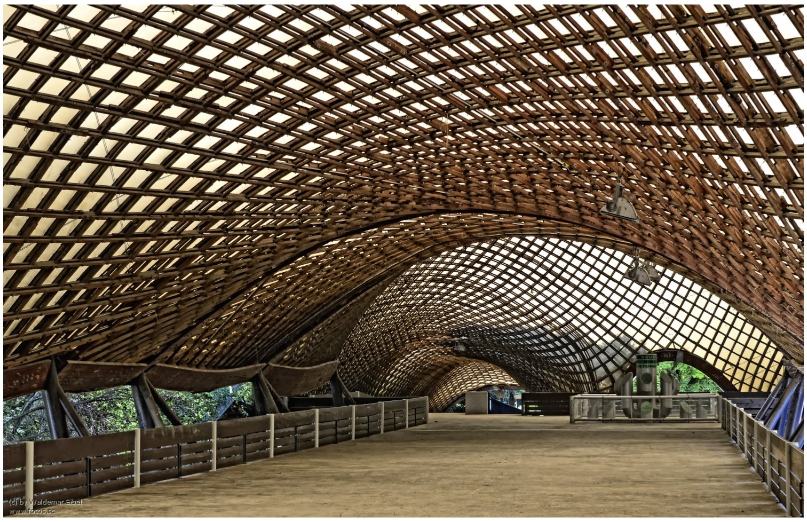


Figure 1.4: Inside the Multihalle in Mannheim. Shell structure with a double-layered lattice grid [9, Fig. 12.1]

There were issues with installing the structure, as they found out that the supports would need to be in their right position when raising the structure. Due to the size of Mannheim, it was nearly impossible to lift the structure in place by cranes and install the shear blocks and diagonal ties, as the original plan was [10]. The solution became to install scaffolding towers at precise intervals and jack them up to raise the structure (see Fig. 1.5). Fork trucks were used to adjust the horizontal placement of the towers. When the towers were lifted to their correct positions, large shape adjustments were necessary as the deflection between the towers were up to $200mm$. This put constraints on how and in what order to start bolting up the double layered lattice. The advantage of using the scaffolding towers was that the lifting process was slow enough to permit adjustments of the shape, and checking the slotted bolt holes.

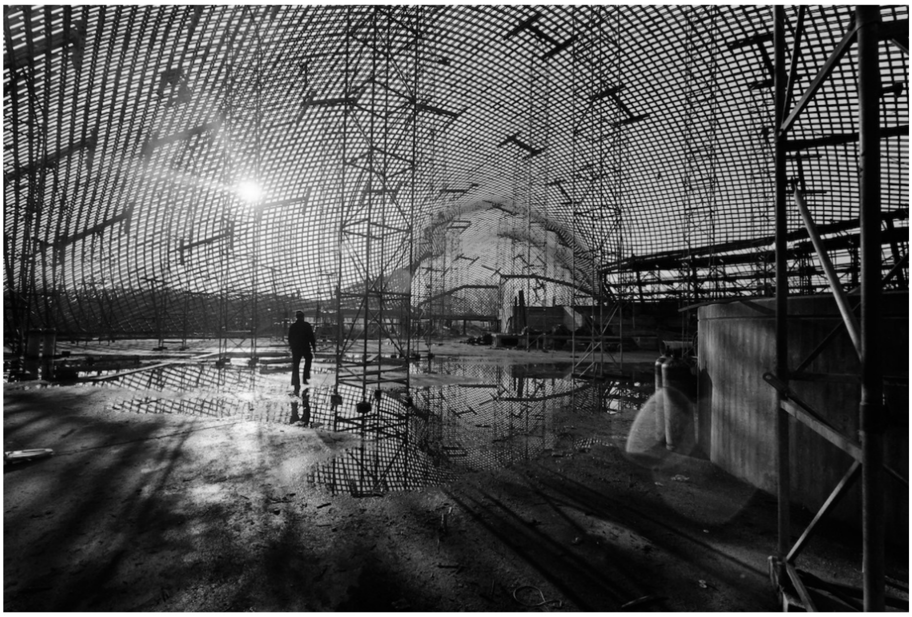


Figure 1.5: Interior shot of the erection of Multihalle in Mannheim [12]

Pods Sports Academy

The Pods Sports Academy, also known as "The Pods", is a state-of-the-art grid shell in Scunthorpe's Central Park in the UK. The sports hall opened in 2011, and the shell roof covers the whole of six badminton-courts, two swimming pools, a gym, a dance studio, and a cafe area [13]. Figure 1.6 shows the finished structure.

The design and concept of the sports hall was completed with an interactive collaboration between the engineer and architect. Where Buro Happold Engineering was the company in charge of the structural design and optimisation of the grid shell, and Andrew Wright Associates were the architects in charge of the conceptual phase [13].

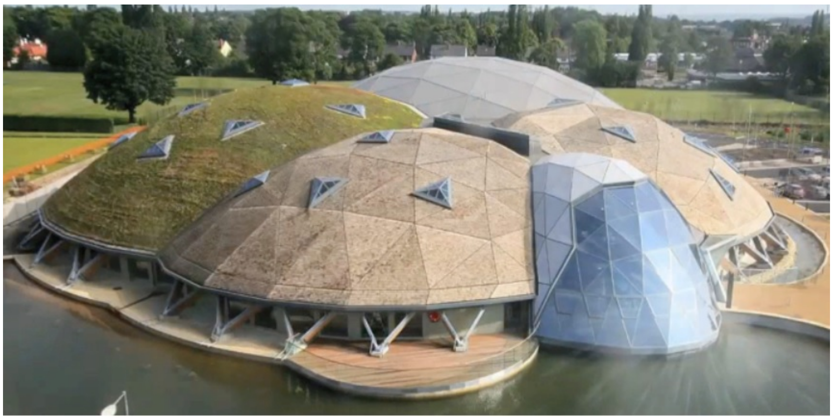


Figure 1.6: Aerial view of the Pods Sports Academy [14, Fig. 27]

The structure is an assembly of five domes or pods, where each of them is optimised for the space it covers. The shape was found by hanging chain models, similar to the exhibition hall in Mannheim, which gave the same results with regards to presence of compression forces only. The Pods also had to increase the design resistance to not only take care of the self-weight, but also consider the effects from wind and snow actions. However contrary to Mannheim's kinematic grid shell, the Pods was a reticulated shell with straight timber beams connected in the nodes (steel joints) (see Fig. 1.7).

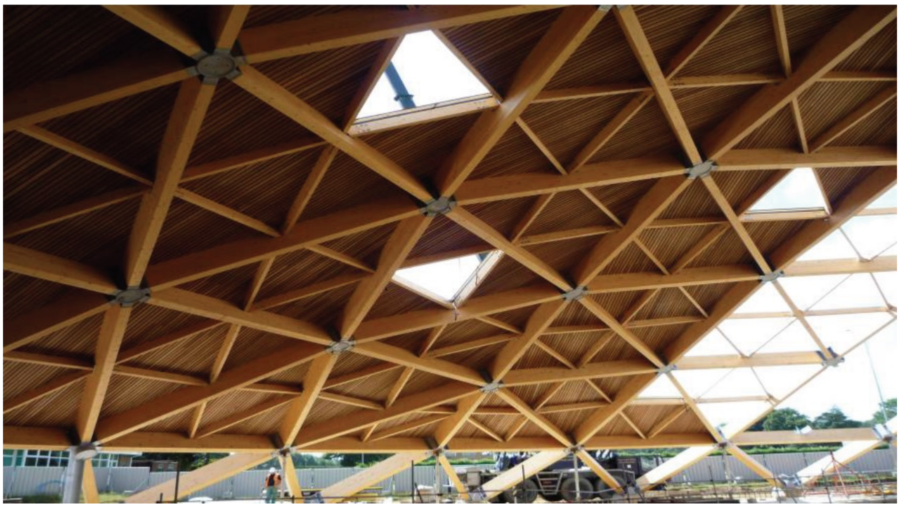


Figure 1.7: Reticulated shell roof of The Pods [14, Fig. 24]

In the design of the grid pattern on the modelled surface, the grid was defined as flat, and then mapped onto the surface. Dynamic relaxation was adopted as the form finding method to even out the network of triangles defining the grid. This made it possible to have standardised beams which could ease manufacture and reduce amount of material [15].

Landesgartenschau Exhibition Hall

The Landesgartenchau Exhibition Hall was conceived through a research project at the University of Stuttgart in 2014. It is a prototype and showcase exhibiting the current developments in computational design and robotic fabrication for lightweight timber construction [16]. The Landesgartenschau Exhibition Hall has displayed inventive and sustainable features both with respect to architecture and engineering. Figure 1.8 shows the structure from the front entrance.

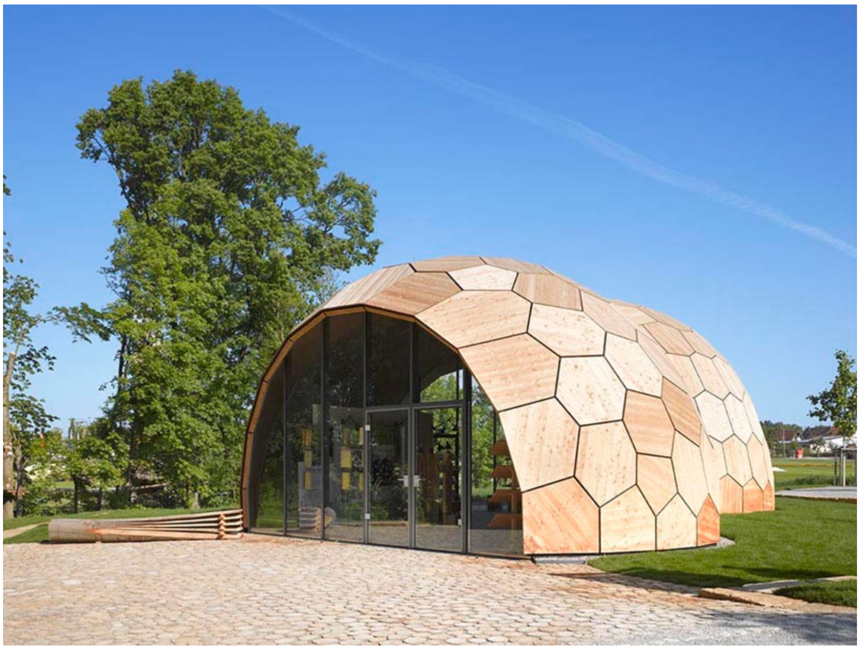


Figure 1.8: The entrance of the Landesgartenschau Exhibition Hall [17]

The structure consists of $50mm$ thick (thin) segmental plates, connected by finger joints to allow for transfer of forces. The structure can hence be seen as the timber-variant of a continuous shell, as the segmental plates ensure that the external loads can be transferred to the ground in all directions. However, the connections between the plates interrupt the continuity of the material and therefore the structural design of the finger joints need specific consideration. This is where the robotic fabrication methods are excellent as they allow for the manufacture of complex geometry.

The Landesgartenchau Exhibition Hall bases its design on plated shell structures found in nature. An example used for this project was the skeleton of the sea urchin. This is known as biomimetics (imitating of models in nature), and was implemented in both the development of the shells and the finger joints. The robotic fabrication allowed for the precision and individual details of the finger joints by being able to replicate the solutions from the digital optimisation process. Figure 1.9 illustrates the sea urchin (left) and the joints between the plates (right) which are looking very similar to finger joints [18].

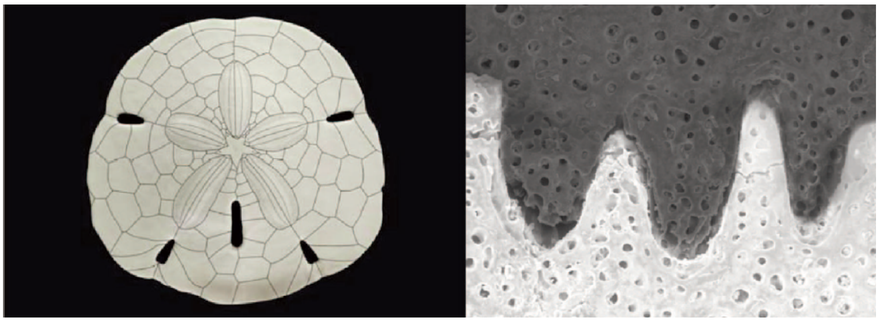
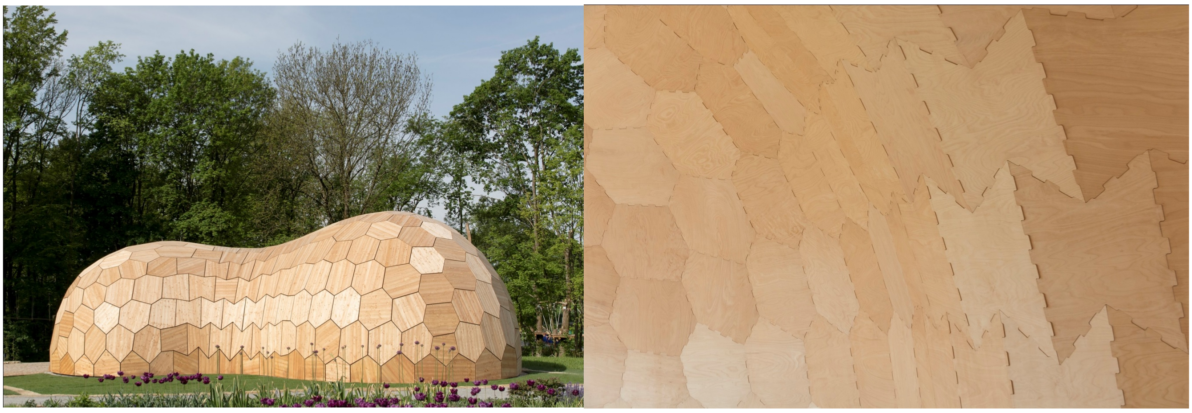


Figure 1.9: The plated shell structure of a sea urchin (left) and the plate edge assimilating finger joints (right) (microscopic view) ([19, Fig. 2])

The architectural vision of the structure was to have a spacious variety dividing the inside space into open - closed - open areas. This led to a division which aesthetically makes up two zones, one entrance space and the main exhibition shape. The shape variation can be seen as having an architectural function by dividing the space, as well as enhancing the structural performance [16]. Both zones have dome-like shapes and build with convex polygonal plates. In the area where the two dome shapes meet, plates that are concave are used. Figure 1.10 shows the plated shell structure (shape of the structure in Fig. 1.10a and the concave and convex plates in Fig. 1.10b).



(a) Dome-like shape of the structure

(b) The concave and the convex plates

Figure 1.10: Segmental Timber Plate Shell for the Landesgartenschau Exhibition Hall [20]

2 | Governing Equations

In this chapter we will dive into the characteristics of shell structures and discover how they achieve their distinct structural behaviour. In addition, the software tools used in this thesis for examining shell structures will be presented.

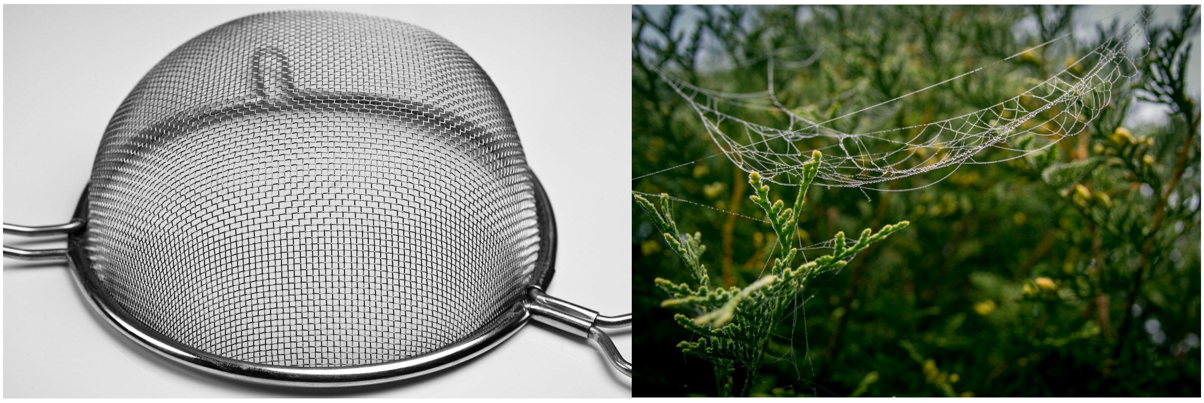
2.1 Characteristics of shell structures

Curved Surface

A shell is a three-dimensional curved surface, where one dimension is smaller compared to the other two [1, pp. 21-31]. It resembles a beam, which is a member defined by a line with a neutral axis and a cross section - a shell can also be defined by its middle surface and a thickness. Both a plate and a shell can be described by a surface, like the beam and the arch can be described by a line [21, pp. 1-2]. However, for both the shell and the arch, the *curvature* of the surface and line is essential. The curvature makes it possible to carry vertical loads with only in-plane forces, which is the primary contribution to the efficiency of the shell (see Sec. 2.3).

Rigidity

A sieve and a spider web can be considered as two different types of shells (see Fig. 2.1). Both are made up of straight elements woven into a grid. The sieve is permanently bent into a curved surface creating a grid shell. The spider web becomes a curved surface with the spider threads acting in tension when the wind blows. However the spider web returns to its original position after the wind gust.



(a) A sieve is form-passive [22]

(b) A spider web is form-active [23]

Figure 2.1: The varying rigidity of the shell

The spider web is a tension structure that adjust its shape under applied load, i.e. it is a form-active system. Contrary, the form-passive sieve appears rigid and will more-or-less stay in the original curved position when exposed to external load. The shape and curvature give rise to the strength and rigidity of the sieve [1, pp.21-22]. Form-passive shells, which are rigid, are primarily of interest from a structural point of view.

Light-weight

Evolution has resulted in development of shells which can function as light-weight shelters protecting the species from outside danger. Examples are the exoskeleton of the turtle and the snail (see Fig. 2.2). The shape and material is carefully developed to achieve an optimal structure.

Figure 2.2: Shells found in nature - the shell of a snail [23]

Drawing a parallel from nature to a structural point of view; if the shell is carefully shaped it allows for achieving a light-weight structure with an efficient use of material.

2.2 Shells - Classification

Geometric Classification

A method of classifying shell structures is to arrange them into groups based on their geometries [1, p.2].

Freeform	The shell is assigned an arbitrary form that is not derived from any geometrical objects and where structural performance is not considered. Optimisation of the predefined freeform shape is important to obtain a stable and efficient structure.
Geometric	The shape of the structure is based on geometrical objects.
Physical	The shape is derived from form findings methods, like hanging chains (see Sec. 2.4), to obtain a structure in a state of equilibrium.

Surface Classification

The Gaussian curvature of the surface is used in the classification of shells. Gaussian Curvature is defined as the product of the two principle curvatures at a given point on a surface [24]. The curvatures are orthogonal to each other. The shell can be divided into three categories with respect to the Gaussian curvature, κ (see Fig. 2.3).

$\kappa > 0$ A dome is an example of a shell surface having a positive Gaussian curvature and the surface can be classifies as synclastic.

$\kappa < 0$ A saddle-like surface has a negative Gaussian curvature and is categorized as anticlastic.

$\kappa = 0$ A single curved surface, like a cylindrical shell has zero Gaussian curvature and is called monoclastic.

All single curved shells with zero Gaussian curvature are said to have a developable surface. The surface can be laid out flat without taring, e.g. paper rolled up into a cylindrical shell [25, pp. 1-4]. Doubly curved surfaces with non-zero Gaussian curvature, on the other hand are said to be undevelopable i.e. cannot be deformed into a plane without being distorted.

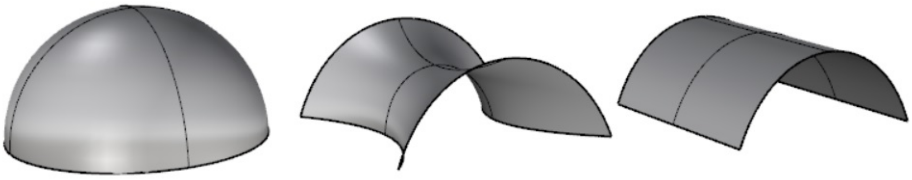


Figure 2.3: From left to right: A synclastic dome, an anticlastic saddle-like surface, a monoclastic cylindrical surface

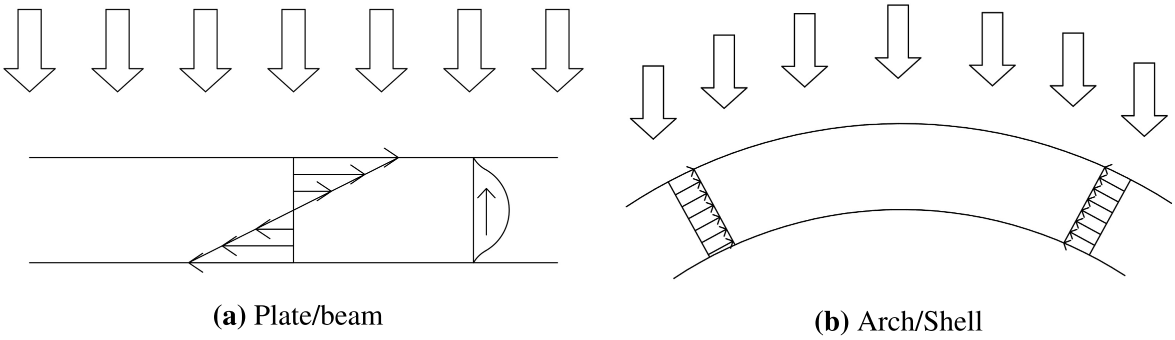
Different types of shell structures

The shell roof can be constructed both as a continuous surface with a thickness, and as a grid of single elements following the surface, a so-called grid shell [1, p.2]. Deitingen Süd Service Station is an example of a continuous concrete shell, built in 1968 (see Fig. 1.2).

The grid shell can be formed by short segments of straight beam elements connected into nodes, or by continuous bending members. The latter is a so-called a kinematic grid shell because of its kinematic construction method where the continuous members are actively bent into shape from an initial flat surface. Hence, timber is the only suitable material which due to its flexibility, can handle the large deformations without failing (yielding) [24]. The Mannheim Multihalle is an example of a kinematic grid shell (see Fig. 1.4. The Pods Sports Academy is an example of a reticulated shell formed by straight beam elements (see Fig. 1.7).

2.3 Membrane forces

The curvature of the shell is what makes it a very effective load bearing structure. Figure 2.4 illustrates how curved shapes (Fig. 2.4b) can carry transverse loads differently from plane shapes (Fig. 2.4a). The plate transfers the load to the supports with bending and shear action, opposed to the shell which mainly transfers the loads through membrane action [26, p. 499].


(a) Plate/beam

(b) Arch/Shell

Figure 2.4: Load bearing by bending and in-plane (membrane) action. Based on Fig. 15.1 in [26]

The efficiency of the shell depends on the ratio between the loads carried by normal action versus bending and shear action. Wilhelm Flügge in [21] explained this concept with a simple comparison of a rolled up paper, identified as a cylindrical shell and an egg shell. If a light lateral force is applied by two fingers pressing on the paper that is rolled and taped together, it will deform easily. The cylindrical shell will resist the force mainly with bending moments. The egg, which in it self is a thin shell made of a fragile material, will withstand a surprisingly high force without breaking or deforming visually. Unlike the cylindrical paper shell, the egg will primarily carry the loads with in-plane shear and normal forces [21, p.8]. When the deformation and the thickness of the shell is small, it is demonstrated that the bending and twisting moments can be neglected in a stress analysis. This theory is called *The Membrane Theory*, however it will not be discussed further in this thesis.

The example with the paper and egg above illustrates how the in-plane stresses (membrane stresses) are more efficient. This behaviour can be explained further by simple mechanics. A beam, with a rectangular cross section, exposed to a bending moment will have a linear stress distribution. The same beam exposed to a compression force will have a uniform distribution of stresses. Let us assume the maximum stress value in both cases is σ_0 (see Figure 2.5).

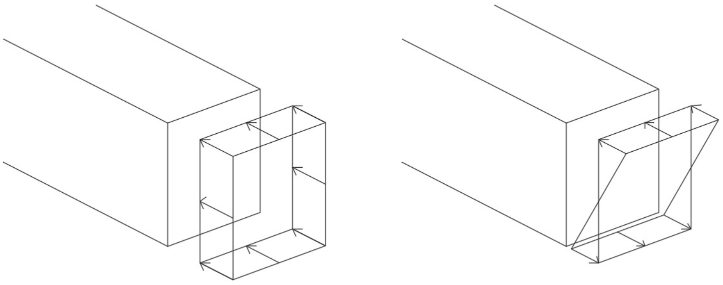


Figure 2.5: Illustration of normal stresses in a beam exposed to a compression force (left) and a bending moment (right)

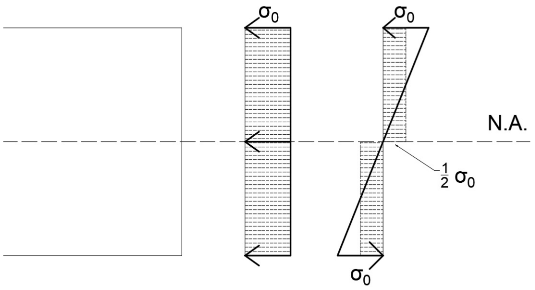


Figure 2.6: Horizontal view of the uniform stress (compression) and linear stress distribution (bending) in a beam.

Figure 2.6 shows how the average stress in the bending situation is half compared to the compression situation. The beam will in other words not be able to take full advantage of its cross sectional area when it is experiencing bending. As a result, it is easier to bend a structural element than stretch it, i.e. change its length. Thus, when constructing a shell structure, engineers pursue to find the shape and the supports leading to an efficient structure working predominantly by membrane action [1, p.23].

2.4 Funicular shape

Robert Hooke wrote a famous anagram in 1675 stating:

> *as hangs the flexible line, so but inverted will stand the rigid arch* [27].

The shape of a chain hanging in tension, corresponds to its equilibrium shape. Due to the chain being a form-active system, it will adjust its shape to different load situations. Thus, the shape of the chains and the path of the resulting axial forces inside will coincide. A chain hanging under its own self weight will be shaped as a catenary curve. If the chain is rigidified and inverted it will form the line of trust of the arch (see Fig. 2.7). This line is a theoretical path of compression forces in the arch, representing static equilibrium. The catenary shaped arc with only self-weight applied will thus carry all the loads in compression, with no occurring bending moments.

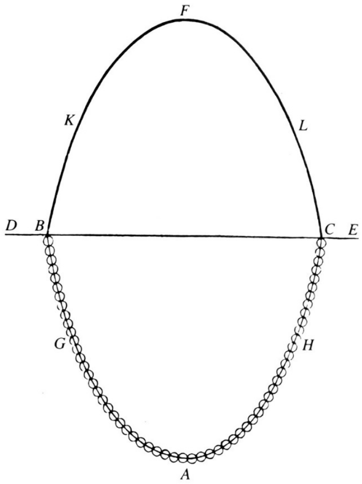


Figure 2.7: Hooke's hanging chain, illustrated by Poleni (1748) [28]

The hanging chain principle may also be used for shells. By loading a hanging membrane, one can find the form of the shell for which the shell only will work in compression. As previously mentioned in Section 1.3, Mannheim Multihalle and Pods Sports Academy are both examples of structures where the form was found using Hooke's hanging chain principle. If an arch or a shell can carry a load without producing bending moment, the shape is *funicular* for that load situation. However, unlike the chain and the membrane, the arch and the shell are form-passive systems. If the load situation changes or the form deviates from the ideal equilibrium shape, additional bending moments arises.

Deviations from the optimal shape will in the real world always exist, either it is from material defects, imperfections due to on-site construction or a combination of them. The deviations cause the structure to be less capable of carrying loads, and it may result in failure. To design a safe shell, it needs to be able to cope with a certain grade of imperfections, as well as load variations.

2.5 Structural behaviour and constraints

Boundary conditions and inextensional deformation

A closed shell is more rigid than an open shell. If you have a closed box, for example a paper coffee cup with a lid; it is more difficult to deform it when the lid is on, than when the lid is off. If you take the lid off, the cup will though still try to resist deformation and not collapse when you lift it to drink. This is partly due to the edge of the cup often being thicker than the rest of the cup. This strengthening ring give the cup more rigidity, even though it is now equivalent with an open shell.

This shows how an open shell can appear rigid if the boundary conditions are adequate and the shape is right. However, if either the boundary conditions or the shape is insufficient, a mechanism may develop and the shell can experience *inextensional deformation*. This means that deformation of the shell can happen without any strain, only with bending. Due to no stretching and changes of length on the surface of the shell, the Gaussian curvature remains unchanged [1, p. 25] .

As previously explained in Section 2.3, it is easier to bend than to stretch a structural element. Hence, structures that can deform inextensionally are more flexible and less efficient. An example of such a structure is the Mannheim grid shell which was not triangulated and could therefore have inextensional modes. The presence of the diagonal cable bracing would only partly contribute in resisting this mode. It is important to have adequate bending stiffness to resist inextensional deformation.

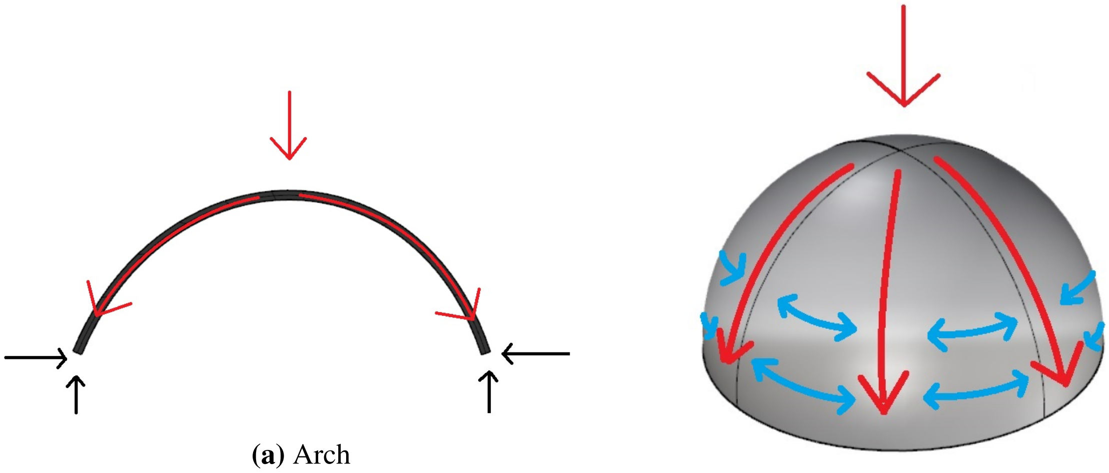
Buckling

Another unwanted effect crucial to take into account in the design of shell structures, is buckling. If a shell has sufficient boundary conditions and cannot undergo inextensional deformation, they tend to be very stiff and efficient. However, these shells are also very sensitive to imperfections. As the shell deflects under loading, the shape becomes less efficient at carrying the load and more prone to instability in the form of buckling. In an efficient shell a minor imperfection or deformation from the ideal shape can lead to a sudden collapse of the structure. The collapse can occur with minimal warning, and without material breaking and yielding. The less efficient the shell is the less sensitive it is to imperfections. The Mannheim Multihalle grid shell which can deform inextensially and thus is considered as a more flexible and less efficient grid shell, is also less sensitive to imperfection [1, pp. 241-243].

Kicking forces

The barrel vault is an extended arch that spans an open space, thus calculating the forces in an arch and a barrel vault are basically the same. If you apply a load onto the arch, the flow of forces is going down trough the arch and pushing outward at the base. These forces, which are directed horizontally outwards, are from here on called *kicking forces* (see Fig. 2.8a). The supports need to be able to withstand these kicking forces, to keep the arch from not collapsing.

A dome is the configuration of an arch rotated around its own center axis. In other words, both the dome and the barrel vault can be formed on the basis of an arch. They differ, however, of course in their double and single curvature (see 2.2 under *Surface classification*) but also with respect to how they are supported. The dome is a closed surface supported on a ring at the bottom. If this ring can handle tension forces, the ring can support the forces that would be directed outwards in the arch (see Fig. 2.8b. Therefore, the dome is less dependent on the external supports. Figure 2.8 illustrates this, and shows the arch (similar to the barrel vault) and the dome and their force flow due the external load.



(a) Arch

(b) Dome

Figure 2.8: Illustration of the flow of forces in a dome and an arch/barrel vault. The dome is less dependent on the external supports.

2.6 Software tools

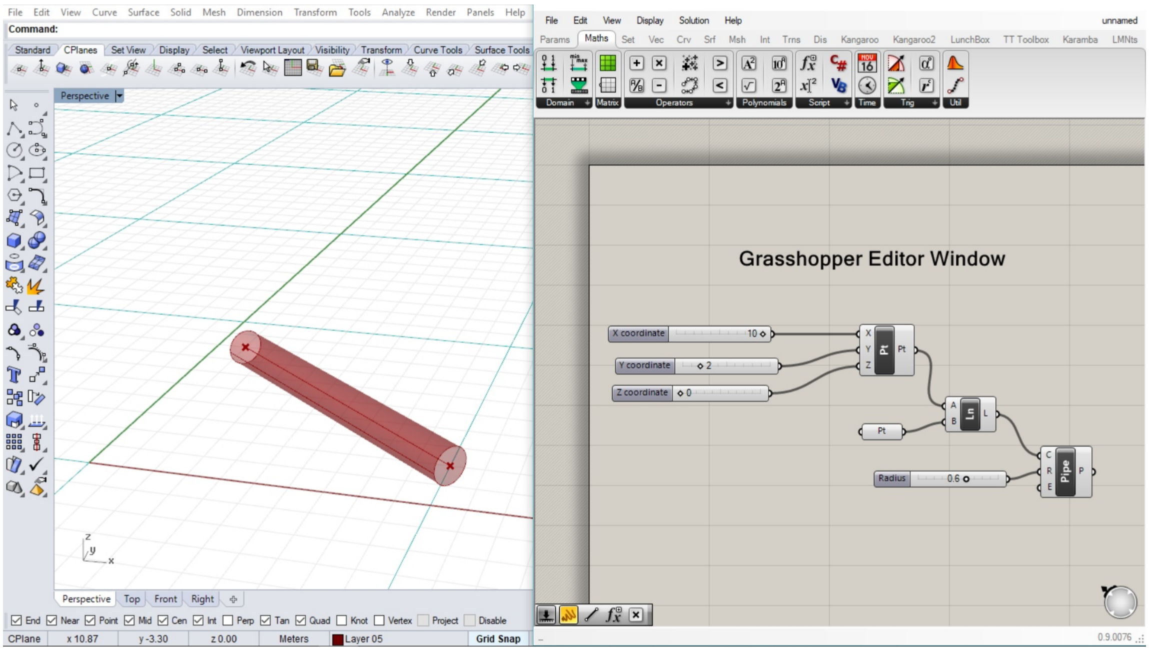
In this thesis the following software will be adopted for geometric design of shell structures in addition to performance of structural analysis (FEA) and optimisation.

Rhinoceros 5.0

Rhinoceros, or Rhino, is a 3D modelling tool - a CAD (computer aided design)-software developed by Robert McNeel & Associates. It provides an interface for defining complex geometry in three dimensions. Rhino provides a visual studio for the programs made in Grasshopper.

Grasshopper

Grasshopper (GH) is a graphical algorithm editor also developed by Robert McNeel & Associates ([29]), and it is a plug-in for Rhinoceros 5.0. The geometry in the program can either be attained from Rhino and controlled parametrically in GH, or defined directly in GH (see Fig. 2.9). GH is integrated within a parametric environment that uses a visual programming language. It uses functional blocks, called "components", which can receive input by connecting them together by "wires" into a sequence of actions. The algorithmic and visual way of scripting the geometry makes it more intuitive to use compared to other programming languages.


Grasshopper Editor Window
Perspective
Pt
Ln
Pipe
Params Maths Set Vec Crv Srf Msh Int Trns Dis Kangaroo Kangaroo2 LunchBox TT Toolbox Karamba LMNts
Domain Matrix Operators Polynomials Script Time Trig Util
Perspective Top Front Right
CPlane x 10.87 y -3.30 z 0.00 Meters Layer 05 Grid Snap

X coordinate 10

Y coordinate 2

Z coordinate 0

Radius 0.6

Figure 2.9: Grasshopper window (right) illustrates how geometry can be defined by components connected by wires. The geometry becomes visible in the Rhino view port (left)

The geometry defined in GH can be easily adjusted by "sliders" connected to the components (parametric geometry). If the slider is adjusted the component will change accordingly, and the rest of the connected geometry will adapt to the new configuration. In the Rhino view port the geometry and the updates can be observed continuously.

Karamba 3D (FEM-software)

Karamba is a structural engineering software embedded in the parametric environment of Grasshopper. There is no transfer between software in order to perform the structural analysis, as the Finite Element-analysis (FEA) can be completed within the same environment as the parameterised geometric model in GH [30]. The model can hence be given structural properties directly by adding the components available in Karamba. Another positive feature is that Karamba is user friendly, by having a similar algorithmic structure and build-up as GH, and also providing components with a clear selection-menu.

Finite Element Method (FEM) divides a system with a complex behaviour into several subsystems where the behaviour is "known". After assembling the system and applying boundary conditions, the result is a system of linear algebraic equations [26]. In this thesis, Karamba will be adopted for evaluating the structural performance of timber grid shells and continuous concrete shells. The finite elements (FE) used to assemble the idealised configurations are different for the two models.

Shell elements in Karamba

To generate the model of a shell structure in Karamba, the surface is subdivided into a finite element mesh. The resolution of the mesh affects the accuracy of the results, but also the computation time of the FEA. The goal is to obtain a solution within a prescribed accuracy at a minimal computational cost.

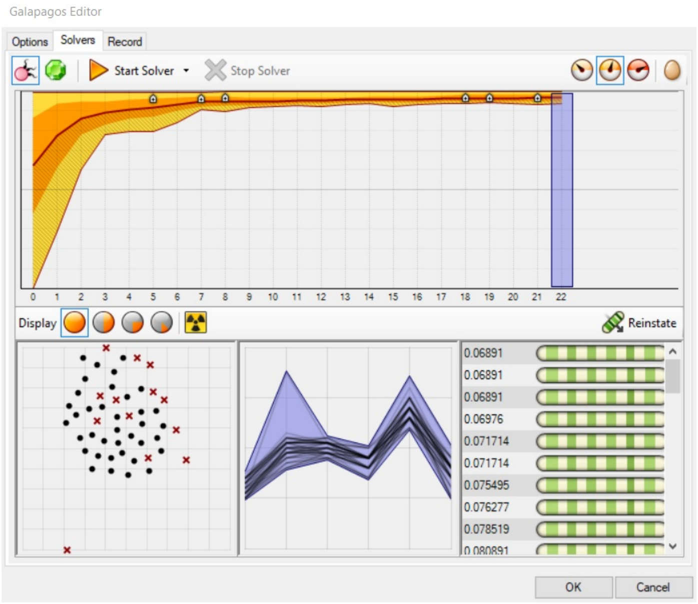
Karamba (Clemens Preisinger, Karamba developer) explained in a Grasshopper3D forum that the shell elements are triangular elements with 6 dofs per node based on Kirchhoff theory. No in-plane rotational stiffness is added [31]. The Kirchhoff theory for thin plates neglects transverse shear deformation, that is $\gamma_{yz} = \gamma_{zx} = 0$ [26, p.461]. If the plate gets too thick this assumption is not valid anymore. Thus, for the thin plate theory to be valid, the following guideline of the thickness/length-ratio should be followed: $t/L \leq 1/10$. The shells investigated in this thesis will have values of t/L smaller than this ratio.

Beam elements in Karamba

The grid shell model is an assembly of beam elements. The type of beam elements used in Karamba is not explicitly defined by the Karamba Manual. However in a forum on the GH web-page, Karamba (Clemens Preisinger) explained how the shear deformation is accounted for, which suggest that Timoshenko beam elements are used [32]. Unlike the Euler-Bernoulli beam theory the Timoshenko theory include shear deformation.

Galapagos Evolutionary solver

Galapagos is an evolutionary solver inside GH that apply genetic algorithms to optimise solutions of problems [33]. The Galapagos-component takes variables, so-called *genes*, as inputs in addition to a *fitness*-function. The gene are lists of sliders (gene-pool) that the Galapagos-solver are allowed to change and create several *Genomes* in order to maximise or minimisee the outcome of the fitness-function. A *Genome* is a the specific values for each of the genes. First the solver populate the fitness-landscape, with random genomes and evaluate the fitness for them all. This is the first generation of solutions. The Genomes with the worst fitness gets killed and the remainder is used to create a new generation. Thus, this second population is not completely random anymore. It continues to kill and create new genomes until an acceptable answer is reached.



0.06891
0.06891
0.06891
0.06976
0.071714
0.071714
0.075495
0.076277
0.078519
0.080891

Figure 2.10: Galapagos Evolutionary solver Editor window

There are pros and cons with the use of evolutionary algorithms. The advantages is flexibility and that they can tackle a wide range of problems. The algorithms are also "forgiving", meaning that they try to solve problems even if they have been poorly formulated. Another advantage is that the run-time process is transparent and it is possible to view the answers during the process. Cons being that evolutionary algorithms are slow and that a single process may run for several days before getting a solution. Secondly, it is not guaranteed that the evolutionary algorithms will find a solution, or recognise the solution when having reached it. If the user does not provide an interval for where an acceptable solution will be the process can run indefinitely.

The combination of Rhino, Grasshopper, Karamba and Galapagos forms an important tool for architects and engineers in the early design phase. Parameterised geometric models in GH can be combined with FE calculations in Karamba and optimisation algorithms.

3 | Actions according to Eurocode

In the world of civil engineering, Standards are commonly implemented to ensure that structural design is being executed on common grounds. In Norway, Eurocode is the Standard frequently adopted for the design of structures both regarding materials and loads. To examine how the Eurocode covers the design of wind and snow actions on shell roofs, three simple cases will be studied (see Fig. 3.1).

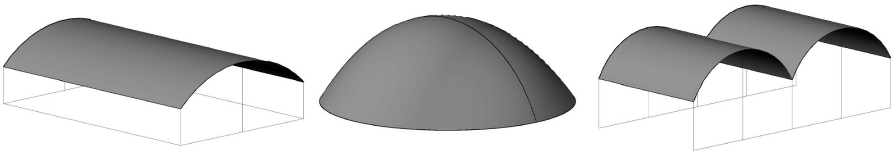


Figure 3.1: Geometric shapes to be investigated in *Actions according to Eurocode*. From left to right: Cylindrical, spherical and multi-span cylindrical roof.

The design actions on roofs depend on the location of the structure as well as the shape of the roof. The location chosen is in accordance with the NTNUI cabin which will be the main object of the Case Study in Chapter 6. The location in question is north of Trondheim, across the fjords in Indre Fosen Kommune. See Figure 3.2 for the map location and Table 3.1 for more exact information. Notice that Indre Fosen Kommune is a relatively new municipal established by the merging of Rissa Kommune and Leksvik Kommune. The National Annex still operates with the old names, thus information on Leksvik kommune is accounted for.

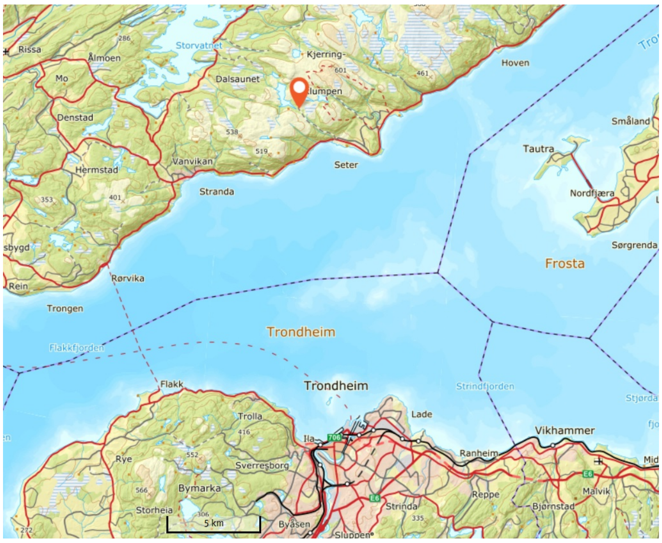


Figure 3.2: Location of the shell structures addressed. Chosen in accordance with the NTNUI cabin which will be designed in Chap. 6 [34]

Table 3.1: Information about the location of the shell structures studied: Basis for the design of actions.

Coordinates	63.5898110, 10.3258360
Kommune	Indre Fosen previously known as Leksvik
Name of area	Mevassetran
Height above sea level	361 m

3.1 Snow calculations according to NS-EN 1991-1-3

Snow calculations are done in accordance with NS-EN 1991-1-3 ([35]), hence all Eurocode references in Section 3.1 will refer to to this part unless specified otherwise.

NS-EN 1990 Section 4.1.1(2) states that snow actions may be considered as either variable actions (Q) or accidental actions (A). The National Annex (NA.2) of NS-EN 1991-1-3 indicates that for Norway, snow actions should not be considered as accidental nor as exceptional. Snow actions are thus regarded as variable actions from here on.

NA.4.1(901) gives the characteristic values on ground for Leksvik kommune:

$$s_k = 4.0kN/m^2.$$

Given the structure is situated at a height approximately $200meters$ above the reference height of Leksvik kommune, the characteristic value of snow load on ground increases (see detailed calculations in Appendix A *Calculations of actions according to Eurocode*):

$$s_k = 4.0 + 3 * 1.0 = 7.0kN/m^2.$$

Snow actions on roofs

According to NA.5.2(2) the expression for the accumulated snow on roofs (given variable action and no exceptional snow load) becomes:

$$s = \mu * C_e * C_t * s_k, \tag{3.1}$$

where

μ is the shape factor and accounts for how much snow will accumulate on the roof depending on its shape. [Section 5.3].

C_e is the exposure coefficient taking into account the surrounding terrain of the cabin. Assume normal topography $C_e = 1.0$ ($C_e = 1.0$ for cylindrical roofs [NA.5.2(7)].

C_t is the thermal coefficient for reduced snow actions on roofs due to heat loss. Assume no reduction $C_t = 1.0$

In all of the following sections below (Sec. 3.1.1-3.1.3), Expression (3.1) will be considered with the shape coefficient $\mu = \mu_i$ as the only variable, where $i = 1, 2$ or 3 depending on the shape of the roof. According to Eurocode μ_1 applies for pitched roofs, μ_2 for accumulation of snow in valleys of multi-span roofs, and μ_3 for cylindrical roof shapes. The shape factors (μ_i) regulate how much snow will distribute on the roof compared to the estimated snow loads on ground, s_k.

Where Eurocode falls short in explaining the shape of the roof, this will be highlighted and assumptions will be made in finding the applicable shape coefficient.

3.1.1 Snow distribution on cylindrical roof shape

The first case is a cylindrical roof shape like shown in Figure 3.3, with the shape continuing along its length (the y-direction into the plane).

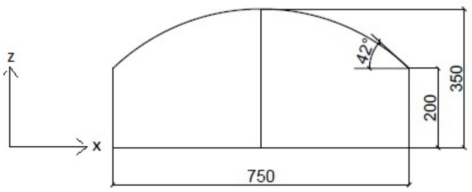


Figure 3.3: Section drawing of the cylindrical roof shape with dimensions $[cm]$.

Section 5.3.5 *Cylindrical Roofs* in Eurocode provides information on the shape factor for a generic cylindrical roof as shown in Figure 3.4, both for undrifted (Case(i)) and drifted (Case(ii)) load configurations.

5.3.5 (1) specifies that $\mu_3 = 0$ for roof curvature $\beta > 60°$, i.e. no snow will accumulate due to the high curvature. This is illustrated as the load width l_s in Figure 3.4, where $l_s \leq b$.

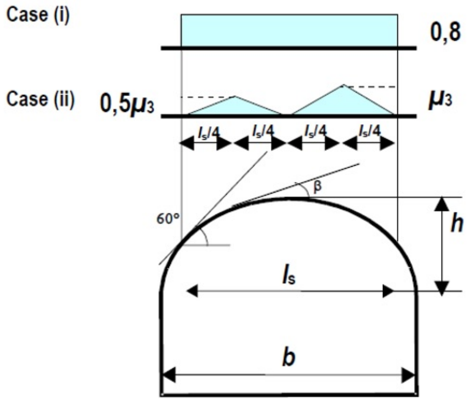


Figure 3.4: Fig. 5.6 in [35]: Snow load shape coefficients for cylindrical roof

The undrifted configuration, Case (i), gives a uniformly distributed snow load with a shape coefficient equal to 0.8 (see Figure 3.4).

For Case (ii), the drifted load case with $\beta \leq 60°$, μ_3 is obtained using Figure 3.5 with the following value for h/b (see Fig. 3.4 for definition of h and b):

$$h/b = 1.5/7.5 = 0.2$$

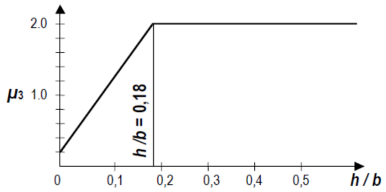


Figure 3.5: Fig. 5.5 in [35]: Shape factor μ_3 for different height to span(h/b) and angle $\beta \leq 60$

Figure 3.5 specifies an upper limit of the shape factor μ_3 as 2.0. As a result the shape factor μ_3 will be equal to 2.0 for the cylindrical roof shape.

Expression (3.1) is then used to obtain the snow actions:

- Undrifted (Case (i)):

$$s(0.8) = 0.8 * 1.0 * 1.0 * 7.0kN/m^2 = 5.6kN/m^2$$

- Drifted (Case (ii):

$$s(\mu_3) = 2.0 * 1.0 * 1.0 * 7.0kN/m^2 = 14.0kN/m^2$$

$$s(0.5\mu_3) = 0.5 * (2.0 * 1.0 * 1.0 * 7.0kN/m^2) = 7.0kN/m^2$$

3.1.2 Snow distribution on spherical roof shape

The second case is a spherical roof shape like shown in Figure 3.6, also called a Dome. Eurocode does not provide information about snow load distributions on spherical roofs or domes.

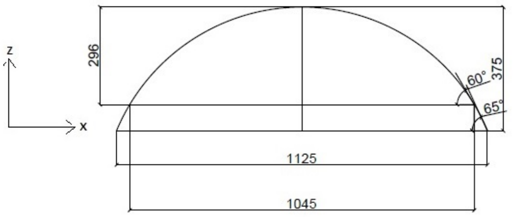


Figure 3.6: Section drawing of the spherical roof shape with dimensions $[cm]$.

It could though be reasonable to assume that the load arrangement could be deduced from the cylindrical roof shape (see Fig. 3.4). A vertical cut through the sphere in Figure 3.6, will in fact be a cylinder with a varying width $b(y)$ and load width $l_s(y)$. The only difference being that the cylindrical shape of the given vertical section has no vertical walls. Given that the only information suitable for the spherical roof is the cylindrical shape with vertical walls provided by EC, Figure 3.4 is still adopted.

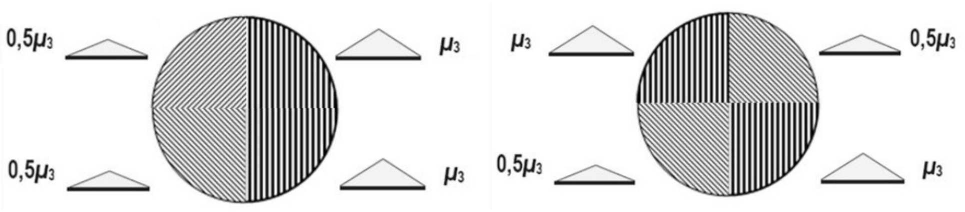
Figure 3.4 is used to find the distribution of snow for the undrifted and drifted load case, and the value of the shape coefficient μ_3 is given in Figure 3.5. The following value for h/b is inserted into Figure 3.4 to obtain μ_3, where h is the height of the curved area and b is the diameter of the spherical shape:

$$h/b = 3.75/11.25 = 0.33$$

The shape coefficient μ_3 is found to be equal to the upper limit 2.0 similar as for the cylindrical roof shape. Hence the design actions presented for the in Section 3.1.1 above for the cylindrical roof will also account for the spherical roof.

Snow distribution over the surface of the spherical shape

It must though be stressed that the shape coefficient found for this case is an approximation based solely on the similarities between the vertical section (xz-plane) of the sphere and the cylinder. The shape coefficient's distribution over the sphere surface is questionable. It could be a fair assumption that the snow would distribute symmetric or asymmetric (resembling a chessboard pattern) into the plane like shown in Figure 3.7.



(a) Symmetric load distribution **(b)** Asymmetric load distribution

Figure 3.7: Two variants of how the snow could be assumed to be distributed on the spherical roof shape

3.1.3 Snow distribution on multi-span roof with cylindrical shape

The last case represents a multi-span roof consisting of a pair of cylinders connected to each other (see Figure 3.8).

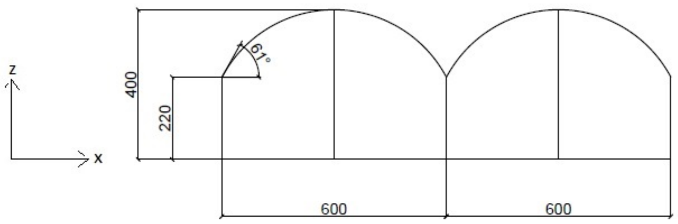


Figure 3.8: Section drawing of multi-span roof with cylindrical shape with dimensions $[cm]$.

Similar to the spherical roof, there is no section in Eurocode which provides information on this roof shape. However, to determine the snow action on the multi-span roof, it is assumed that super positioning of Sections 5.3.5 *Cylindrical roofs* (Figure 3.4) and 5.3.4 *Multi-span roofs* in [35] (see Figure 3.9) will give reasonable results.

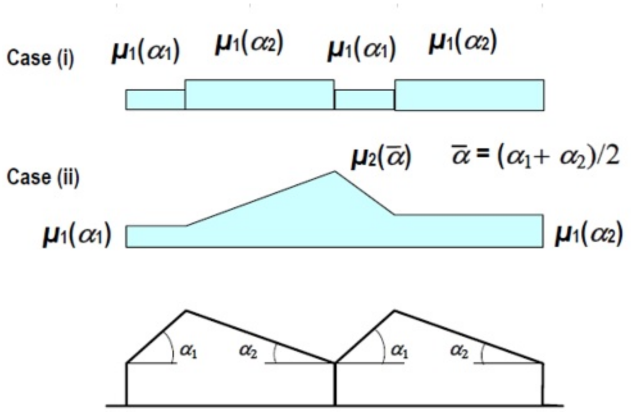


Figure 3.9: Fig. 5.4 in [35]: Snow load shape coefficient for multi-span roof for straight roof elements (pitched roofs)

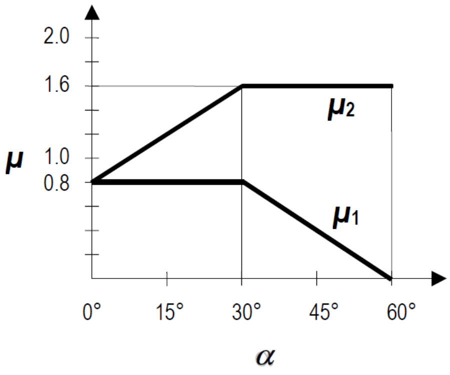


Figure 3.10: Fig. 5.1 in [35]: Snow load shape coefficients, μ_1 and μ_2

The multi-span roof illustrated in Figure 3.9 with the shape coefficient μ_2 found from Figure 3.10, is used as a basis and applied to the multi-span cylindrical roof. The deduced model is illustrated in Figure 3.11, and the shape of the accumulated snow is simplified into triangles.

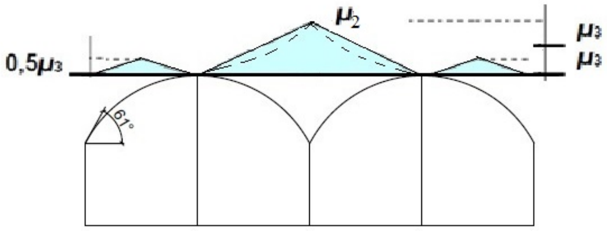


Figure 3.11: Assumed shape coefficients/snow distribution on the multi-span cylindrical roof with the assuption of superposition of both Case(ii) from Fig. 3.9 and Fig. 3.4

Figure 3.10 indicates that the maximum value of μ_2 (accumulation of snow in the valley) is 1.6, i.e. twice the maximum value of μ_1 when the angle is $\alpha = 30°$.

For $\alpha \geq 30°$ the relationship between μ_1 ans μ_2 even increases. For the multi-span cylindrical roof the angle α is assumed to be approximately 60°, meaning the accumulated snow in the valley should be $\mu_2 \geq 2 * \mu_3 = 4.0$ (illustrated in Fig. 3.11).

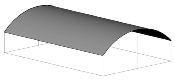
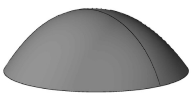
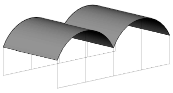
The snow load in the valley would therefore become 4 times higher then the snow load on ground, which with expression (3.1) gives:

$$s(\mu_2) = (2 * \mu_3) * C_t * C_e * s_k = (2 * 2.0) * 1.0 * 1.0 * 7.0 = 28.0kN/m^2.$$

The resulting snow load for the cylindrical multi-span roof is thus quite high compared to the mono-span roofs, questioning the reliability of adopting super positioning of the cylindrical and multi-span roof from EC (Fig. 3.4 and 3.9). It needs to be considered if an upper limit of the accumulated shape factor should be determined.

3.1.4 Summary of snow calculations

Table 3.2: Summary of snow actions calculated for the cylindrical, spherical and multi-span roof shapes

Shell Shapes	EC Figures	*Load case*	Snow loads	$[kN/m^2]$
	*Directly used:*	*Drifted*	$s_3(\mu_3)$	14
	Fig 3.4		$s_3(0.4\mu_3)$	7
	Fig 3.5(μ_3)	*Undrifted*	$s(0.8)$	5.6
	*Approximated:*	*Drifted*	$s_3(\mu_3)$	14
	Fig 3.4		$s_3(0.5\mu_3)$	7
	Fig 3.5(μ_3)	*Undrifted*	$s(0.8)$	5.6
	*Approx.*	*Drifted*	$s_3(\mu_3)$	14
	superpos:		$s_3(0.5\mu_3)$	7
	Fig 3.4	*Undrifted*	$s(0.8)$	5.6
	Fig 3.9			
	Fig 3.10(μ_2)	*Accum. valley*	$s_2(\mu_2)$	28
	Fig 3.5(μ_2)			

In Table 3.2 the snow loads found for the three shapes are presented. The values are equal for the drifted and undrifted load case, with the assumption that Figure 3.4 can be adopted directly or with an approximation.

The ability of adopting EC for snow actions on the shell shapes investigated are seen as the following:

- Cylindrical shape - EC can be used directly
- Spherical shape - EC can be used to draw parallels with the cylindrical shape, but not clear how it will distribute into the plane. Also EC does not clarify if the cylindrical shapes accounts for the shapes both with and without vertical walls.
- Multi-span cylindrical - The shape is similar to the cylindrical mono-roof. However an upper limit for the accumulation in the valley is missing, questioning the reliability of adopting superposition of Figures 3.4 and 3.9.

3.2 Wind calculations according to NS-EN 1991-1-4

Wind calculations are done in accordance with NS-EN 1991-1-4 ([36]), hence all Eurocode references will refer to to this part unless specified otherwise.

Wind action calculations depend on the location of the structure as well as the shape and size of the structure. National Annex and Annex A provide necessary information regarding the location, type of terrain and the meteorological data. Assumptions are done with regards to the topography, climate and terrain roughness of the area in order to determine the required coefficients and values. First the wind pressure is calculated, before the geometry of the three shapes and the wind distribution on each of them are considered.

The calculations of the wind pressure can be found in Appendix A *Calculations of actions according to Eurocode*. The resulting peak velocity pressure is found to be:

$$q_p(z) = 694N/m^2$$

The pressure $q_p(z)$ is then multiplied with the pressure coefficient for external pressure(c_{pe}) to obtain the wind pressure on surfaces [5.2(1) ex. 5.1].

$$w_e = q_p(z_e) * c_{pe} \tag{3.2}$$

Negative sign on wind pressure means suction, and the wind action will have a direction perpendicular to the roof surface. The coefficient, c_{pe} for external and c_{pi} for internal pressure, takes into account the aerodynamic properties for the structure [7.1(1)]. For buildings both internal and external pressure should be considered. However for these calculation, only external pressure and thus only the external pressure coefficient is determined for the roof surface.

The external pressure coefficients $c_{pe,1}$ and $c_{pe,10}$ are given for loaded areas A of $1m^2$ and $10m^2$, respectively. For buildings with loaded area between $1m^2$ and $10m^2$, the following expression is used for determining the external pressure coefficient [7.2.1(1) Fig. 7.2]:

$$c_{pe} = c_{cp,1} - (c_{pe,1} - c_{pe,10})log_{10}A \tag{3.3}$$

3.2.1 Wind distribution on cylindrical roof shape

Figure 3.12 display the cylindrical roof shape used in the calculations. To find the wind load distribution on the shape Figure 3.16 is adopted. Very little information regarding wind pressure on cylindrical roof shape is found. Thus, the majority of the wind pressure calculations are done only on the basis of Figure 3.16 and remarks given in NA.7.2.8.

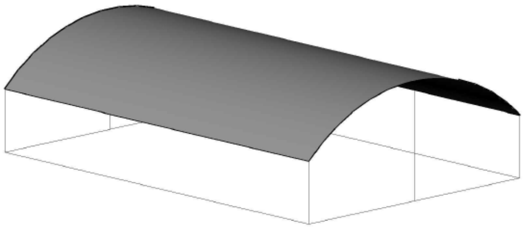


Figure 3.12: Structure with cylindrical roof shape used in the wind action calculations

Wind blowing parallel to the eaves, $\theta = 0°$

For the wind blowing parallel to the eaves, the code shows a sketch of how the wind zones are positioned on the roof (see Figure 3.13). However, the Eurocode does not give the dimensions of the distribution area.

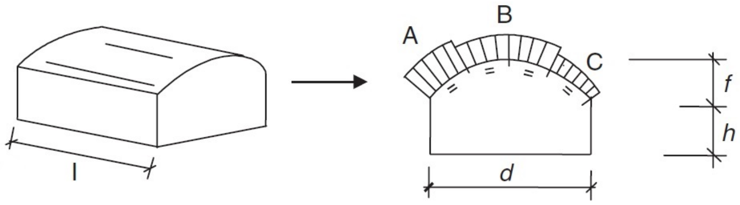


Figure 3.13: Cut-out of Fig. 7.11 in [36]: Wind distribution for wind parallel to eaves, $\theta = 0°$. Dimensions of the wind zones divisions are **not** provided.

Figure 3.14 however, illustrates how the Australian and New Zealand code divides the area. U corresponds to zone A in NS-EN 1991-4, T to B and D to C. Continuing, it is assumed these dimensions for the zones will be acceptable for our case.

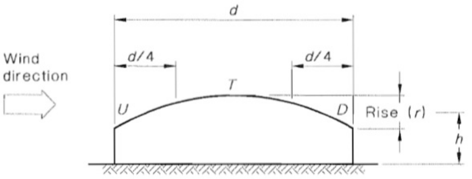


Figure 3.14: Fig. C3 in [37]: Wind distribution for wind parallel to eaves, $\theta = 0°$. Dimensions of the wind zone divisions are provided.

Figure 3.15 shows the side view dimensions of the roof. The following values are then calculated and taken as inputs in Figure (3.16) to obtain $c_{pe,10}$:

$$f/d = 1.5/7.5 = 0.2$$
$$h/d = 2/7.5 = 0.27$$

Notice that only $c_{pe,10}$ is given in [36] for the cylindrical roof when the wind is blowing parallel to the eaves, but no $c_{pe,1}$-value. A note in Section 7.2.8(1) explain that the values of $c_{pe,10}$ and $c_{pe,1}$ may be given in the National Annex. However, the Norwegian National Annex refers back to Section 7.2.8, despite that no information about how to obtain $c_{pe,1}$ can be found

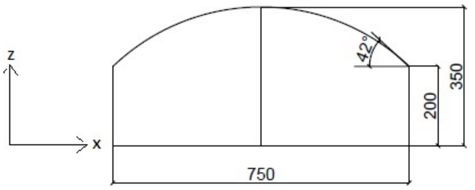


Figure 3.15: Section drawing of the cylindrical roof shape with dimensions

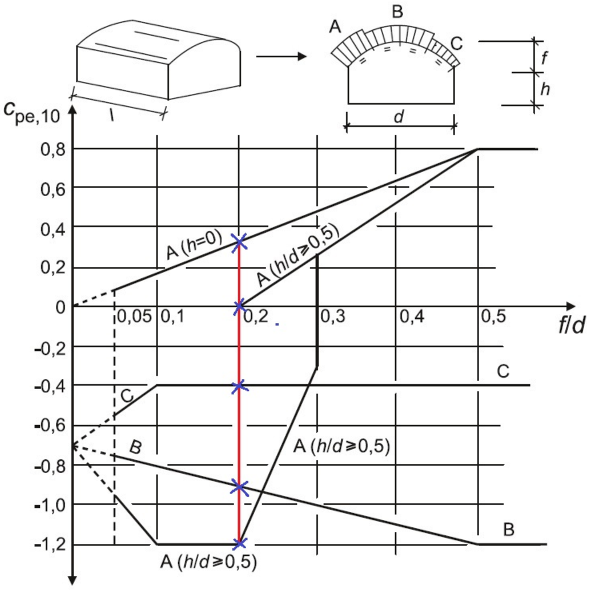


Figure 3.16: Fig. 7.11 in [36]: Graph with recommended values of external pressure coefficients $c_{pe,10}$ for roofs with a cylindrical shape and a rectangular base. The values corresponding to the cylindrical roof addressed in this section are highlighted

there. Fortunately for the cylindrical roof shape considered, all of the zone areas are bigger than $10m^2$, thus only the $c_{pe,10}$ pressure coefficient is relevant.

As stated in 7.2.8 (1), the coefficient for zone A should be obtained with linear interpolation if $0 \leq h/d \leq 0.5$. Since $h/d = 0.27$ the coefficient for zone A is obtained with linear interpolation between the $h = 0$ value and $h/d \geq 0.5$. For $h/d \geq 0.5$ two values for $c_{pe,10}$ need to be considered (see calculations below):

$$c_{pe,10,zoneA}(h = 0) = 0.36$$
$$c_{pe,10,zoneA}(h \geq 0.5) = 0$$

or

$$c_{pe,10,zoneA}(h \geq 0.5) = -1.2$$

This result in:

$$c_{pe,10,zoneA}(h = 0.27) = 0.36 - \frac{0.36 - (-0)}{0.5} * 0.27 = 0.17$$

or

$$c_{pe,10,zoneA}(h = 0.27) = 0.36 - \frac{0.36 - (-1.2)}{0.5} * 0.27 = -0.47$$

Table 3.3 shows the resulting pressure coefficient and external wind pressure blowing parallel to the eaves. The external pressure, w_e, is obtained with multiplying the pressure coefficient with the peak velocity pressure (see Eq. (3.2)):

$$w_e = q_p(z_e) * c_{pe} = 0,694 kN/m^2 * c_{pe}.$$

Figure 3.17 shows the distribution of wind zones for the roof when the wind is blowing parallel to the eaves.

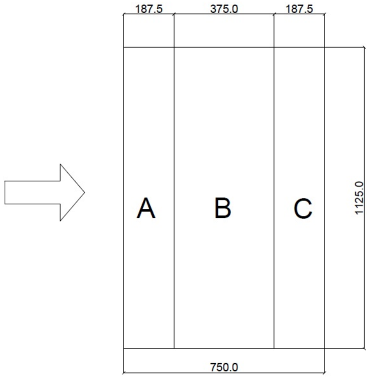


Figure 3.17: Wind zones for cylindrical roof when wind direction is parallel to eaves, $\theta = 0°$

Table 3.3: Cylindrical Roof Shape: Pressure coefficients and wind pressure due to wind blowing parallel to the eaves. Two alternative solutions are presented, and both of them needs to be considered when designing the structure

		Zone A	Zone B	Zone C
$c_{pe,10}$	Alt. 1	0.17	-0.9	-0.4
	Alt. 2	0.47	-0.9	-0.4
$w_e[kN/m^2]$	Alt. 1	0.12	-0.62	-0.28
	Alt. 2	-0.33	-0.62	-0.28

Wind blowing perpendicular to the eaves, $\theta = 90°$

For the situation of the wind blowing perpendicular to the eaves (see Fig. 3.18), National Annex 7.2.8 states that the data from the duopitched roofs found in Section 7.2.5 [Figure 7.8 c)] should be used (see Fig. 3.19). This corresponds to the recommendations of the Australian and New Zealand code ([38]).

NA.7.2.8 states that the following changes should be done:

- For Zone G, $c_{pe,10} = -1.4$ and $c_{pe,1} = -2.0$ should be used
- Zone F should be treated as Zone H

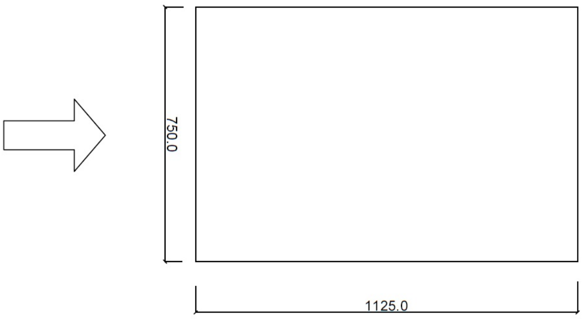


Figure 3.18: Aerial section drawing of the cylindrical roof model with wind direction perpendicular to the eaves, $\theta = 90°$

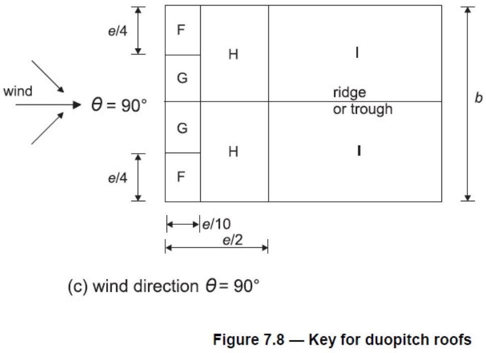

e/4
e/4
wind
θ = 90°
F
G
G
F
H
H
I
I
ridge
or trough
b
e/10
e/2
(c) wind direction θ= 90°
Figure 7.8 — Key for duopitch roofs

Figure 3.19: Fig. 7.8c in [36]: Wind zone division for a cylindrical roof shape with wind blowing perpendicular to eaves, $\theta = 90°$.

Taking these changes into consideration and using Table 7.4 b) in [36] (see Fig. 3.21), the external pressure coefficients $c_{pe,10}$ and $c_{pe,1}$ are obtained. The roof angle varies along the surface of the roof due to the cylindrical shape. In [38] it is concluded that the effective pitch angle of the curved roof should be taken as:

$$\alpha = \tanh{(f/0.5d)}, \tag{3.4}$$

hence giving:

$$\alpha = \tanh{1.5/(0.5 * 7.5)} = 21.8°.$$

The effective pitch angle is illustrated in Figure 3.20. Linear approximation between the pressure coefficients for 15 ° and 30 ° given in Table 7.4 b) is thus used to obtain the pressure coefficients for 21.8 °.

If the surface area is smaller than $10m^2$, Expression (3.3) is used to determine the pressure coefficient. Hence, AutoCAD is used to find the roof area and an approximation is done for the surface area of each zone. The results are presented in Table 3.4.

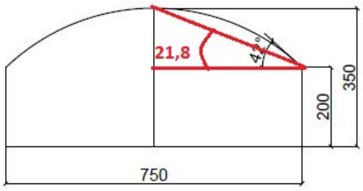


Figure 3.20: Section drawing of the cylindrical roof. Effective pitch angle 21.8° for the curved roof is presented.

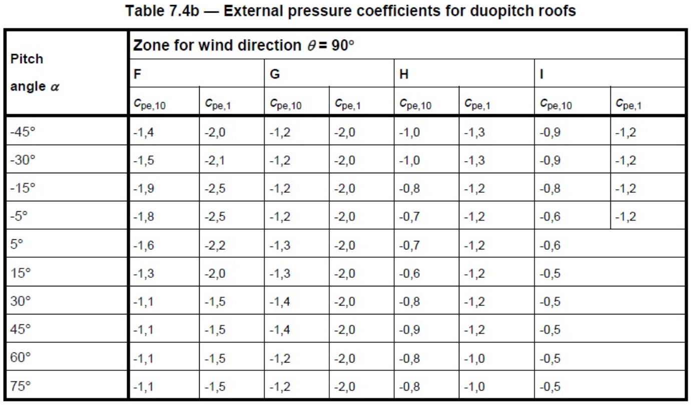


Table 7.4b — External pressure coefficients for duopitch roofs

Pitch angle α	Zone for wind direction θ = 90°							
	F		G		H		I	
	$c_{pe,10}$	$c_{pe,1}$	$c_{pe,10}$	$c_{pe,1}$	$c_{pe,10}$	$c_{pe,1}$	$c_{pe,10}$	$c_{pe,1}$
-45°	-1,4	-2,0	-1,2	-2,0	-1,0	-1,3	-0,9	-1,2
-30°	-1,5	-2,1	-1,2	-2,0	-1,0	-1,3	-0,9	-1,2
-15°	-1,9	-2,5	-1,2	-2,0	-0,8	-1,2	-0,8	-1,2
-5°	-1,8	-2,5	-1,2	-2,0	-0,7	-1,2	-0,6	-1,2
5°	-1,6	-2,2	-1,3	-2,0	-0,7	-1,2	-0,6	
15°	-1,3	-2,0	-1,3	-2,0	-0,6	-1,2	-0,5	
30°	-1,1	-1,5	-1,4	-2,0	-0,8	-1,2	-0,5	
45°	-1,1	-1,5	-1,4	-2,0	-0,9	-1,2	-0,5	
60°	-1,1	-1,5	-1,2	-2,0	-0,8	-1,0	-0,5	
75°	-1,1	-1,5	-1,2	-2,0	-0,8	-1,0	-0,5	

Figure 3.21: Table 7.4b in [36]: External pressure coefficient for duopitch roofs. The table correlates with Figure 3.19 and both is used in calculations of wind pressure on cylindrical roof shapes when wind blowing perpendicular to eaves, $\theta = 90°$.

Table 3.4: Cylindrical Roof Shape: Pressure coefficients and wind pressure due to wind blowing perpendicular to the eaves. Linear approximation between c_{pe} for 15° and 30° is used to find the resulting c_{pe} for 21.8°, and Expression (3.3) is used for zone G

		Zone G	Zone F/H	Zone I
$c_{pe,1}$	15 °	-2.0	-1.2	-0.5
	30 °	-2.0	-1.2	-0.5
	21.8 °	-2.0	-1.2	-0.5
$c_{pe,10}$	15 °	-1.3	-0.6	-0.5
	30 °	-1.4	-0.8	-0.5
	21.8 °	-1.35	-0.7	-0.5
Area $[m^2]$		3.1	25.9	64.1
Resulting c_{pe}		-1.71	-0.7	-0.5
$w_e[kN/m^2]$		-1.19	-0.48	-0.35

3.2.2 Wind distribution on spherical roof shape

Figure 3.22 shows the spherical roof shape in question and Figure 3.24 is adopted to find the wind load distribution. Notice that similarly to the cylindrical roof shape calculations, the majority of the calculations are solely based on one figure. Very little information to how this figure should be interpreted is given in the Eurocode, thus there have been difficulties in thoroughly understanding the provided information.

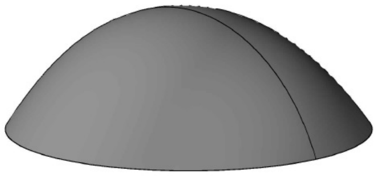


Figure 3.22: Structure with spherical roof shape used in the wind action calculations

The total height of the spherical roof in question $3.75m$ (see Fig. 3.23). Figure 3.24 presents the recommended values of external pressure coefficient $c_{pe,10}$. In accordance with this figure and with Figure 3.23 the following values are obtained: $f = 3.75$, $h = 0$ and $d = 11.25m$. Hence giving the following relations:

$$h/d = 0/11.25 = 0$$

$$f/d = 3.75/11.25 = 0.33$$

These values are used to extract the relevant data for $c_{pe,10}$ from the graph presented in Figure 3.24. Similar to the cylindrical roof, only the values of $c_{pe,10}$ can be found in the Eurocode. The load areas are not clearly defined for spherical roof shapes, and thus the corresponding pressure coefficient is difficult to determine. Although this is not stated in Eurocode, it is assumed that only the values of $c_{pe,10}$ will be relevant for the spherical roof shape.

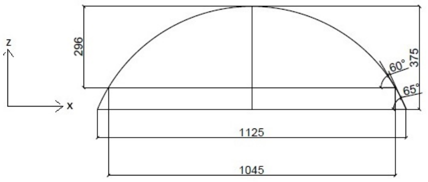


Figure 3.23: Section drawing of the spherical roof shape with dimensions

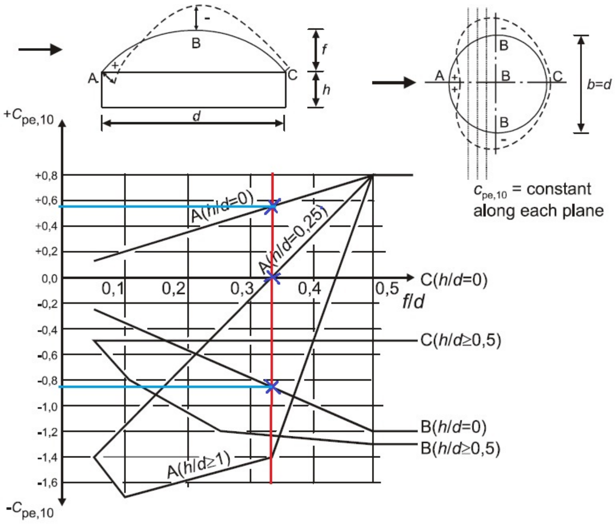


Figure 3.24: Fig. 7.12 in [36]: Graph with recommended values of external pressure coefficients $c_{pe,10}$ for roofs with a spherical roof shape and a circular base. The values corresponding to the spherical roof shape addressed in this section are highlighted

The pressure coefficients are constant along the planes perpendicular to the wind. The location of the different zones are defined in Figure 3.24. The figure also states that as a first approximation to the distribution along the arches of circles parallel to the wind, the zones can be determined by linear interpolation between the values in A, B and C.

The results are presented in Table 3.5 below, together with the wind pressures obtained with Equation (3.2). Figure 3.25 shows the distribution of the wind load together with the values obtained for the different wind zones.

Table 3.5: Spherical roof shape: Resulting pressure coefficients and wind pressures corresponding to the wind zones illustrated in Figure 3.25

	Zone A	Zone B	Zone C
$c_{pe,10}$	0.55	-0.85	-0
$w_e[kN/m^2]$	0.38	-0.59	0

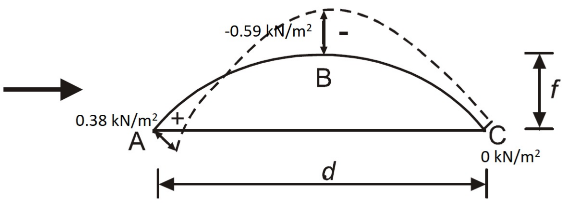


Figure 3.25: Wind distribution on the spherical roof shape with wind pressure values

3.2.3 Wind distribution on multi-span roof with cylindrical shape

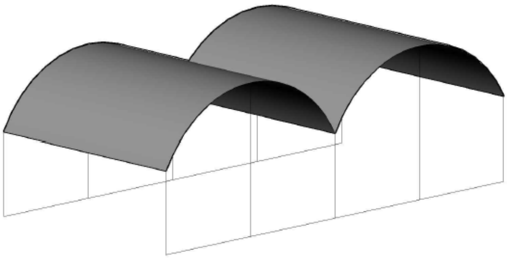


Figure 3.26: Structure with multi-span roof with cylindrical shape used in the wind action calculations

Section 7.2.7 *Multi-span roofs* in [36] suggests that the pressure coefficients for each span may be derived from the pressure coefficient for each individual span. Modification factors for the pressure in the wind directions $\theta = 0°$ and $\theta = 180°$ should be applied on each span. Figure 3.27 proposes factors which can be used.

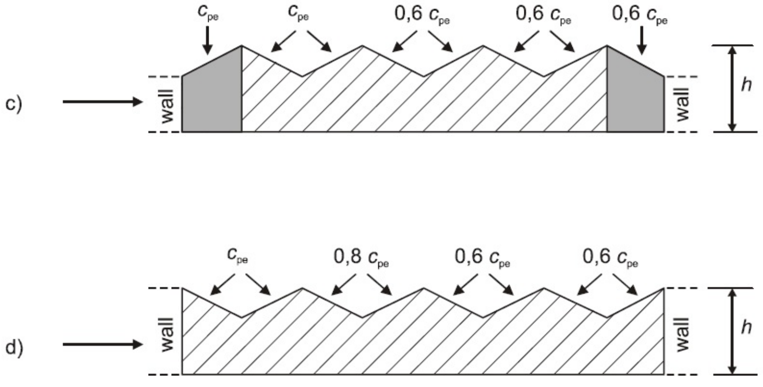


Figure 3.27: Fig. 7.10 c) and d) in [36]: Modification factors for wind pressure coefficients for multi-span roofs for wind directions $\theta = 0°$ and $\theta = 180°$

There are no modification factors for multi-span cylindrical roofs, and hence there are difficulties regarding the application of the suggested factors. Assuming that the illustration c) in 3.27 (Fig. 7.10 c) in [36]) will be most reasonable to use, a structure consisting of only two cylindrical shapes will most likely have negligible effects caused by the modification factors. Thus, this structure is assumed to have the same wind distribution for each of the two spans as derived for the individual cylindrical roof in Section 3.2.1.

3.2.4 Summary of wind calculations

Table 3.6: Summary of wind actions calculated for the cylindrical and spherical roof shape

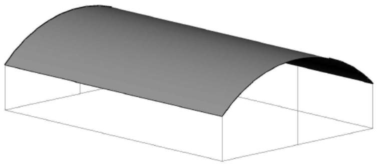
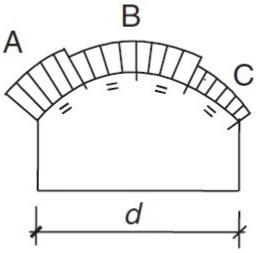
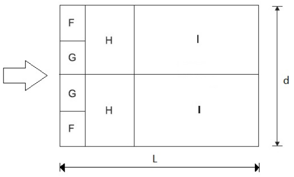
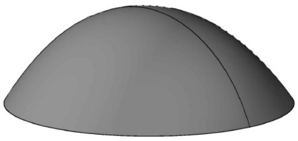
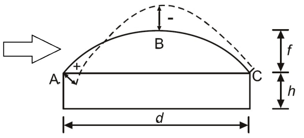
Shell Shapes	EC Figures	Zone	$w_e[kN/m^2]$	
	Parallel to the eaves, $\theta = 0°$			
		-	Alt. 1	Alt. 2
		A	0.12	-0.33
		B	-0.62	-0.62
		C	-0.28	-0.28
	Perpendicular to the eaves, $\theta = 90°$			
		G	-1.12	
		F/H	-0.48	
		I	-0.35	
		A	0.38	
		B	-0.59	
		C	0	

Table 3.6 summarise the wind calculations done for the cylindrical and spherical roofs. The multi-span roof is not accounted for as it was assumed to have the same wind distribution as two cylindrical roofs.

The wind action calculation depends on many variables. Several assumptions and simplifications regarding the wind direction and terrain were made to obtain the peak velocity pressure. The pressure coefficients and wind load distribution depend on the geometry of the roof. The Eurocode presented inadequate information with regards to the cylindrical and spherical roof shape. Only one Figure for each of the roof shapes was given to obtain the pressure coefficient and describe wind distribution. Difficulties in understanding the limited information leads to ecen more uncertain results.

3.3 Review of Actions according to Eurocode

In the beginning of Eurocode under *In Status and field of application of Eurocodes*, the following is stated:

> *The Eurocode standards provide common structural design rules for everyday use for the design of whole structures and components products of both traditional and innovative nature. Unusual forms of construction or design conditions are not specifically covered and additional expert consideration will be required by the designer in such cases.* [39]

After studying the aforementioned cases, it is clear that the Eurocode falls short in providing information on snow- and wind load distributions on shell structures, and roofs of particular geometry.

As Standards are built on empirical data, there are many uncertainties connected to the information given as there might be simplifications and generalizations in order to transform practice into theoretical expressions. This is an important factor needed to be addressed if the Eurocode is adopted for shapes not explicitly defined by the code. In other words, deducing information cannot be done with great reliability.

4 | Parametric Study

In this chapter a further investigating of the shapes from Chapter 3 will be performed using Grasshopper and the plug-in Karamba (see Ch. 2 for information about the software). These shapes include the cylindrical and spherical roof. In addition, a flat roof is included to examine the structural differences between curved and flat structures. Going forward, the shapes will be considered as three-dimensional structures and addressed as the barrel vault, dome and the plate (see Fig. 4.1).

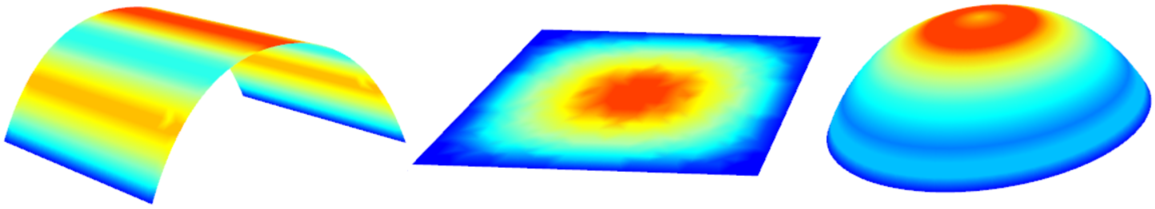


Figure 4.1: Geometric shapes to be investigated in *Parametric Study*. From left to right: Barrel vault, plate and dome.

To examine how the grid shell behaves compared to a "perfect" shell with a continuous surface, both a timber grid shell and concrete shell is developed in Grasshopper through algorithmic modelling. The aim is to understand and compare the efficiency of the shapes, as well as the differences of using reticulated bars vs. a continuous surface. The three shapes are parametrically defined and assigned concrete and timber grid shell properties with Karamba. The parametric behaviour of the models ensure flexibility with regards to alternating between the geometric shapes.

Parameters considered

The parameter (variable) of this Parametric Study will be rise/span (r/s)-ratio

Structural response and internal actions

The structures in question will be analysed with regards to displacements, bending moments and axial forces. These are considered to be the most affected when alternating between the shapes and given parameter (r/s-ratio). The *Shear forces* were found to be small in the initial stage, and will consequently not be considered in this chapter.

Loads considered

The loads considered in the Parametric Study is self-weight of the structures and uniformly distributed snow load. The magnitude of the snow load is selected to be $6.11kN/m^2$. In Section 3.1.1 (Sec. 5.3.5(1) in [35]), it was found that the snow load will only be distributed on the structure where the curvature is less than 60 degrees as the snow is assumed to slide off when the angle is steeper. This assumption will be accounted for in this chapter.

4.1 Establishing geometry in Grasshopper

To make the models as parametric as possible, the geometry is modelled in Grasshopper, making Rhino merely a visual environment in this case. Lines, generic curves and Non-Uniform Rational B-Splines (NURBS) are used to establish the geometry. Lines mainly to represent the grid framework of the timber grid shell, and the curves to establish the overall shape of the structures. The NURBS curves are mathematical representations of 3D geometry and are defined by degree, control points, knots and evaluation rule [40]. The given definitions will be not be further explained in this thesis, but more information can be found on the Rhino home page [40].

In order to make the task of alternating between shapes as clear and straightforward as possible, the basic geometry of the structures are defined in the very beginning of the code. The geometries are given a surface, which is the starting point of the code for the structural analysis performed with Grasshopper and Karamaba.

Barrel vault

The **barrel vault** is symmetric over its length, and is modelled by the use of NURBS curves to get the cylindrical shape. The geometry of the barrel vault is defined by a NURBS curve with degree and vertices (control points) taken as inputs. The geometry is assembled with a second degree NURBS curve consisting of 4 control points constrained by symmetry, see Figure 4.2. The symmetry enables the possibility of having fewer parameters, simply a slider for span and height. The outer points (point 1 and 4 in Fig. 4.2) can be moved in the y-direction, changing the span of the barrel vault. The y-coordinate of the inner points is defined as two-thirds of the y-coordinate of the outer points. Hence, the inner points adjust to the change of span. The height of the Barrel vault is modified by moving the slider for height connected to the z-coordinate of the inner points (point 2 and 3). The length of the Barrel vault is defined to be equal to the span, i.e the floor plan is quadratic.

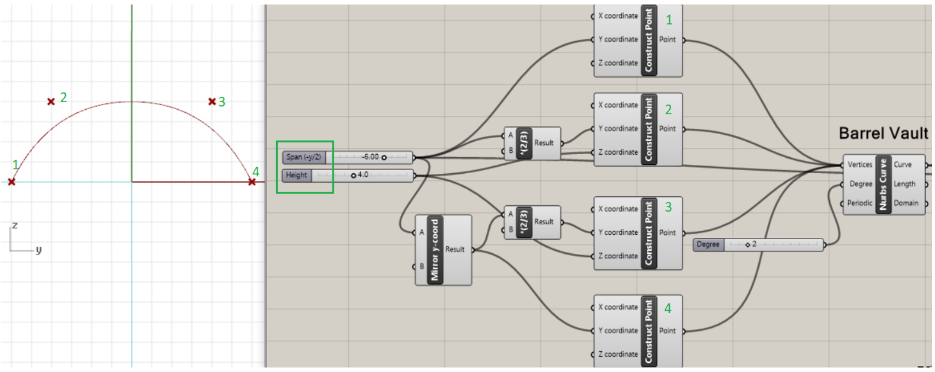


Figure 4.2: Barrel vault constructed by NURBS curve of 2nd degree with 4 control points

Plate

The **plate** is modelled in Grasshopper by constructing four corner points, and creating a surface from these points. Figure 4.3 shows how the plate surface was established. The surface is a quadrilateral with equal side, and is established by constructing two points and thereafter translating these points in a distance equal to the span with the $Move$-component.

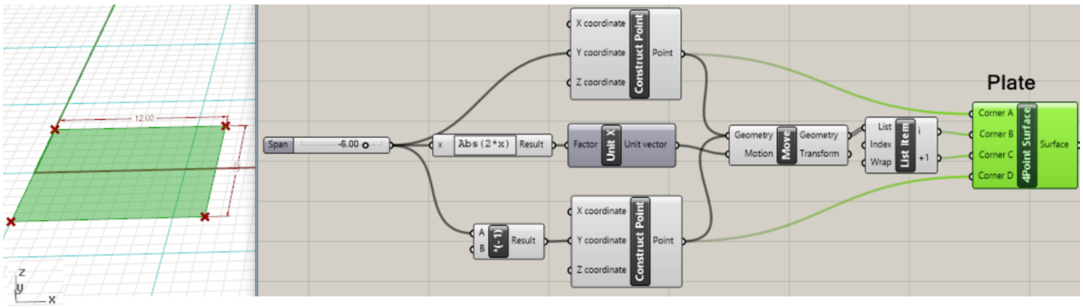


Figure 4.3: Surface of plate modelled by the input of 4 corner points

The purpose of adding the plate to this parameter study is mainly to observe the different structural behaviour of plane structures (plate) vs. curved ones (barrel vault and dome). This will be further addressed later in this chapter (see Section 4.4.4).

Dome

The **dome** is modelled by defining a curve similar to an quarter-arc and revolving it about a vertical axis. Figure 4.4 shows the code for how the curve is created. A slider is defined both for the radius and the height of the dome, making it possible to change one dimension independently of the other. These two sliders define the beginning and end point of the curve. A line is drawn between these two points, and the center point on the line is found. The point is then moved with the $Move$-component a distance (amplitude) in a direction pointing outwards from the center of the dome. It is observed that the height divided by 3.5 results in a nice shape of the curve. The middle point of the curve is thus defined, and the curve is created by interpolating between the radius, middle and height point. The surface of the dome is further created by the $Revolution$-component with the input of the 3 point curve and the axis of revolution (z-axis), see Figure 4.5.

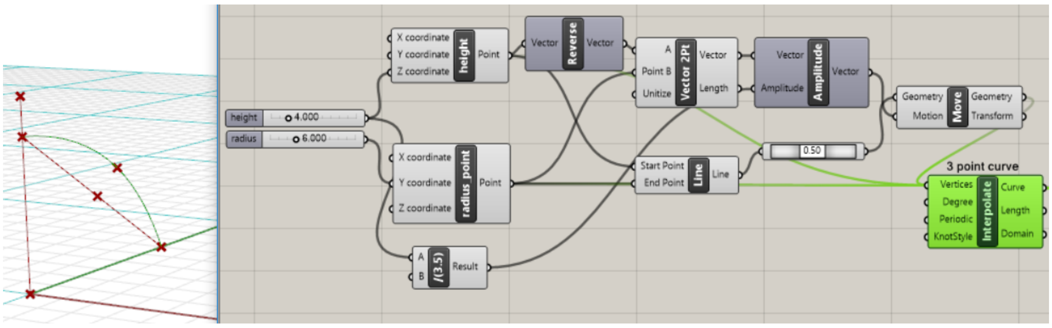


Figure 4.4: Dome geometry - construction of a 3 point curve

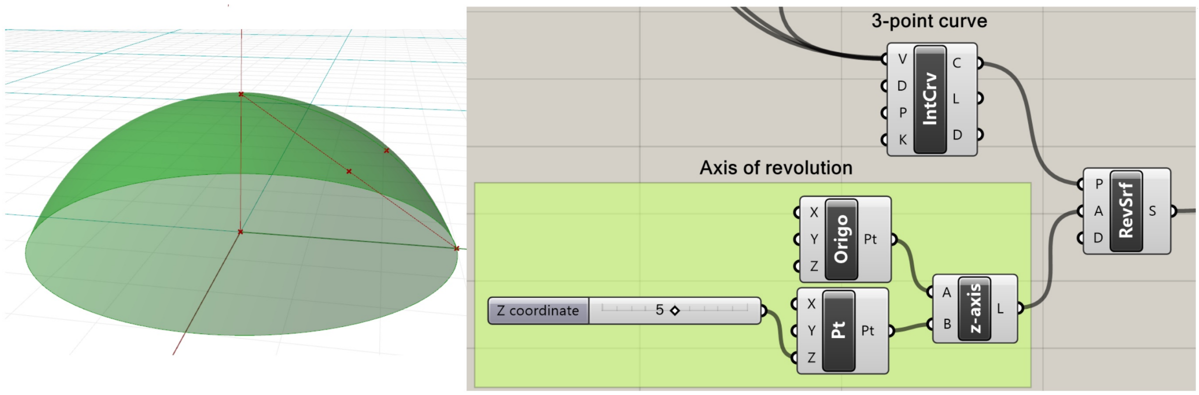


Figure 4.5: Dome geometry - surface constructed by revolution of the 3 point curve

From geometric shapes to structural model

In order to start the FE-analysis of the concrete shell and the grid shell, the geometric shapes need to be assigned structural properties. The next two sections (Sec. 4.2 and 4.3) explain how the Grasshopper code was established for the grid shell and the continuous shell. It will also be explained how the structural analysis was performed with Karamba, and how the data (which will be presented in Sec. 4.4) was collected and organised. In Appendix B *Grasshopper code for Parametric Study* a more detailed illustration of the code is presented.

4.2 Grid shell in Grasshopper and Karamba

The pattern of the grid will have an effect on the results. For the first analysis, a grid shell consisting of rectangles with diagonal bracing is chosen (see Figure 4.6). Different patterns and their behaviour will be examined further in Chapter 5.

4.2.1 Creating Beam Elements

The surface defining the shape is divided into rectangular surfaces with an area of approximately $1mx1m$, using the $DivideDomain$- and $IsoTrim$-components. In order to create the grid shown in Figure 4.6, the u count in $DivideDomain$ is always set to an even number, and the v count is set to an odd number. This is ensured using an even- and odd-number generator created with $C\#$.

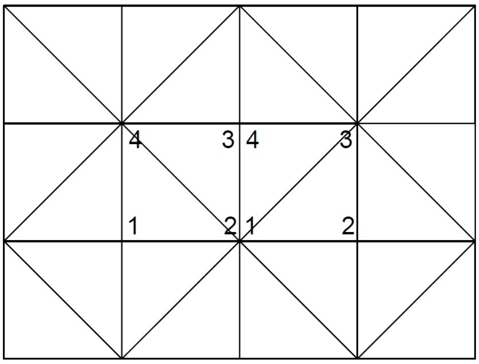


Figure 4.6: Grid shell pattern adopted for the Parametric Study of the reticulated timber shell

Figure 4.6 shows the grid pattern adopted for the Parametric Study, with the nodes 1-4 positioned in a specific order. Every second face of the grid is selected using $CullPattern$ and the vertices with index 4 and 2 are connected. The face selection is then inverted and the other diagonal is created by connecting vertices 1 and 3.

Figure 4.7 shows how the diagonal bracing as well as the main grid and edge beams are generated.

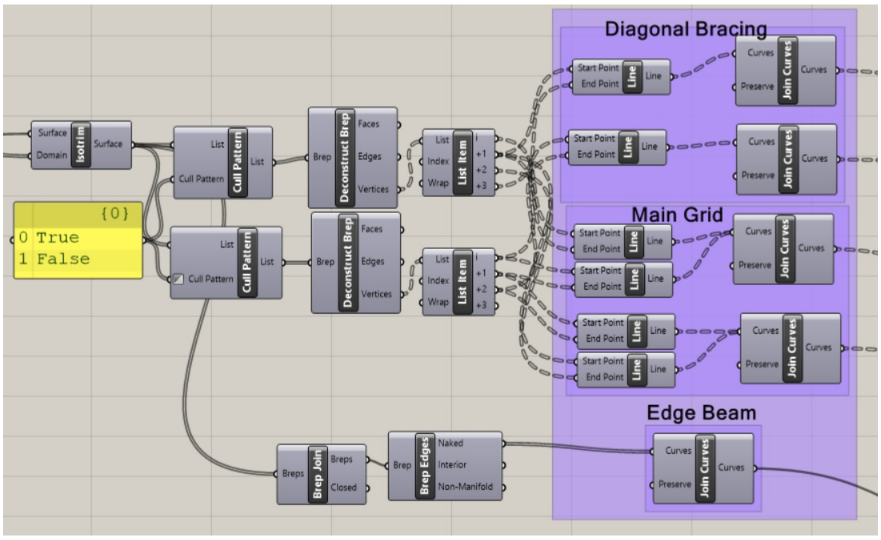


Figure 4.7: Grasshopper code for establishing grid pattern like illustrated in Fig. 4.6

This way of dividing the shape into equally sized rectangles and creating a grid, only works for a rectangular surface. The dome however, will have a spherical form with curved boundaries. Therefore, to ensure that the analysed grid shell dome has the same pattern as the barrel vault, the grid for the dome is first generated on a flat surface and then projected onto the surface of the curved dome (see Fig. 4.8).

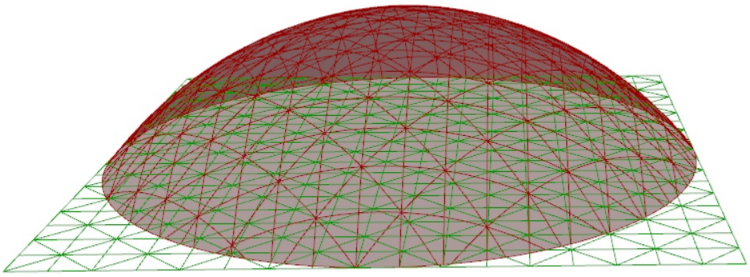


Figure 4.8: Projecting grid pattern onto the surface of the dome

$Explode$ is used to divide the curves between each intersection in to line-like segments which are then converted to beam elements using $LineToBeam(Karamba)$. The beam elements are assigned an identifier (BeamID). This is later used to assign properties to the grid. The Element-output is a list over all the beam elements in the model, and the Points-output is the global coordinates of all the end nodes of the beam elements. The beam elements are orientated so that the local z-axis of each beam element is perpendicular to the surface in the midpoint of each element (see Fig. 4.9).

At the end-points of the beam elements the $Beam - Joints(Karamba)$ is used to control the joint properties. For the analysis in this chapter, the joints are assumed to be fixed.

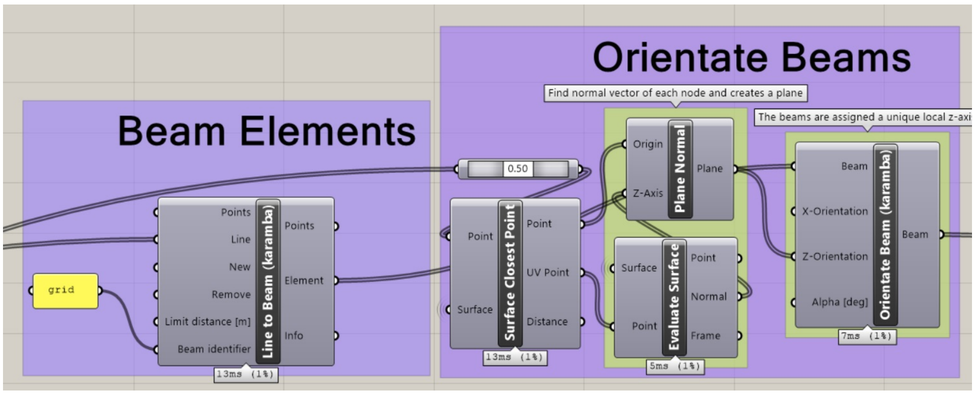
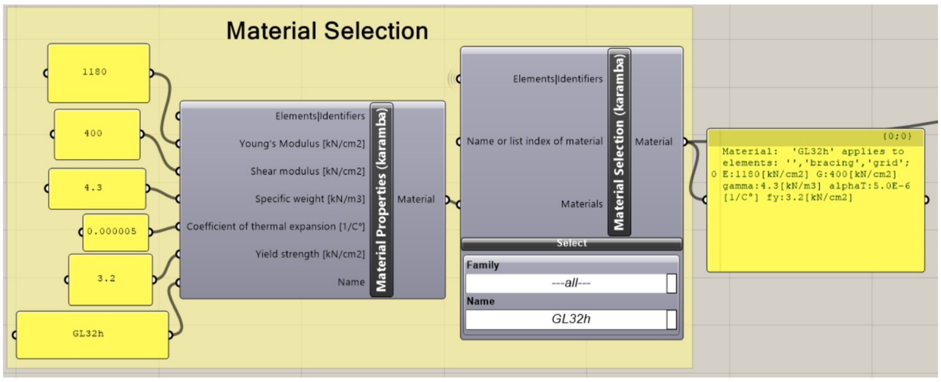


Figure 4.9: Line segments taken as inputs in the $LineToBeam$-component and orientated to make sure the local z-direction of the beams measured in the middle point of the beam is located in the direction parallel with the normal vector of the surface in that point

4.2.2 Specifying Material and Cross Section Properties

Gl32h is used as the material for the Parametric Study. The characteristic properties of GL32h are found in NS-EN 14080 ([41]) and taken as inputs in the $MaterialProperties(Karamba)$-component. The value of the shear modulus however, differ from the one given in [41]. In Karamba the shear modulus needs to be larger than E/3 (E= Young's Modulus) and smaller than E/2, because it assumes an isotropic material. Timber is an orthotropic material and will have a smaller shear modulus than E/3, hence this assumption is not completely accurate. However, to continue the analysis and make sure the material behaves as expected by Karamba, the shear modulus is set between E/2 and E/3.

The material properties are then transferred to the $MaterialSelection(Karamba)$ and linked to the beam elements with the BeamID's (see Fig. 4.10).



```
0 Material:  'GL32h' applies to
  elements: '','bracing','grid';
  E:1180[kN/cm2] G:400[kN/cm2]
  gamma:4.3[kN/m3] alphaT:5.0E-6
  [1/C°] fy:3.2[kN/cm2]
```

Figure 4.10: Specifying material properties for GL32h in Karamba

A solid quadrilateral cross section with $h = b$ is used for all components in this analysis. This is achieved by selecting the Trapezoid cross section in the $CrossSections(Karamba)$-component and setting upper width, lower width and height to the same value (see Figure 4.11).

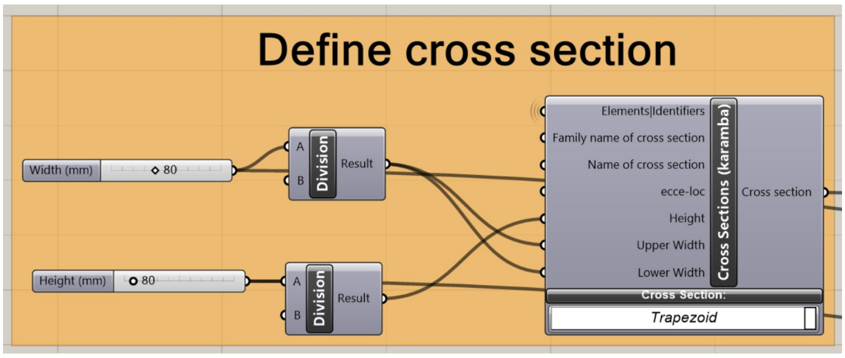
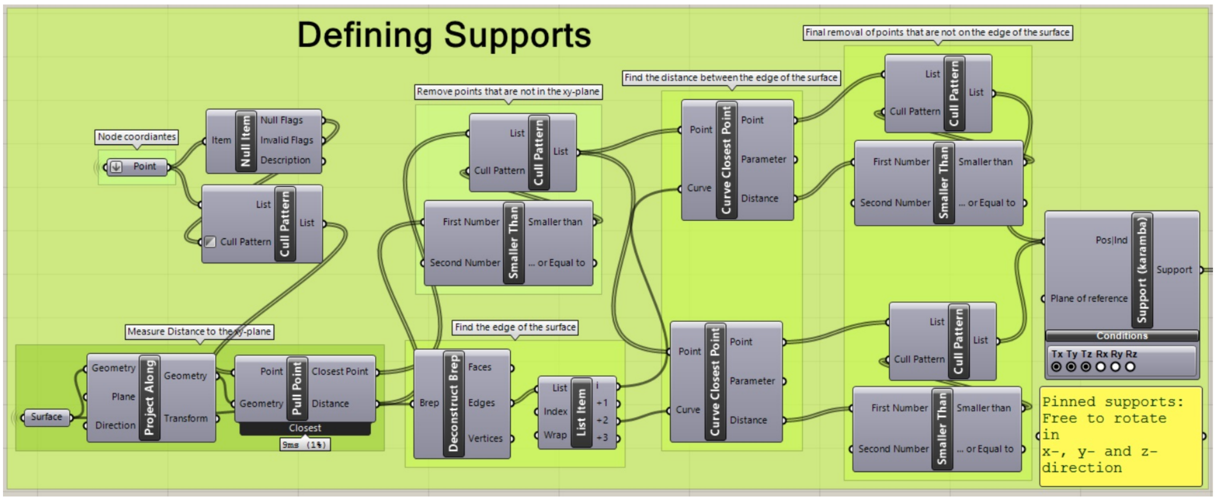


Figure 4.11: Defining a solid cross section with $CrossSections(Karamba)$-component

4.2.3 Specifying Supports

Once the elements are created and cross sections with corresponding material are assigned, the supports needs to be defined. A series of points located in the xy- plane as well as on the edges of the shape are set as supports. The points already exist in the model as end-points of the beam elements.

To locate the relevant points $PullPoints$ is first used to measure the distance between each point and the xy- plane. Then $CullPattern$ is used to remove the points where the distance is larger than zero. The remaining points is again taken as input in $PullPoints$, to measure the distance between the points and the edge of the surface. Finally, the points at the edge curve is retrieved and taken as inputs in $Support(Karamba)$ (see Fig. 4.12).



Pinned supports:
Free to rotate in
x-, y- and z-direction

Figure 4.12: Grasshopper code for locating the points (nodes) on the ground (xy-plane) and feeding them as input for the $Support(karamba)$-component. The supports are pinned.

In the $Support(Karamba)$-component the support condition are specified with six degrees of freedom (dofs), three rotational and three translational. The supports are assumed to be pinned, hence only the three translational dofs are fixed (see Fig. 4.13).

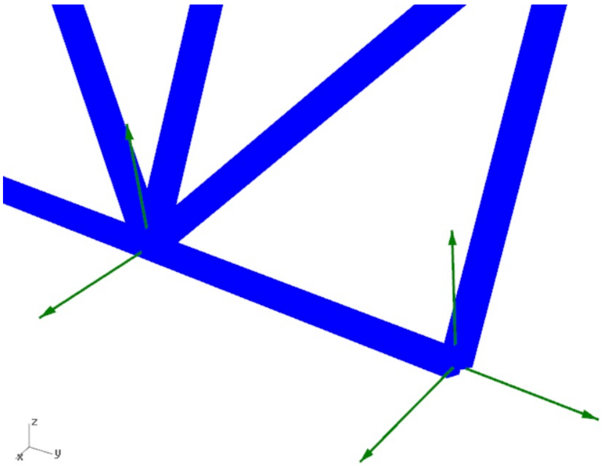


Figure 4.13: Zoom-in on the pinned supports with fixed translations, and three rotational dofs

4.2.4 Specifying Loads

As mentioned previously only the snow load and the self-weight of the structure will be considered in this analysis. The snow load (Fig. 4.14 (A)) is projected onto the surface (Fig. 4.14 (C)). This is a simplification as in reality the snow load on the grid shell will be globally projected as illustrated in Figure 4.14, (B). For a curved surface this means that the points located on top of the surface (where the surface has the least curvature) will be affected by a greater snow load compared to the points located near a steeper curvature.

For simplicity, the illustration (C) will be adopted, and the same point loads are lumped onto the end-points (nodes) of the beam elements. The snow load will only affect the points which are located on the surface with a curvature less than 60° (see Sec. 3.1.1).

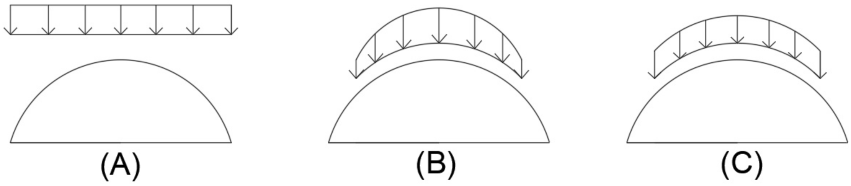


Figure 4.14: (A) uniform snow load. (B) globally projected snow load for angle less than 60° (C) simplified projected snow load for angle less than $60°$.

To find the magnitude of the point load, the projection of surface area with the relevant curvature and the points located in this area needs to be found.

First the uv-coordinates of all the node points are found and taken as inputs in the $EvaluateSurface$-component. Here the normal vectors of each point is extracted and used to determine the inclination of the surface. All the normal vectors are set to have positive valued components only. The angle between the normal vector of the surface and the unit vector $(0, 0, 1)$ is then determined (see Fig. 4.15). All points with an angle higher than 60° are removed using $CullPattern$ (see Fig. 4.16).

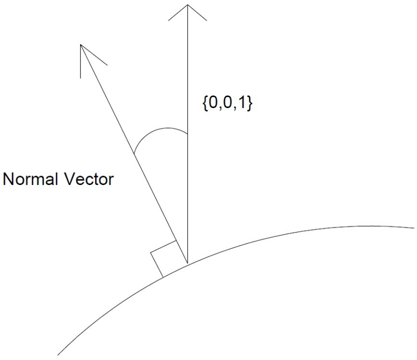


Figure 4.15: Illustration of the angle between the normal vector and unit vector

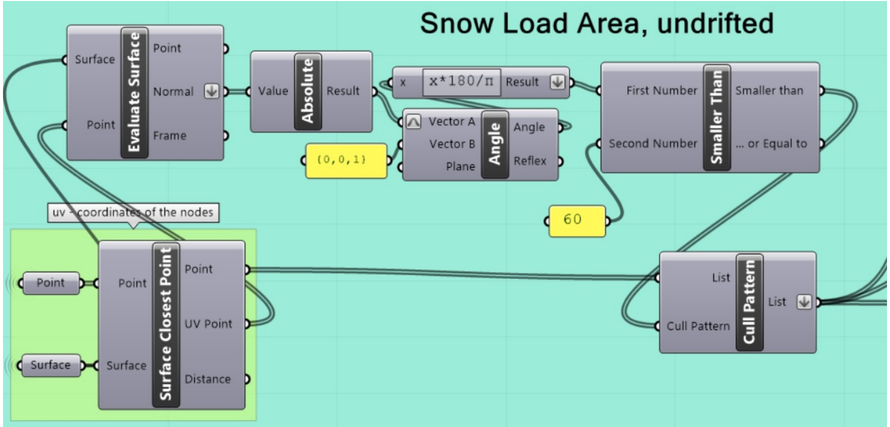


Figure 4.16: Points on the surface with curvature smaller than 60°, and hence the area defining the domain for where the snow will accumulate according to [35]

The remaining points are first projected to the ground and then used as inputs in $DelaunayMesh$ to establish a triangulated mesh (see plane red area in Fig. 4.17). The mesh represents the projected area of the surface with a curvature less than 60°, hence the points used to create the mesh are the points affected by the snow load (see green points in Fig. 4.17).

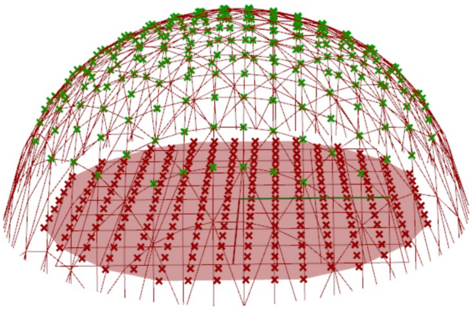
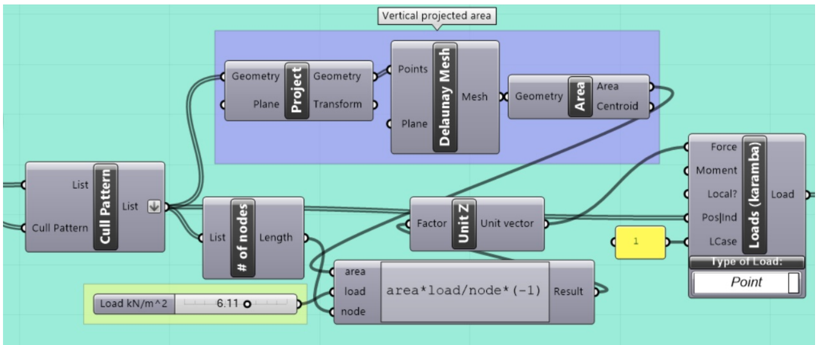


Figure 4.17: Illustration of the points (green) where angle of surface $\leq 60°$ and hence the domain for the snow load. The figure also shows the projected area on the xy-plane

The area of this projected surface and the number of points within the area are used to approximate the magnitude of the point loads (see Fig. 4.18). As a result the value of the point loads and the points of interest are found and used as inputs in $Loads(Karamba)$.



Load kN/m^2 6.11

Type of Load: Point

Figure 4.18: Input to the $Loads(karamba)$-component with magnitude of point loads based on the area of where the snow distributes on the roof shape

4.2.5 Assemble the model and perform structural analysis with Karamba

To perform a structural analysis in Karamba all the necessary information is collected in the $AssembleModel$-component. Initially small deformations are assumed, and the model output is connected to the analysis component $AnalyzeThI(Karamba)$ which performs a first order analysis (see Fig. 4.19). Later, a linear buckling analysis is done with $AnalyzeThII(Karamba)$ and the $BucklingModes(Karamba)$. $ModelView(Karamba)$- and $BeamView(Karamba)$-components are used to visualise and interpret the results (see Fig. 4.20).

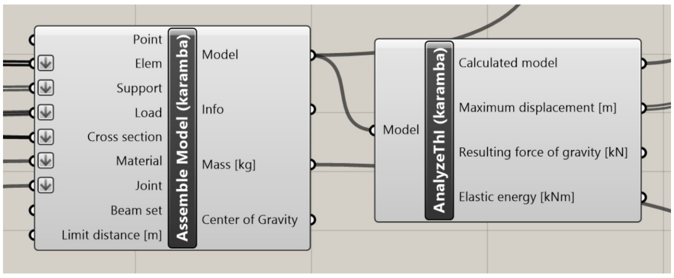
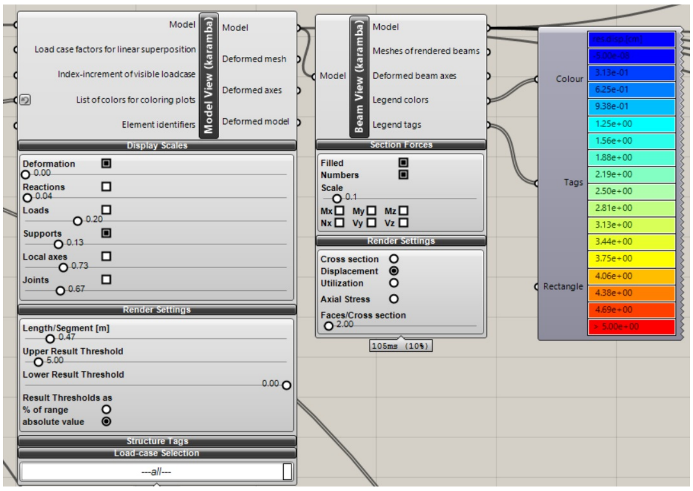


Figure 4.19: $AssembleModel$-component used to collect all the information about the structural model



Deformation ☒ 0.00
Reactions ☐ 0.04
Loads ☐ 0.20
Supports ☒ 0.13
Local axes ☐ 0.73
Joints ☐ 0.67

Length/Segment [m] 0.47
Upper Result Threshold 5.00
Lower Result Threshold 0.00
Result Thresholds as
% of range ○
absolute value ●

---all---

Filled ☒
Numbers ☒
Scale 0.1
Mx ☐ My ☐ Mz ☐
Nx ☐ Vy ☐ Vz ☐

Cross section ○
Displacement ●
Utilization ○
Axial Stress ○
Faces/Cross section 2.00

Figure 4.20: $ModelView(Karamba)$- and $BeamView(Karamba)$-components used to view the result from the analysis in Rhino

4.2.6 Retrieving Results

Beam Resultant Forces

The $BeamResultantForces(Karamba)$-component retrieves the maximum resultant forces for all beam elements in the model (see Fig. 4.21). The output is a data tree where each branch represent a load case with resulting beam forces. A C# script is created to control which load cases should be added together to achieve the total beam force (see Fig. 4.22). The yellow panels in the figure shows the results of the largest axial force N, the resulting bending moment M_{res} and shear force V for all beam elements and load cases/ load combinations. The resulting bending moment M_{res} is defined as:

$$M_{res} = \sqrt{M_y^2 + M_z^2}$$

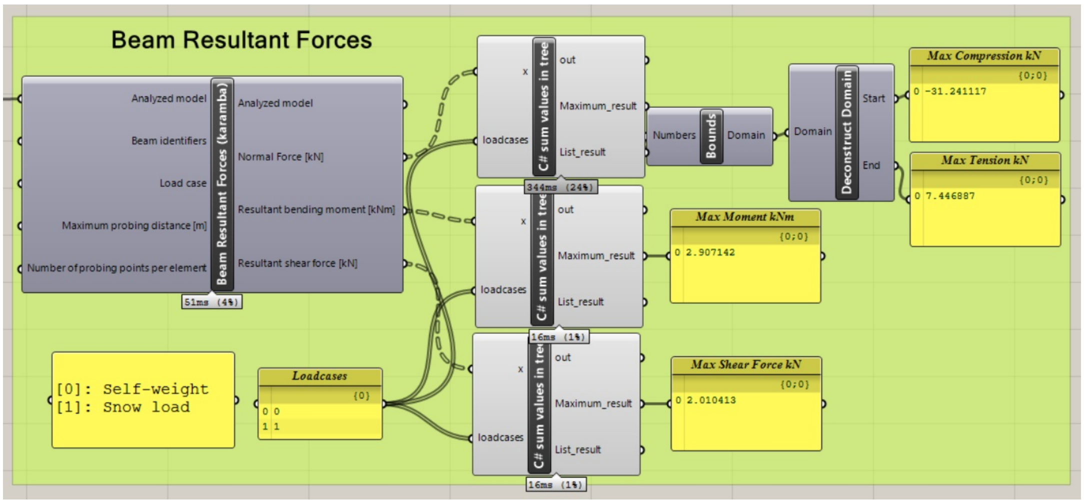
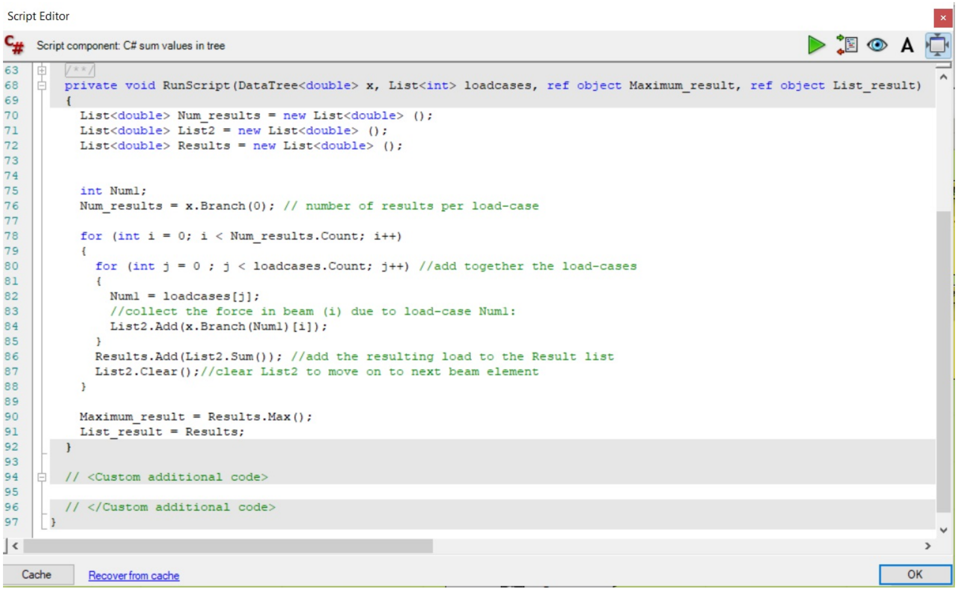


Figure 4.21: $BeamResultantForces(Karamba)$-component used to collect axial force N, the resulting bending moment M and shear force V



```
/**/
private void RunScript(DataTree<double> x, List<int> loadcases, ref object Maximum_result, ref object List_result)
{
  List<double> Num_results = new List<double> ();
  List<double> List2 = new List<double> ();
  List<double> Results = new List<double> ();


  int Numl;
  Num_results = x.Branch(0); // number of results per load-case

  for (int i = 0; i < Num_results.Count; i++)
  {
    for (int j = 0 ; j < loadcases.Count; j++) //add together the load-cases
    {
      Numl = loadcases[j];
      //collect the force in beam (i) due to load-case Numl:
      List2.Add(x.Branch(Numl)[i]);
    }
    Results.Add(List2.Sum()); //add the resulting load to the Result list
    List2.Clear();//clear List2 to move on to next beam element
  }

  Maximum_result = Results.Max();
  List_result = Results;
}

// <Custom additional code>

// </Custom additional code>
}
```

Figure 4.22: C#- script used to combined and retrieve the correct beam forces

Figure 4.23 shows the direction of the local coordinate axis of the beam elements. The cross sectional forces are defined according to this coordinate system. Negative normal force means the beam element is in compression, positive means tension.

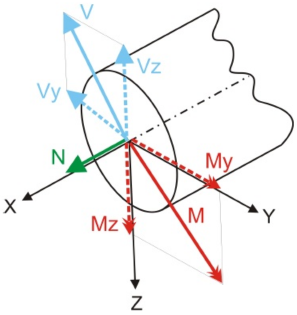
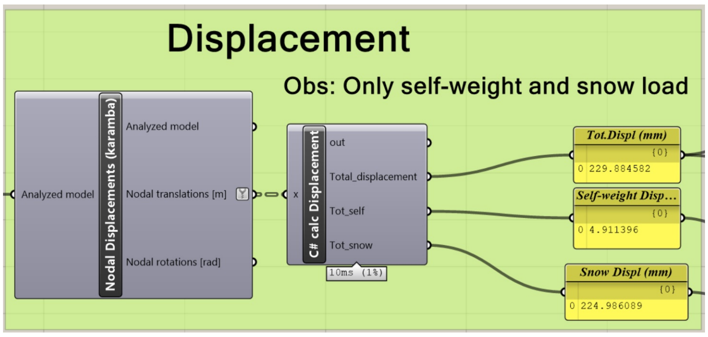


Figure 4.23: Normal force, resulting shear force and bending moment related to the local coordinate system of the cross section (Fig. 83 in [42])

Reaction Forces in the grid shell
$ReactionForces(Karamba)$- component are used to find the reaction forces at the supports. Also here a C# script is created to sort the reaction forces after the different load cases. The results are presented as the average reaction force per meter support.

Displacements in the grid shell
The displacement of the structure is found with collecting the nodal displacements with the $NodalDisplacement(Karamba)$- component and processing these with C#. The nodal translation output gives the nodal translations in the global x-, y- and z-direction (d_x, d_y, d_z). The total displacement is defined as:

$$d_{total} = \sqrt{d_x^2 + d_y^2 + d_z^2} \tag{4.1}$$


Displacement

Obs: Only self-weight and snow load

Tot.Displ (mm)
{0}
0 229.884582

Self-weight Disp...
{0}
0 4.911396

Snow Displ (mm)
{0}
0 224.986089

Figure 4.24: $NodalDisplacement(Karamba)$-component used to retrieve the maximum displacement due to self-weight and snow load

4.3 Concrete shell in Grasshopper and Karamba

The parametric model of the concrete shell is built in the same manner as the grid shell. However, there are some principal differences. The main one being that the concrete shell model is based on shell elements, contrary to the beam elements in the grid shell. Another difference is that the concrete model of the barrel vault, plate and dome is created in the same GH model.

4.3.1 Mesh and Shell Elements

In order to start the FE-analysis of the concrete shell a mesh needs to be created for the shell surface. The mesh density affects the accuracy of the finite element model as well as computational power needed. A denser mesh gives more accurate result but also demands longer computation time. $MeshBreps$ is used to mesh the surface, and a mesh resolution of $0.5meters$ is assumed to be sufficient and taken as input (see Fig. 4.25). $MeshtoShell(Karamba)$ converts the mesh geometry in Grasshopper to shell elements in Karamba (see Sec. 2.6). The shells are assigned an identifier called "s".

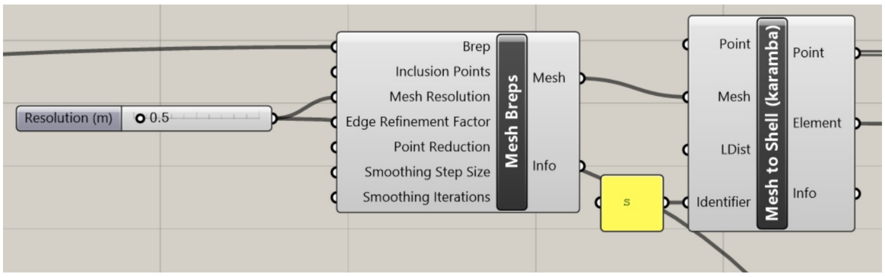


Figure 4.25: $MeshBrep$-component used to generate the mesh with a selected resolution of 0.5. The $MeshToShell(Karamba)$-component creates the structural Shell Elements from the mesh geometry.

4.3.2 Specifying Material and Cross Section Properties

The material properties are specified using $MaterialSelection(Karamba)$. In the drop-down menu the predefined material concrete C30/37 is chosen. Note that default values for concrete C30/37 in Karamba are used and the material properties will differ from the ones provided by the code for concrete (NS-EN 1992-1-1).

The cross section is defined using the $CrossSections(Karamba)$- component, where shell is selected in the drop-down menu and a thickness is assigned. The shell identifier, "s" is connected to $MaterialSelection(Karamba)$- and the $CrossSectionSelector(Karamba)$-components to apply the properties to the shell elements.

4.3.3 Specifying Supports

The vertices of the shell meshes that are located on the edge of the surface and in the xy-plane are specified as supports (see Fig. 4.26).
The vertices are extracted from the $MeshToShell(Karamba)$-component, otherwise the grasshopper code for determining the relevant points are similar to the one used for grid shells (see Fig. 4.12).

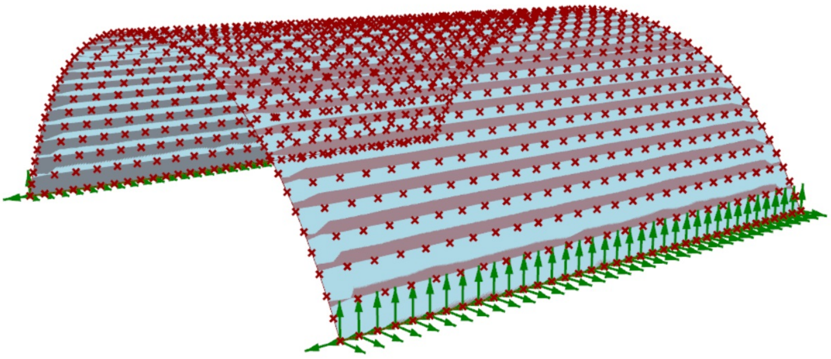
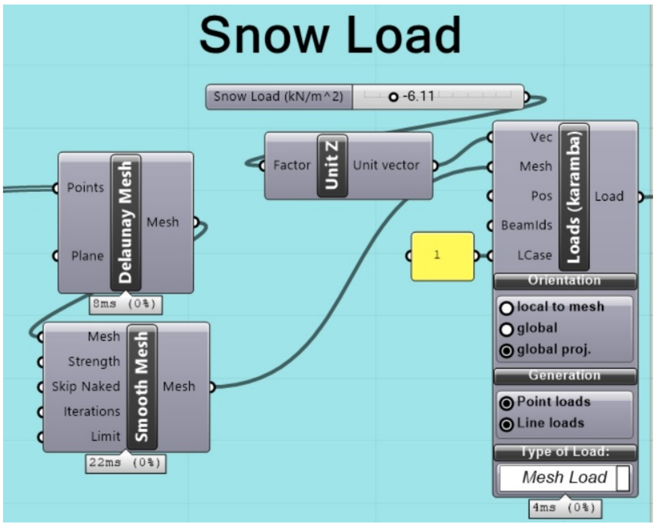


Figure 4.26: Illustration of the supports for the concrete barrel vault. The points on the shell surface represent the vertices of the shell elements

4.3.4 Specifying Loads

Similar to the grid shell, the self-weight of the structure and snow load is applied. The surface area of the structure with a curvature less than 60° is found using the same code as for the grid shell (see Fig. 4.16). However, instead of approximating the globally projected load (see Fig. 4.14 (C)), the $Loads(Karamba)$-component for mesh loads allows for a simpler and more accurate method. First $DelaunayMesh$- and $SmoothMesh$- component is used to establish a mesh based on the relevant points (i.e. the points located on the surface with a curvature less than 60°). Then the mesh is directly taken as input in the $Loads(Karamba)$-component, where "Mesh Load" is selected as the type of load (see Fig. 4.27). "Global Proj." is checked off under "Orientation", which ensures that the load vector is orientated according to the global coordinate system. It also ensures that the area where the load is distributed is the projected area of each mesh face. Thus, the load on each face will vary due to the varying size of the projected area. This will replicate a more realistic snow distribution, than the one used for the grid shell model (see Fig. 4.14, (B)).



Snow Load

Snow Load (kN/m^2) -6.11

Orientation
○ local to mesh
○ global
● global proj.

Generation
● Point loads
● Line loads

Type of Load:
Mesh Load

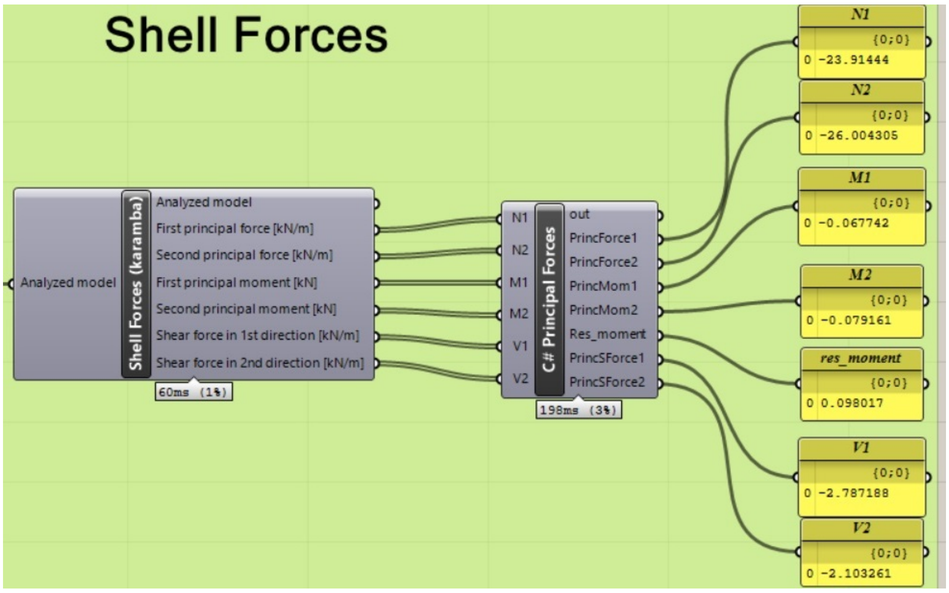
Figure 4.27: Grasshopper code for the globally projected snow load on the concrete shell, added to the points of the mesh as MeshLoad

4.3.5 Assemble the model and perform structural analysis with Karamba

Information about the shell element, cross section, material, loads and supports is collected in the $AssemblyModel$-component, before a first order analysis with the $AnalyzeThl$-component is performed. For the concrete shell, $ModelView(Karamba)$, $ShellView(Karamba)$ and $ResultVectorsonShell(Karamba)$ are used to visualise the results.

4.3.6 Retrieving Results

From the $ShellForces(karamba)$ the values of the principal normal forces and moments are collected. A C# script is made to find the maximum values (see Fig. 4.28 for the output of the C#-code). The displacements and reaction forces are collected in the same manner as the grid shell model (see 4.2.6).



```
N1
{0;0}
0 -23.91444
N2
{0;0}
0 -26.004305
M1
{0;0}
0 -0.067742
M2
{0;0}
0 -0.079161
res_moment
{0;0}
0 0.098017
V1
{0;0}
0 -2.787188
V2
{0;0}
0 -2.103261
```

Figure 4.28: $ShellForces(Karamba)$ and C#- component used to retrieve the principle forces in $[kN]$ and principal moments in $[kNm]$

4.4 Results from the concrete shell and grid shell analysis in Karamba

As mentioned in the introduction of this chapter, the idea is examining the differences of perfect shell behaviour (concrete) vs. reticulated bars (timber grid shell). In addition a shape analysis will be completed of the dome, barrel vault and plate. Their efficiency and response due to change in different geometrical parameters will be investigated.

To begin with a first order analysis will be performed with main focus on mass, displacements and normal forces. The span is set to 12 m and the height will vary to study the behaviour with a changing r/s-ratio. Self-weight and snow load is the loads considered. As mentioned in the previous sections, the snow load on the grid shell has a different distribution compared to the continuous concrete shell (see Sec. 4.2.4 and 4.3.4 and Fig. 4.14). However, this difference is considered to be negligible in the following analyses.

The analyses are carried out by the use of the Grasshopper codes defined in Sections 4.2 and 4.3 (See Appendix B for more details about the parametric codes). The results obtained from the structural analysis in Karamba are transferred to Excel, where the results are be presented in graphs and illustrations.

The materials selected and their properties are presented in Table 4.1.

Material	E $[MPa]$	G $[MPa]$	f_y $[MPa]$	γ $[kN/m^3]$	α_T $[1/°C]$
C30/37	33000	13750	20	25	$1.0E^{-}5$
GL32h	11800	400	32	4.3	$5.0E^{-}6$

Table 4.1: Material properties (concrete and timber) adopted in the Parametric Study

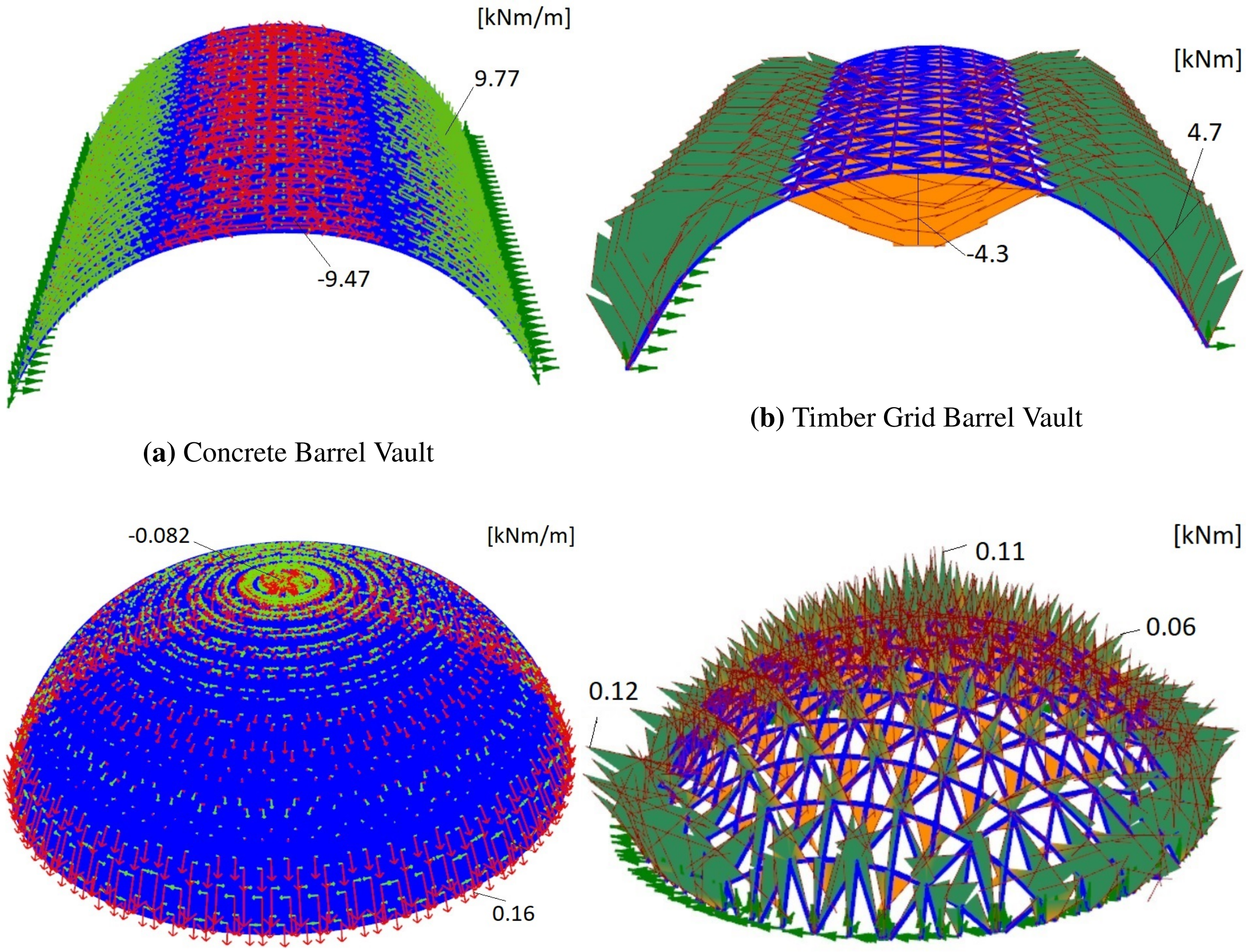
Timber is a light-weight material with a low density of $\gamma = 4.3kN/m^3$, compared to concrete where $\gamma = 25kN/m^3$. This, in addition to the stiffness of concrete being approximately 3 times as large as for timber (E-modulus, see Tab. 4.1), gives the materials different structural behaviours. Timber is a an anisotropic material and has a high capacity in both compression and tension in the grain direction but is weak perpendicular to the grain. Concrete is very good in compression, but requires reinforcement in order to handle tension. Timber has a more ductile failure mechanics in compression compared to concrete, i.e. it can handle a lot of deformation before breaking.

Despite the fact that the two materials are quite different, their behaviours in the configuration of shell structures is assumed to be comparable in the following sections.

4.4.1 Continuous concrete vs. reticulated timber

A general comparison is made of the two shells to gain an overall understanding of the structural behaviour of a continuous concrete shell and a reticulated timber shell. This will give an idea on how to further compare the two and correctly interpret the data obtained.

Figure 4.29 portrays the moment distribution for the dome and the barrel vault constructed with similar dimensions both as a concrete shell and a grid shell. For the concrete shells the moment distribution is illustrated with vectors (see Fig. 4.29a and 4.29c). The green arrows illustrate the first principle moment (M_1) in the global coordinates on the deformed structure, and the red illustrate the second principle moment (M_2). The bigger the arrows, the bigger the principle moments are at that point. For the grid shell, the moments are presented with regular moment diagrams (see Fig. 4.29b and 4.29d).



(a) Concrete Barrel Vault

(b) Timber Grid Barrel Vault

(c) Concrete Dome

(d) Timber Grid Dome

Figure 4.29: Moment distribution of the dome and the barrel vault both as concrete shell and grid shell

From Figure 4.29 is appears that the distribution of arrows in the concrete shell corresponds to the moment distribution in the timber grid shell. This indicates that the grid shell have a similar force distribution and bearing system as the concrete shell. Which again implies that the behaviour in the timber grid shell i.e. reticulated bars can be compared to the behaviour in a "perfect"/continuous shell.

To evaluate this statement further, the moment distribution and the displacement of the four structures in Figure 4.29 are presented and compared in the two following sections (Sec. 4.4.2 and 4.4.3).

4.4.2 Moment relative to r/s-ratio

Figures 4.30 and 4.31 show the maximum resulting moments M_{res} occurring in the concrete shell and grid shell, respectively, when shaped as both a dome (red) and a barrel vault (green). Both self-weight and snow load is applied to the structures.

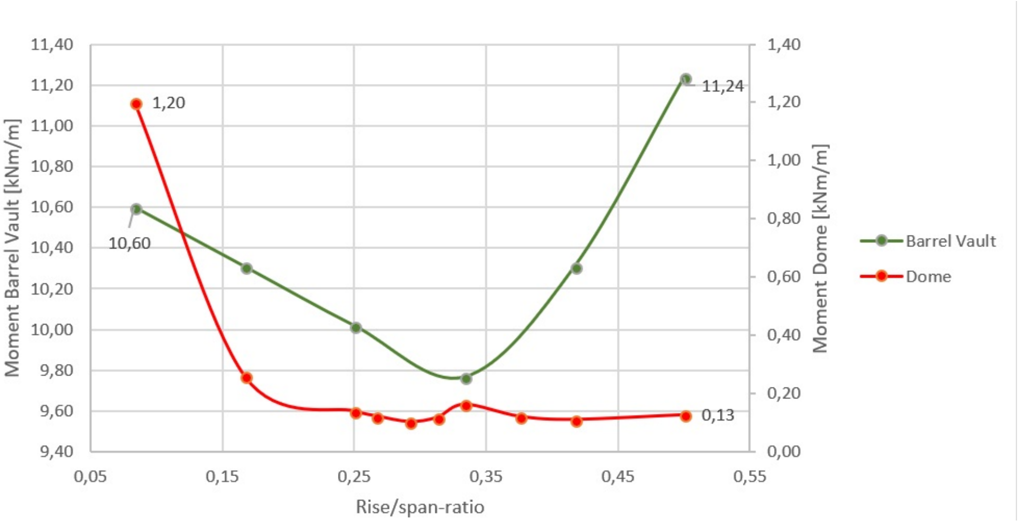


Figure 4.30: Concrete shell: Moment $M_{res} = \sqrt{M_1^2 + M_2^2}$ $[kNm/m]$ occurring in barrel vault and dome with changing r/s-ratios

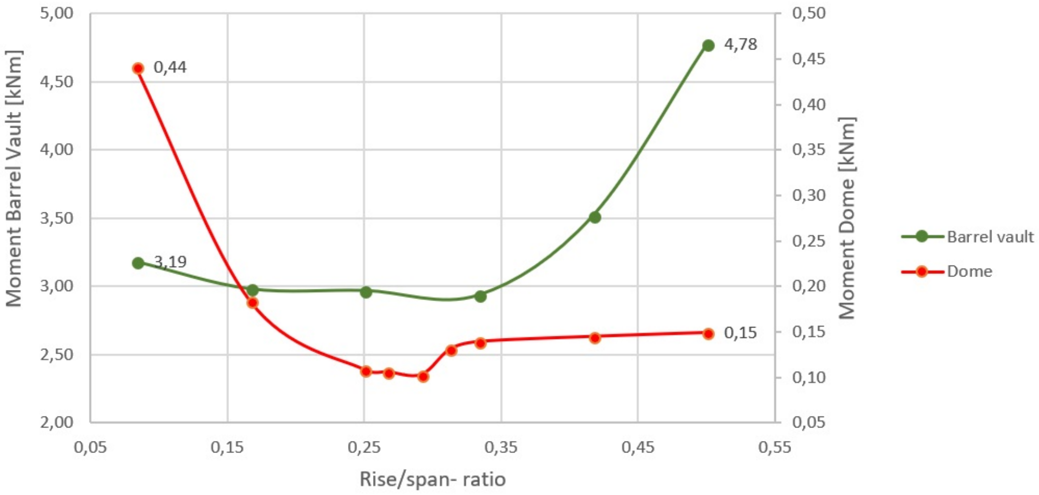


Figure 4.31: Grid shell: Moment $M_{res} = \sqrt{M_y^2 + M_z^2}$ $[kNm]$ occurring in barrel vault and dome with changing r/s-ratios

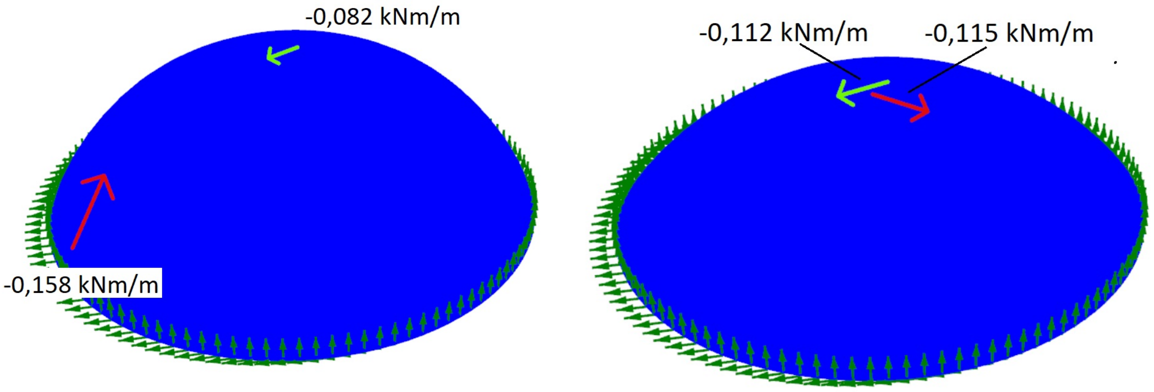
From the figures above it appears that the change in maximum moment with respect to r/s-ratio for the concrete shell-shapes (Fig. 4.30), resembles the change in moment for the grid shell-shapes (Fig. 4.31). In addition, the order of magnitude of the moment values for the concrete dome and the grid shell dome coincide, and similar observations can be done for the barrel vaults. This substantiates that the behaviour of the concrete shell corresponds to the grid shell behaviour.

Higher moments in the barrel vault

Notice that the left axis in the graphs presented in Figures 4.30 and 4.31 corresponds to the moment obtained in the barrel vault, and the right side corresponds to the moment in the domes. This is solely done to get a better observation of the values, as the acting moments in the barrel vault are considerably higher than in the dome.

Uneven increase in moments in the dome

From Figure 4.30 and 4.31 it can be observed that the maximum moment in the dome will have a uneven increase after rise/span - ratio 0.25. The dome will have a small peak around r/s-ratio 0.33. This peak is due to the redistribution of maximum moment.Figure 4.32 illustrates the maximum principle moments M_1 and M_2 with green and red arrow, respectively. The location of the maximum M_2 (red arrow), change as the r/s-ratio climb over 0.25.



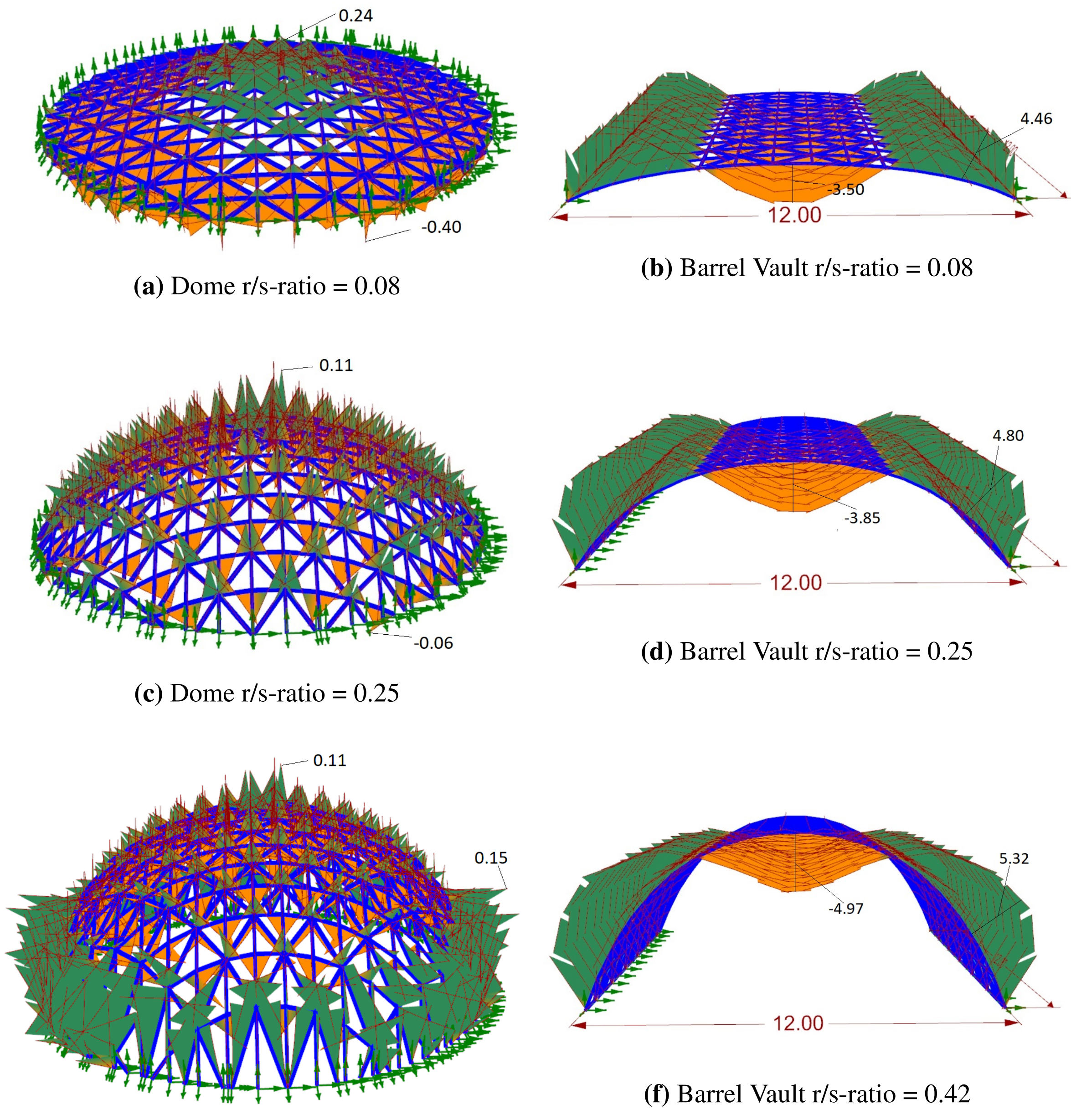
(a) r/s-ratio=0.25

(b) r/s-ratio=0.33

Figure 4.32: Location of maximum principle moment M_1 (green) and M_2(red) in kNm/m (illustrated with arrows) in the concrete dome

Higher relative change in moments for the dome

The maximum moment in the dome undergoes only modest changes when the r/s-ratio climbs above 0.2. Decreasing the r/s-ratio however ($rise/span \leq 0.2$) the dome will have a sudden increase in moment. The barrel vault will also experience an increase in moment when the r/s-ratio decreases below approximately 0.2, yet the *relative* increase in the dome will be notably higher. The dome will in general have a greater relative change in moment compared to the barrel vault, indicating that the change in r/s-ratio is of less significance for the barrel vault (see Fig. 4.33).



(a) Dome r/s-ratio = 0.08

(b) Barrel Vault r/s-ratio = 0.08

(c) Dome r/s-ratio = 0.25

(d) Barrel Vault r/s-ratio = 0.25

(e) Dome r/s-ratio = 0.42

(f) Barrel Vault r/s-ratio = 0.42

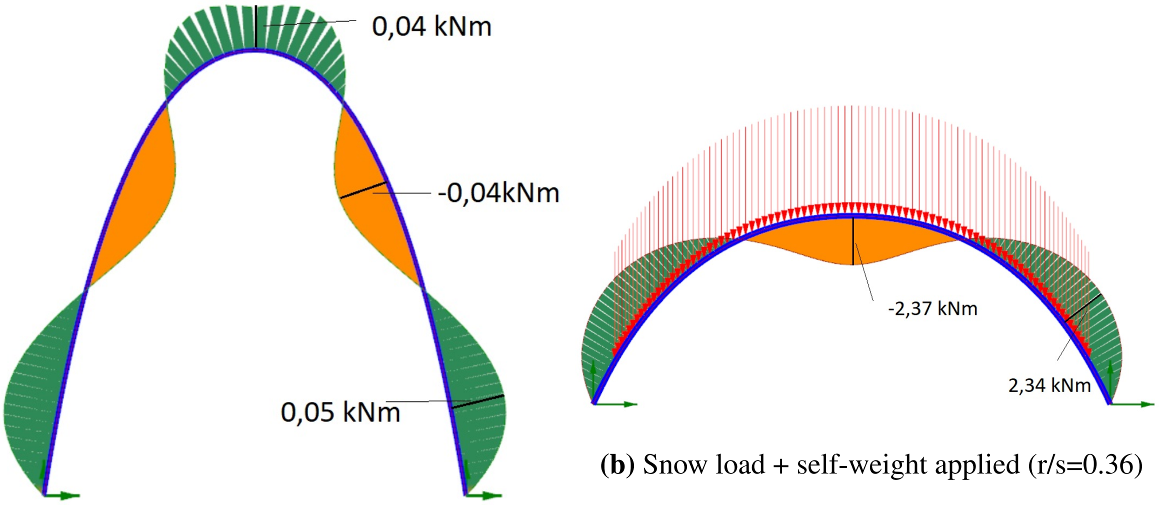
Figure 4.33: Moment distribution M_y $[kNm]$ for the grid shell dome and barrel vault for changing r/s-ratios

The barrel vault will have a smoother and more predictable change in maximum moment as the r/s-ratio increase. Figure 4.33 illustrates how the moment distribution for the dome and barrel vault develops when the r/s-ratio is changing. The figure indicates how the dome will adjust its moment distribution according to the curvature more than the barrel vault.

Optimal r/s-ratio for arch-like structures exerted to different load situations
As previously mentioned in Section 2.5 a barrel vault and a dome can both be defined by an arch. It could therefore be interesting to see what the optimal r/s-ratio is for an arch, and compare it to the values presented in Figures 4.30 and 4.31.

The arch is modelled as a concrete arch in GH, discretised into 100 beam elements and applied loads in the end-nodes of the beam elements. Galapagos is used to minimise the moment by keeping the span fixed and changing the height, to seek for the shape which is (approximately) funicular for the load situation in question. Figure 4.34 shows the two various shapes obtained when minimising the moment for only self-weight applied (see Fig. 4.34a) and both self-weight and snow load (see Fig. 4.34b).

With only self-weight applied the concrete arch is found to have an optimal ratio around 1.05, while for the second case with snow load + self weight the concrete arch had an optimal ratio equal to 0.36. The variance between these two ratios display how the optimal shape will adjust to the load situation. Referring to the latter case, the optimal r/s-ratio is comparable with the results presented in Figures 4.30 and 4.31 (also self-weight + snow load applied). This substantiate that the behaviour of the arch can relate to the behaviour of the barrel vault and the dome.



(b) Snow load + self-weight applied (r/s=0.36)

(a) Only self-weight applied (r/s=1.05)

Figure 4.34: Illustration of the optimal arch shape obtained with Galapagos with regards to minimising moments. The arch is modelled in concrete and discretised into 100 beam elements. The two shapes are results of two different load situations.

4.4.3 Displacement relative to r/s-ratio

Figures 4.35 and 4.36 below illustrate the largest displacement for the dome and barrel vault modelled as a concrete shell and grid shell for different r/s-ratios. The displacement due to self-weight and snow load is considered, and the maximum total displacement in $[mm]$ (see Eq. (4.1)) is presented in the graphs.

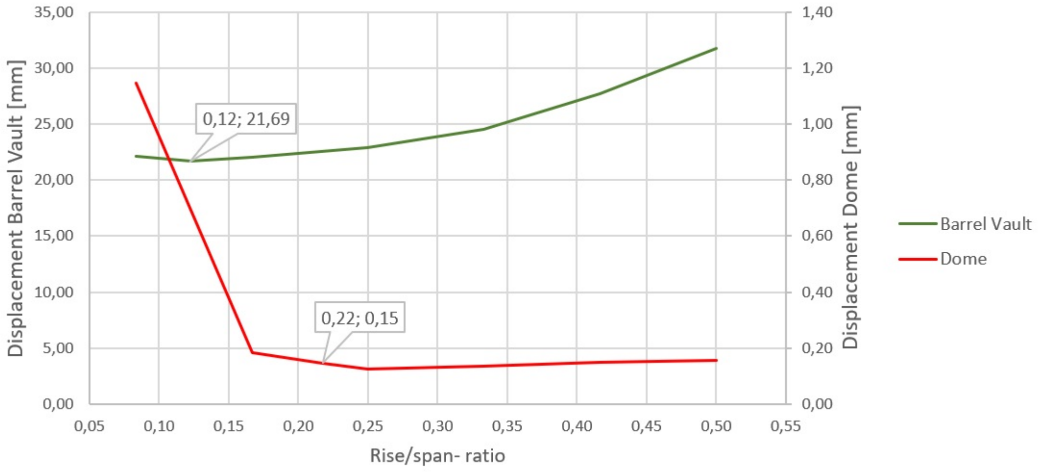


Figure 4.35: Concrete shell: Total displacement $[mm]$of the barrel vault and dome with changing r/s-ratios

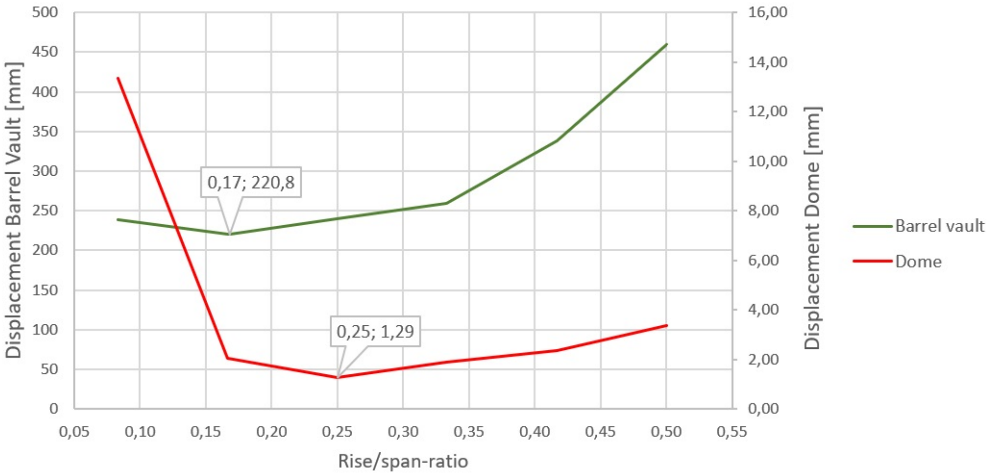


Figure 4.36: Grid shell: Total displacement $[mm]$ of the barrel vault and dome with changing r/s-ratios

Analogous to Figures 4.30 and 4.31, both Figures 4.35 and 4.36 show how the displacements of the concrete barrel vault and dome resemble the displacements of the reticulated dome and barrel vault. This underlines once again that the concrete shell and grid shell have similar behaviours.

In Figures 4.35 and 4.36 the lowest displacement found for all configurations are highlighted. The Galapagos evolutionary solver in GH was used to minimise the displacement and the corresponding r/s-ratio.

Lower displacements in the dome

Figure 4.36 illustrates the maximum total displacement for the grid shell dome and barrel vault. The barrel vault will have a considerably higher displacement compared to the dome, and the displacement is also more affected by the change in height in comparison with the dome. Between the r/s-ratios 0.12 and 0.5, the barrel vault will experience an increase in displacement of 48.4% compared to the dome where the increase is 28.6% for the same interval. Due to the barrel vault having larger values of displacements, the relative increase will influence the barrel vault structure considerably more than the dome structure. This indicates that the dome is a more efficient structure than the barrel vault.

Developable and non-developable surface

The dome is simply supported around the entire edge of the surface, contrary to the barrel vault which only has supports on two of its four edges. In addition, the dome is doubly curved and a non-developable structure compared to the single curvature and developable barrel vault (see Sec. 2.2). Referring to Section 2.5, the dome is more rigid and a more stable structure due to its closed ring at the bottom.

4.4.4 The effect of curvature

The shell is defined by a curved surface (see Sec. 2.1). Section 2.3 *Membrane forces* explained how the curvature is the reason for why the shell can carry vertical loading with only in-plane forces. It was also explained how it is easier to deform the structure by bending it, than by stretching it. Figure 4.37 illustrates the concrete model, with the displacement of the plate, barrel vault and dome for an increasing span (with height equal to $5m$ for the two latter). The figure illustrates how curvature clearly is an important factor in withstanding deformation. A plate supported on all sides will only experience bending forces when vertical load is applied, however no compression forces. Thus, the plate's ability to withstand the deformation only depends on its bending resistance. Contrary to a structure with a curvature, like the barrel vault and the dome, where the in-plane compressive forces will contribute in resisting the deformation (see Fig. 4.37).

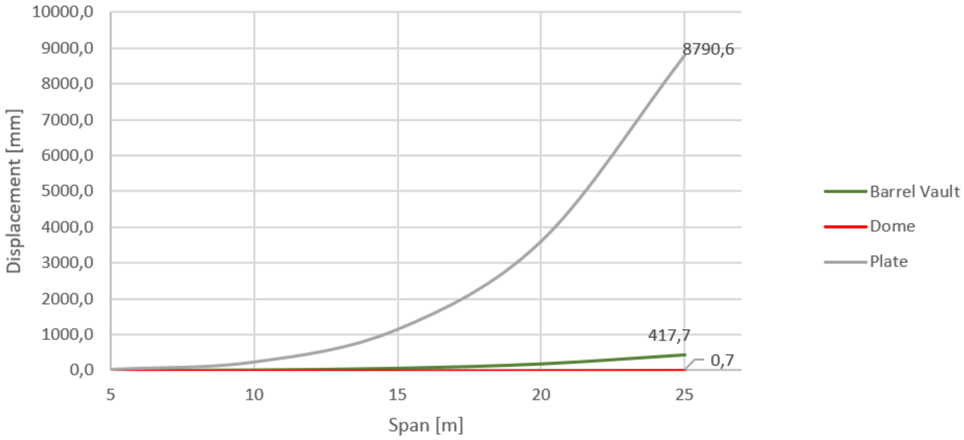


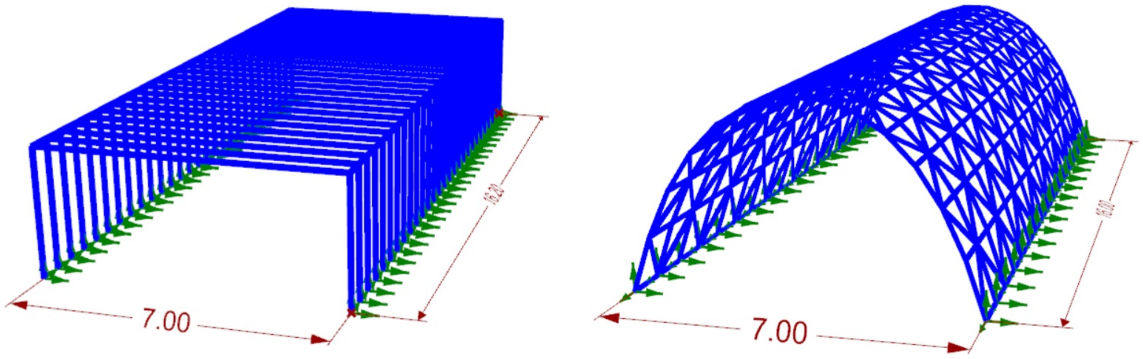
Figure 4.37: Total displacement $[mm]$ of the concrete model, modelled as a plate, barrel vault and dome for varying span. The height is fixed to $5m$ for the dome and barrel vault.

To highlight this further; the displacements of the barrel vault and the dome when laid out flat (height= 0, illustrating a plate) were found to be $1614mm$ and $484.2mm$, respectively. By increasing the height to 1 meter, the displacement seems to decrease with 98.6% for the dome and 99.8% for the barrel vault. In addition the moment will decrease greatly. This shows how the curvature plays an important part in the efficiency of shells.

Traditional flat roof vs. barrel vault

Traditionally, roof structures will be built with simple geometric shapes like mono-or duo-pitched roofs. Hence, to emphasize the efficiency of the less traditional shell structure, a comparison is made between a flat roof (mono-pitched with zero angle) and a barrel vault. The flat roof structure consists of rafters supported on columns c/c $600mm$. The roof is covering approximately the same floor area for the two structures. The total length of timber material used is $696m$ for the barrel vault and $625m$ for the flat roof. Figure 4.38 shows the two structures in question.

The necessary cross section for the two structures is found by limiting the displacement to be equal to $5cm$. The corresponding weight of the structures is also found, and both cross sections and weights are presented in Table 4.2.



(a) Traditional Flat Roof c/c $600mm$

(b) Timber Grid Barrel Vault

Figure 4.38: Illustration of a traditional flat roof structure and a barrel vault grid shell with approximately the same floor area and size of structure. The size of cross section is chosen with regards to keeping the $displacement = 5cm$.

Table 4.2: Cross section and weight for the barrel vault and the traditional flat roof, given a $displ. = 5cm$

	$displacement = 5cm$		
Shape	Cross section $[mmxmm]$	Weight $[kg]$	tot. length of beams $[m]$
Barrel Vault	90 x 90	2424	696
Flat Roof w/col	225 x 115	6952	625

As seen from Table 4.2 the necessary cross section to keep the displacement at approximately $5cm$ is greater for the flat roof than for the barrel vault. The total mass of the flat roof structure will thus also be larger. This illustrates why timber grid shells and shell structures in general can be characterized as being *light-weight structures*. The shape of the structure makes it more efficient and the necessary material (and thus the material costs) will be lower than for the traditional roof structures.

Even though curved structures are found to have benefits with respect to deformation and material use, there are still some concerns that need to be taken. As mentioned in Section 4.4.2, large moments were found for low r/s-ratios, which in combination with compression forces and slender cross sections poses a threat to the stability of the structure. This will be further addressed in Section 4.5.

4.4.5 Normal forces in grid shell

The largest axial forces of all beam elements in the grid shell when both self-weight and snow load is applied are presented in Figure 4.39 (compression) and Figure 4.40 (tension).

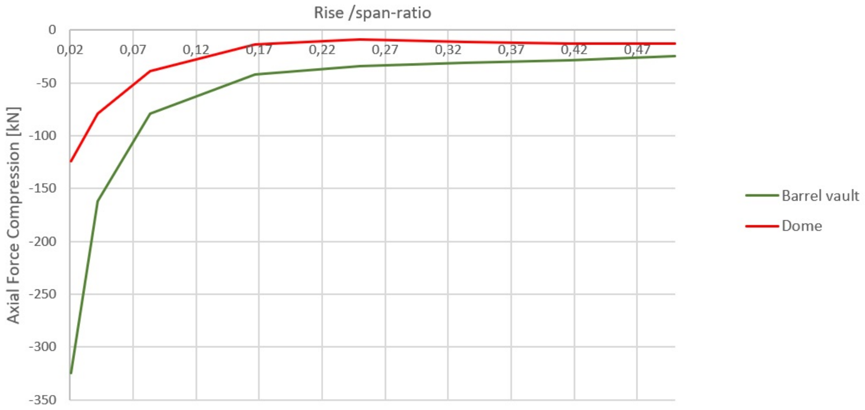


Figure 4.39: Largest compression force N_c $[kN]$ of all beam elements in the grid shell dome and barrel vault relative to r/s-ratio

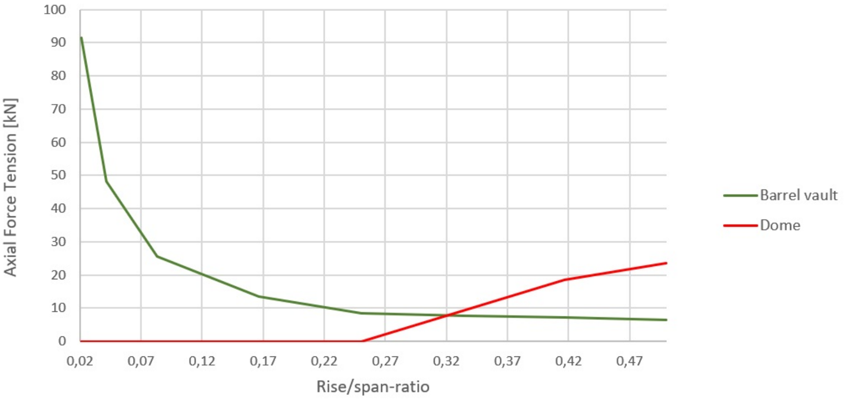


Figure 4.40: Largest tension force N_t $[kN]$ of all beam elements in the grid shell dome and barrel vault relative to r/s-ratio

As seen from Figure 4.40 the dome will not have any beam elements in tension before reaching a r/s-ratio above 0.25. The forces are able to flow down to the ground in all directions due to the the curvature of the surface and the supports placed continuously around the edges of the surface. The loads will be evenly transferred to all the beam elements. For the singly curved barrel vault, on the other hand, the forces can only be transferred to the ground on two of the sides (where there are supports). The barrel vault will thus experience a higher stress concentration near the supports.

4.4.6 Axial stresses in grid shell

As seen in Figure 4.40 above, the dome showed to not have normal forces acting in tension in any of the beam elements before a r/s-ratio above 0.25. The barrel vault however, showed to have tension members for all r/s-ratios. To investigate this difference further the axial stresses were found for the barrel vault and dome at a r/s-ratio of 0.08.

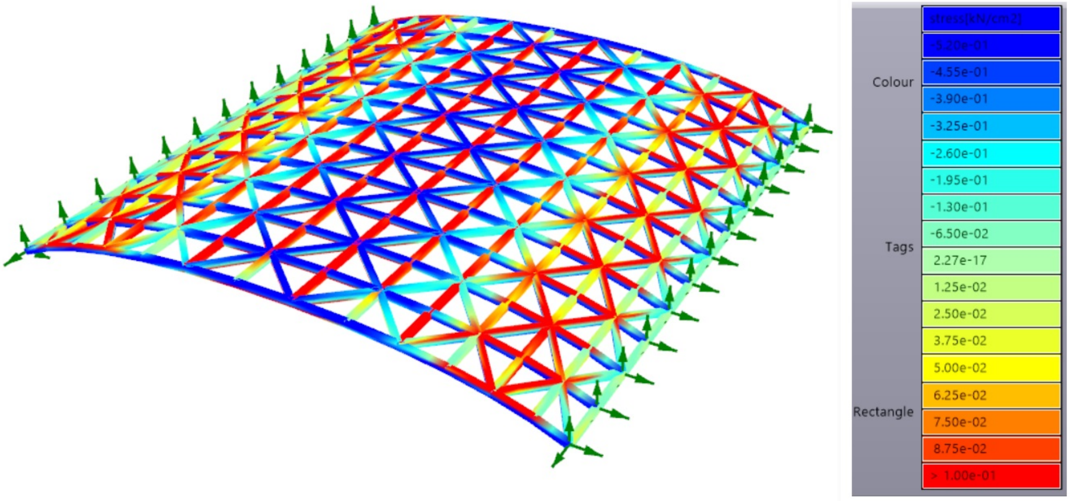


Figure 4.41: Axial stresses in a grid shell barrel vault with r/s-ratio of 0.08. Red colour indicate tension, and blue indicate compression axial stresses

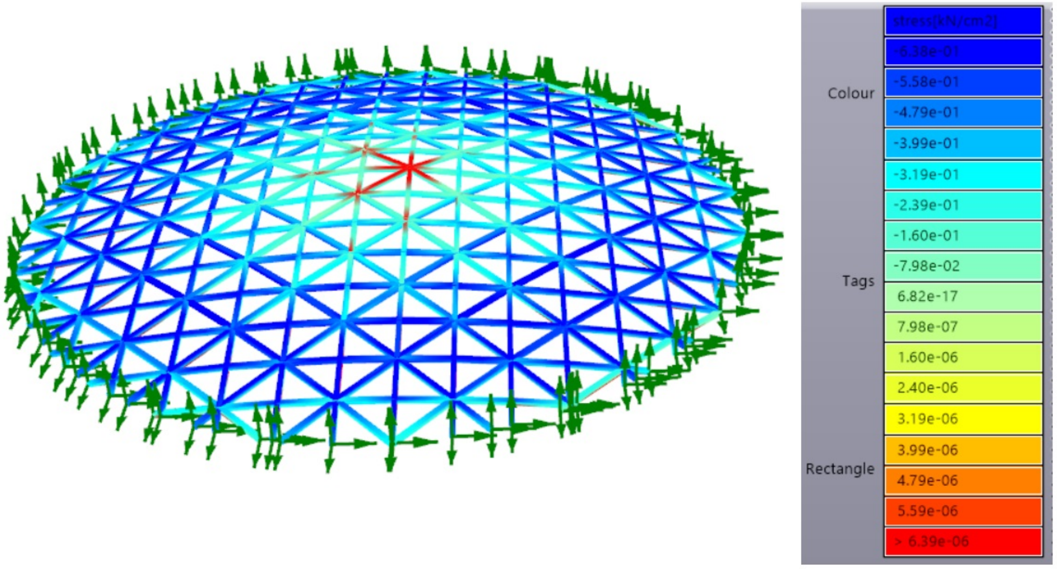


Figure 4.42: Axial stresses for a grid shell dome with r/s-ratio of 0.08. Red colour indicate tension, and blue indicate compression axial stresses

Figures 4.41 and 4.42 shows that the distribution of stresses will be quite different for the two shapes (same r/s-ratio). The barrel vault in Figure 4.41 will experience more tension stresses, as indicated with the red and yellow colours. Most of the red area will be close to the support and in beam elements located parallel to the supports. This illustrates the disadvantage of force transfer in the barrel vault (only one direction) - as the beams parallel with the supports will have to transfer the loads to the ground orthogonal to their neutral axis. The dome in Figure 4.42 will mostly experience compressive stresses. As mentioned above in Section 4.4.5, the shape of the dome enables the forces to flow to the ground in all direction, which distributes the loads evenly to all element.

At the top of the dome, the bending moment causes tension in the upper fibers of the cross section. In the dark blue area the bending moment will cause compression in the upper fibers and tension in the lower (this tension area appears if the dome is seen from below). Note that the magnitude of the stresses in the barrel vault is much greater than in the dome.

To conclude this, the dome will have a more efficient stress distribution than the barrel vault for a low r/s-ratio.

To investigate how the axial stress distribution of the dome changes with a changing curvature, two additional r/s-ratios of the dome were investigated. Figures 4.43 and 4.44 illustrate the axial stress distribution in the dome when the r/s-ratio is 0.25 and 0.42, respectively.

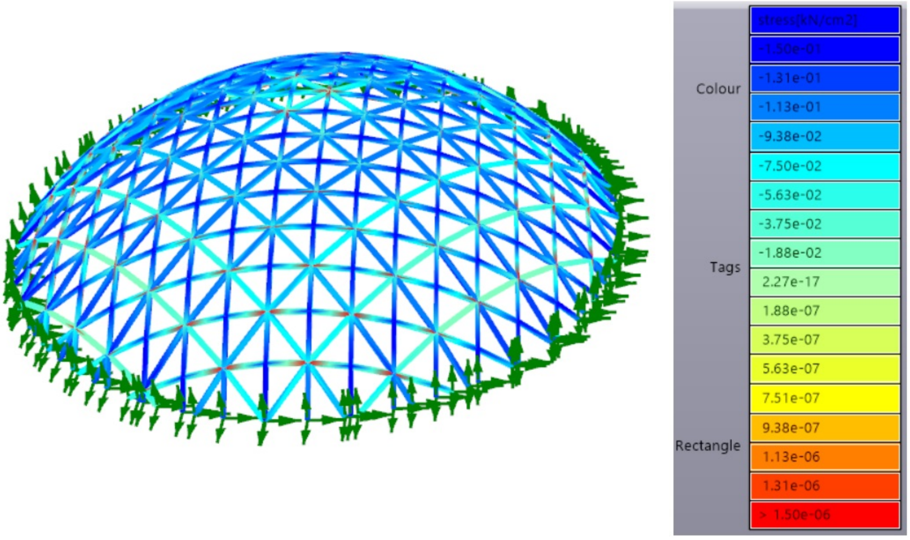


Figure 4.43: Axial stresses for a grid shell dome with r/s-ratio of 0.25. Red colour indicate tension, and blue indicate compression axial stresses

From Figure 4.43 it can be observed how most of the dome will be in compression, as indicated with the blue colours. The colours are mainly in the same colour shade, i.e. the stresses are evenly distributed. This is favourable as all the beam elements contribute equally in carrying the load, and thus avoiding that one element fail ahead of the other beam elements.

When the rise/span ratio increases further, the members close to the support will start acting in tension (see Figure 4.44). Due to the configuration of the model in Grasshopper, the length of these bean elements will also be longer and thus more exposed to high moments and axial forces. However, a higher curvature will cause the dome to deform differently, regardless of the length of the beam elements. As seen from Figure 4.45 the dome with the higher r/s-ratio

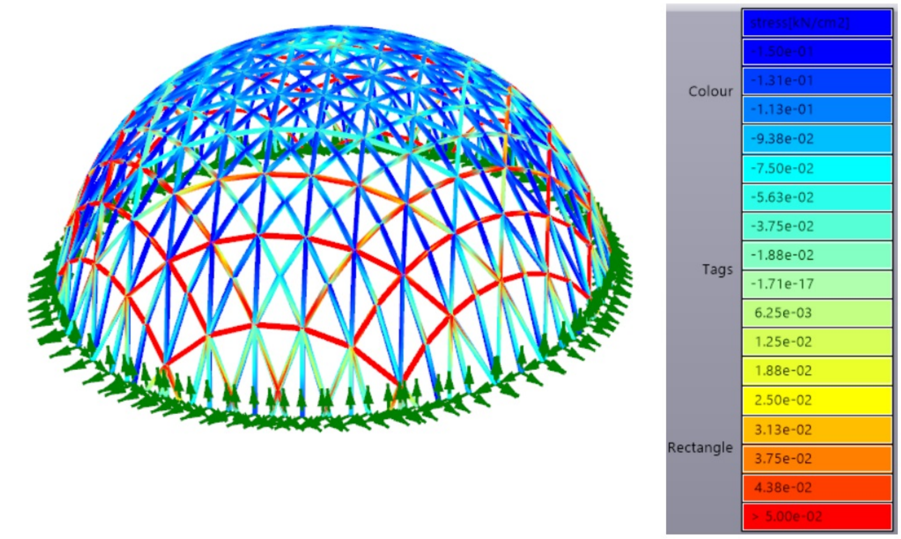
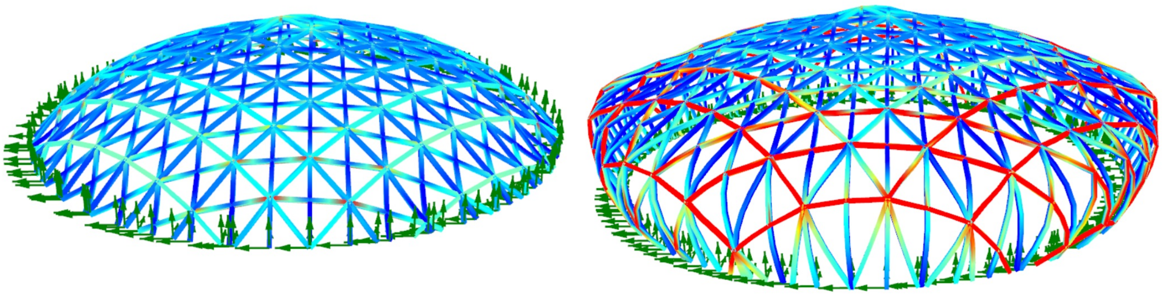


Figure 4.44: Axial stresses for a grid shell dome with r/s-ratio of 0.42. Red colour indicate tension, and blue indicate compression axial stresses

will tend to deform by bending outwards, compared to a more inwards deformation for a lower r/s-ratio. This implies that as the curvature gets two steep, the flow of forces will be less smooth and the structure will be more prone to higher deformations and axial stresses in tension.



(a) Rise/span ratio: 0.25

(b) Rise/span ratio: 0.42

Figure 4.45: The different types of deformation for grid shell dome with two different r/s-ratios. Inwards deformation for low r/s-ratio and outwards deformation for high r/s-ratio. Scale: 500:1

4.4.7 Kicking forces

For structures with low curvature i.e. a low r/s-ratio, *kicking forces* needs to be taken into account (explained in Sec. 2.5). Figure 4.46 shows the reaction force R which is the force needed to be transferred to the ground. The arch on the left has a low curvature which will give rise to a large horizontal force R_y, the kicking force. For a high curvature arch (the right illustration in Fig. 4.46) the vertical component R_z from the reaction force R is the largest, hence the forces from the structure will easily be transferred to the ground mainly from the weight of the structure.

Figure 4.47 illustrates the magnitude of the kicking force for different r/s-configurations for grid shell shaped as a barrel vault and a dome. The kicking force is determined by considering

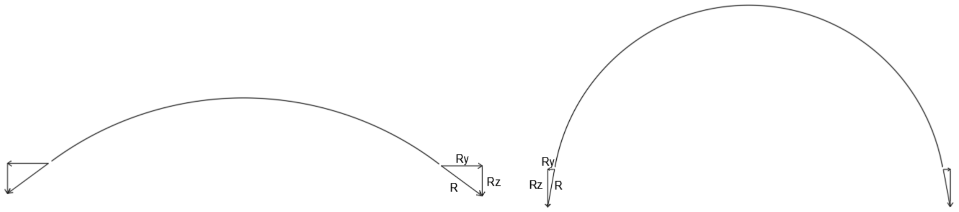


Figure 4.46: Kicking forces (Ry) for low and high r/s-ratio (low and high curvature arches)

the resultant of the x- and y-component in the reaction forces obtained in Karamba. The kicking forces are presented as the force per meter support.

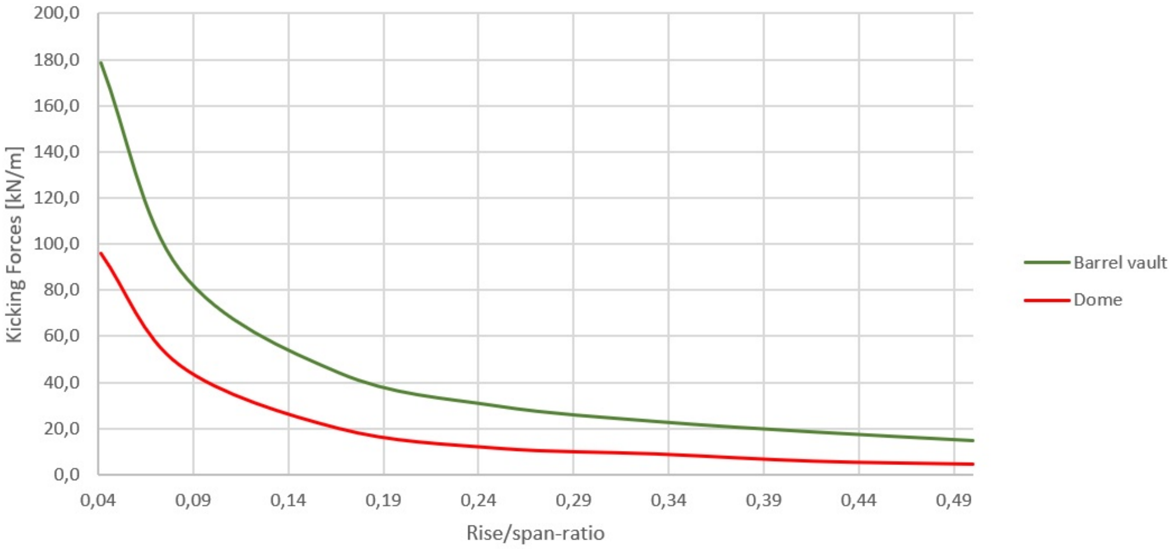


Figure 4.47: Kicking forces (Horizontal reaction force in the plane) for grid shell barrel vault and dome with regards to changing r/s-ratio. Kicking Force R = $\sqrt{R_x^2 + R_y^2}$

The kicking forces become activated when the curvature increases from height equal to zero (where zero height is equivalent with a plate i.e. zero normal forces, and $R_y = 0$). As seen from Figure 4.47, the values decrease exponentially when the ratio, i.e the height of the structure, continue to increase. It can be observed how the kicking forces for the barrel vault is larger than the kicking forces in the dome. This coincide with the concept discussed in Section 2.5. Here it was explained how the dome is less dependent on the external support due to the ability of supporting forces directed outwards from the structure with tension rings.

4.5 Buckling

As mentioned in Section 2.5 (and repeated in Section 4.4.4), shell structures are prone to buckle as they in general are slender structures with large compression forces. Axial forces in beams and trusses as well as in-plane forces in shells influence the stiffness of the elements in second order theory calculations. Tension makes them stiffer and increase their bending stiffness, compression has a softening effect. Slender columns or thin shells may fail due to buckling before the stresses in the cross section reach the material strength. Stability analysis therefore plays an important role in structural design.

Karamba allows for considering second order theory (Th.II) via the $AnalyzeThII$ -component. It is based on small displacements and takes account of axial forces via the element's geometric stiffness matrix, $\mathbf{K_G}$ [42].

The buckling analysis is based on the linear eigenvalue problem:

$$[\mathbf{K} + \lambda\mathbf{K_G}]\mathbf{x} = \mathbf{0}, \tag{4.2}$$

where $\mathbf{K}$ is the stiffness matrix of the element, $\mathbf{K_G}$ is the geometric stiffness matrix, λ is a scalar (the eigen-value), and $\mathbf{x}$ is the eigenvector [26].

The buckling load factor is the output of the 2nd order analysis in Karamba, and is equivalent with the eigenvalue λ, obtained by expression (4.2). The buckling factor reveals if the structure is safe or not, and safety is ensured if the following expression holds:

$$\lambda = \frac{N_{crit}}{N^{II}} \geq 1.0,$$

where the expression shows how λ is the ratio between the normal force acting within the structure (N^{II}) and the normal force at which the structure will fail due to buckling (N_{crit}).

Buckling analysis of the barrel vault

The buckling load factor was found for the grid shell barrel vault and the concrete barrel vault. Comparing the buckling factor for two different materials is not ideal. However, as one of the objectives of this chapter is comparing a reticulated shell with a continuous shell, a comparison of timber and concrete is difficult to avoid - as these are the most popular materials to use for such structures.

The displacement was set to $5cm$, and the cross section of the beam elements and the thickness of the shell was changed accordingly. Figure 4.48 shows the BLFacs for the two shells.

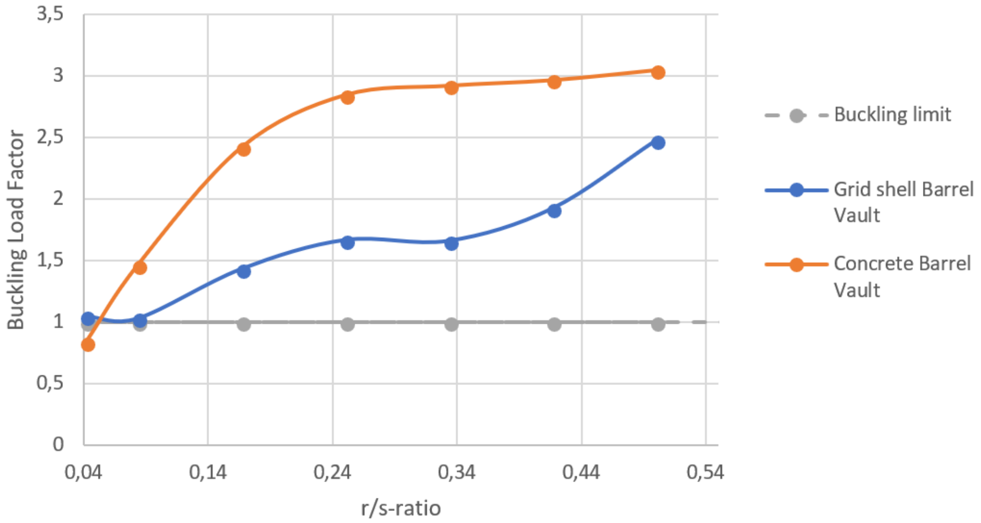


Figure 4.48: Buckling load factor for a barrel vault with a change in r/s-ratio (concrete shell and grid shell). Cross section and thickness chosen in accordance with reaching a displacement of the barrel vault of $5cm$.

Different stiffnesses of the two shells

The solid concrete vault will be stiffer than a reticulated grid shell. The buckling factor, λ, depends on the stiffness of the system, where a higher stiffness gives a higher buckling safety. This is illustrated in Figure 4.48 where the concrete shell displays higher buckling factors than the timber grid shell.

BLFac relative to mass of the structures

The concrete shell has a very high mass compared to the timber grid (see Fig. 4.49), and the mass of the structure will in general have a positive influence on the buckling safety. More mass will normally indicate more material and higher moment of inertia, i.e. higher stiffness of the structure. However note that this is only the case if the mass is located efficiently (bigger cross section rather than longer beams).

Given the significance of the mass the relative value between the BLFac and the mass of the structure is found (see Fig. 4.50).

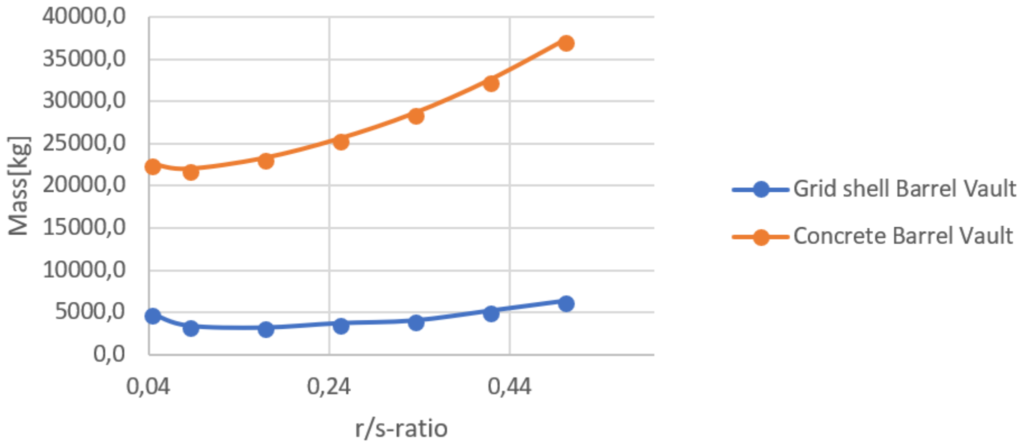


Figure 4.49: Mass $[kg]$ of the concrete and grid shell barrel vault for a changing r/s-ratio ($displ. = 5cm$)

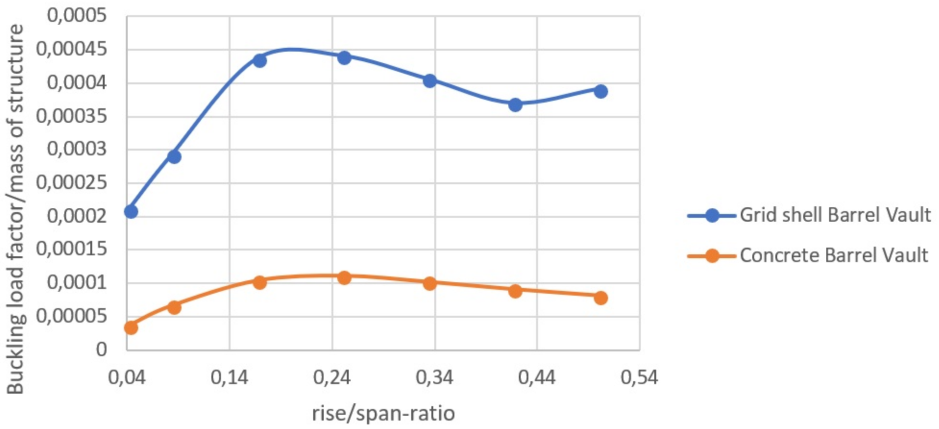


Figure 4.50: Buckling load factor relative to mass of the structures (concrete and grid shell barrel vault, $displ. = 5cm$).

Figure 4.50 indicates that the timber grid shell is in fact a more efficient structure as less material is needed to keep the structure safe from buckling. This indicates how the mass in the reticulated shell is exploited in a better way than in the concrete shell.

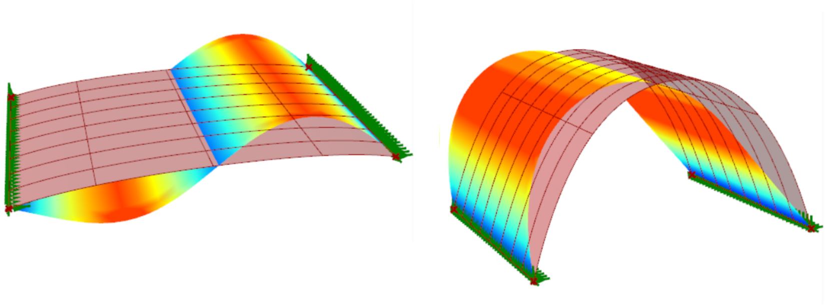


Figure 4.51: Buckling shapes for a concrete barrel vault with low (left) and high (right) r/s-ratios. S-shape for low r/s-ratio, sway-shape for high r/s-ratio

The buckling deformation shown in Figure 4.51 the figure is the first buckling mode, which is the mode for when the structure will first buckle (at the lowest load). With an increasing r/s-ratio the buckling shape will move from being vertical to horizontal, from an s-shape to a sway shape. The first mode tend to always be asymmetric, indicating that symmetric buckling will demand higher forces.

Buckling of the concrete and grid shell dome

It was also done a short buckling check of the dome, and the BLFacs were found to be very large. Thus, the dome was considered as safe and not further investigated.

4.6 Summary of the Parametric Study

In this chapter different shells, constructed both as a reticulated shell in timber and a concrete shell shapes have been studied. It is emphasized that the comparison is done based on their structural behaviour for a changing r/s-ratio rather than their point-wise values. It is understood that the materials used each have unique properties and behaviours, and will demand separate designing procedures. The Parametric Study in this chapter however, does not cover the design of the structures.

In the beginning of the analysis it was disclosed that a reticulated shell can be treated as a shell, equivalent to the continuous concrete shell. Their behaviours regarding displacement and moments were found to be quite similar.

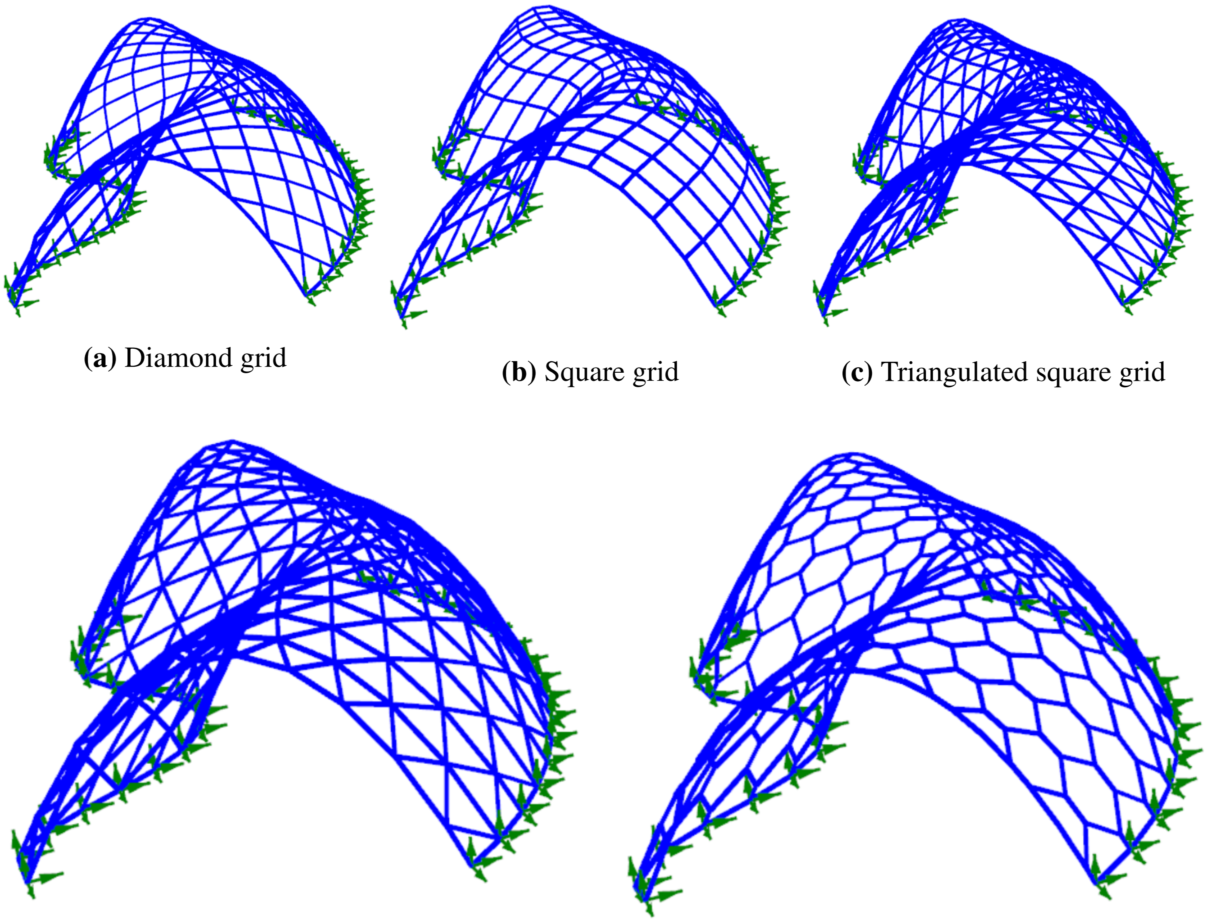
The analyses showed that a r/s-ratio between 0.2 and 0.4 was in general the most optimal (for both barrel vault and dome). Too low ratios led to high moments, normal forces, kicking forces and displacements, in addition to low buckling factors for barrel vault. For high ratios the structure became less stable with larger displacements, axial stresses and moments.

It was observed how the dome is a very efficient structure. The reasons being the double curvature and the smooth flow of forces, which is the main features of a great shell structure. The barrel vault with its singly curved surface, was found to have higher moments, displacements and normal forces compared to the dome, indicating a less stable structure.

A comparison was done between a plate, barrel vault and dome, to show the effect of curvature. A large difference was found for only a small increase in the height, stating how a curved shape is beneficial as it decreases both the occurring moments and displacements.

This demonstrates how it is possible to reach great spans for curved shell structures, whereas typically shorter spans and heavier structures are seen for the traditional buildings with beams and columns.

5 | Grid patterns on freeform structures



(**a**) Diamond grid

(**b**) Square grid

(**c**) Triangulated square grid

(**d**) Triangle grid

(**e**) Hexagon grid

Figure 5.1: Different grid patterns for a freeform timber grid shell

Objective of the grid pattern analysis
In Chapter 4, it was depicted how the shell behaviour depends on both the shape and the type of shell (continuous vs. reticulated). The aim of this chapter, is to further investigate the reticulated shell, and move beyond the shapes of regular geometry to freeform structures (see definition of *Freeform* in Sec. 2.2 *Shells-Classification*).

In a grid shell the forces will be transferred to the beams and down into the ground. Hence, the grid's shape and orientation will influence how well the forces flow though the structure. The idea of this chapter is to start out with an arbitrary shape of the grid shell, and explore its behaviour for different patterns of the reticulated bars. The grid patterns analysed are presented in Figure 5.1.

To draw the main advantage of a continuous shell i.e force flow in all directions, the arbitrary shape of the grid shell will also be presented as an equivalent concrete shell. The principle stress lines will be considered, as they give an indication on what would be the optimal grid pattern topology. However, this is only included for a comparison to the geometric patterns in Figure 5.1 (the grid pattern topology will not be modelled/adopted).

5.1 Creating grid patterns

To compare the grids of interest, the geometry of the patterns shown in Figure 5.1 is defined in Grasshopper. We adopt the Grasshopper code explained in Section 4.2, but replace the part of creating beam elements (Sec. 4.2.1) with grid patterns established by the use of the Grasshopper plug-in LunchBox. Figure 5.2 illustrates how the square grid (Fig. 5.1b) was established by the use of LunchBox.

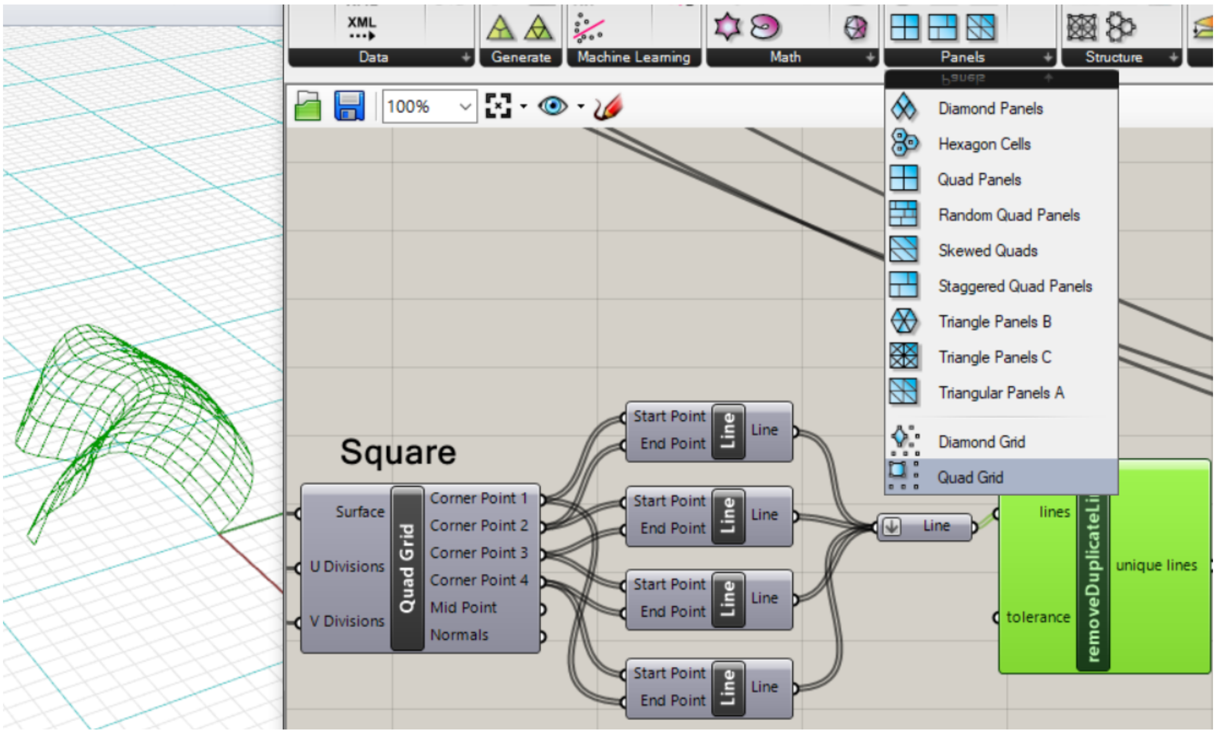


Figure 5.2: Establishing the grid patterns with the Grasshopper plug-in LunchBox. The example here shows the square pattern presented in Fig. 5.1, created with the $QuadGrid$-component in Lunchbox.

The squared grid pattern is created by inputting a freeform-surface and a uv-domain to divide the surface. The output is four corner points defining the location of each square in the grid. Lines are created by the $Line$-component connecting the start- and end-points of the four corners. The $removeDuplicateLines$-component is added to remove any excess lines.

This procedure is done for all the grid patterns in Figure 5.1. The uv-domain are chosen individually for the different grids to pursue nice and even geometries with approximately $1m$ length of beam elements. The output-lines generated for each pattern, are transferred to the $LineToBeam$-component in Karamba to create the timber beams for the structural model (see Appendix C for more details about the code).

5.2 Grid pattern analysis

The comparison is made on behalf of displacements, normal forces and moments occurring for the same load situation. Both snow load and self-weight will be considered in this analysis. The aim is to understand which geometric pattern gives the best structural performance with respect to the abovementioned.

It will also be highlighted how much timber is used to generate the different patterns, as this may vary with the geometry. This will be an important factor with respect to minimising the use of material for better costs (given the structural performance is sufficient). A simple investigation will also be done with respect to the spacing length in between the beams.

An arbitrary cross section and snow load was assigned to the grid shells. A first order analysis was performed with Karamba, and the values obtained for the grids are presented in Figures 5.3-5.5.

Normal forces in the grid shell for different grid patterns

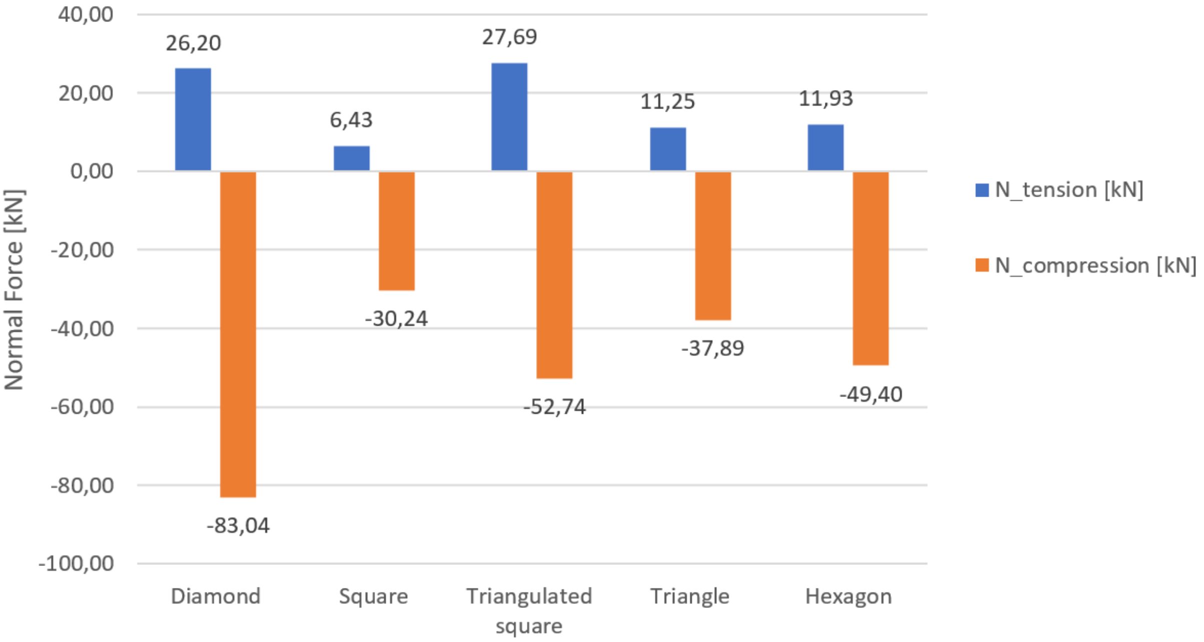


Figure 5.3: Normal forces $[kN]$ for the different grid patterns presented in Fig. 5.1

Figure 5.3 below illustrates the normal forces occurring in the grid shells (presented in Fig. 5.1). The maximum values $N_{compression}$ and $N_{tension}$ represent the extreme values found for the grids, where the negative value is compression (orange) and the positive is tension (blue).

From the figure it can be seen that the highest compression force is found for the diamond pattern. The triangulated square grid has the highest tension force. The square pattern and the triangle seem to have the smallest tension and compression forces and thus the lowest range between the maximum compression and tension.

Moments in the grid shell for different grid patterns

Figure 5.4 presents the moments occurring in the grids, where M_{max} and M_{avg} represent the maximum and average value of all moments in the reticulated shells. The maximum moment in $[kNm]$ is considered as:

$$M_{max} = Max(M_{y,max}, M_{z,max}),$$

and similarly for the average moment M_{avg}.

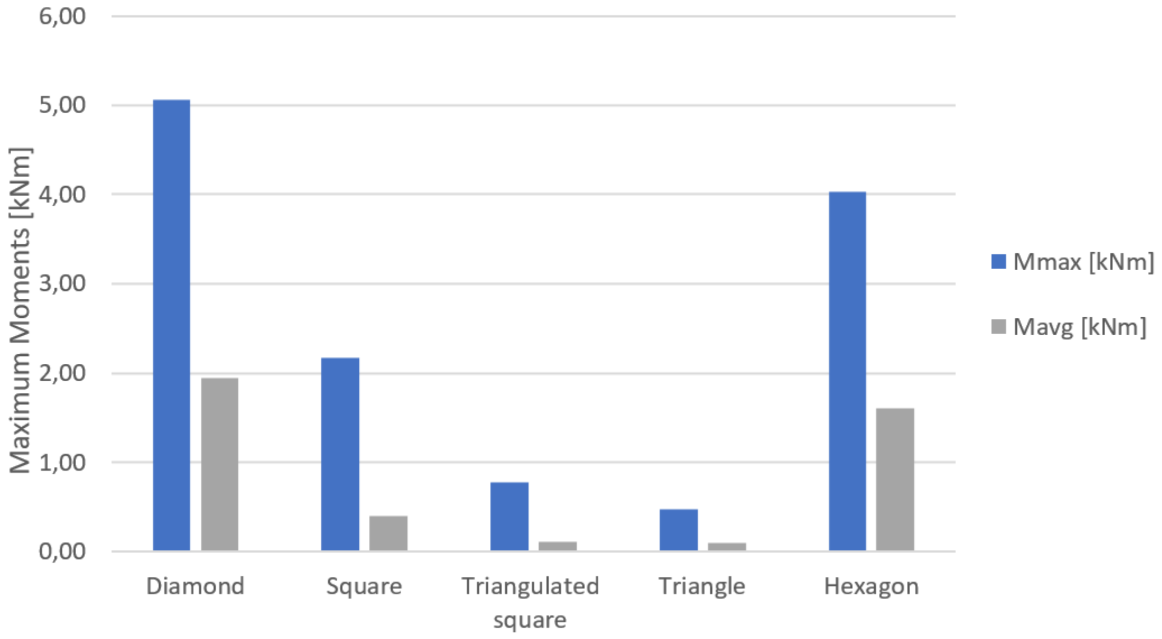


Figure 5.4: Moments (M_{max} and M_{avg}) $[kNm]$ for the different grid patterns presented in Fig. 5.1

The moments are found to be fairly low for all grids, which could suggest that the shape investigated is a decent shape for a shell structure. The grids presented on the far left and right in Figure 5.4, namely the diamond grid and the hexagon, show the largest moments. The grid pattern made of triangles (see Fig. 5.1d) show the smallest maximum and average moments with values $M_{max} = 0,47kNm$ and $M_{avg} = 0,09kNm$.

Displacement and amount of material used to create the grid shell for different grid patterns

Maximum displacement and total length of material used to generate the grids is found and presented in Figure 5.5.

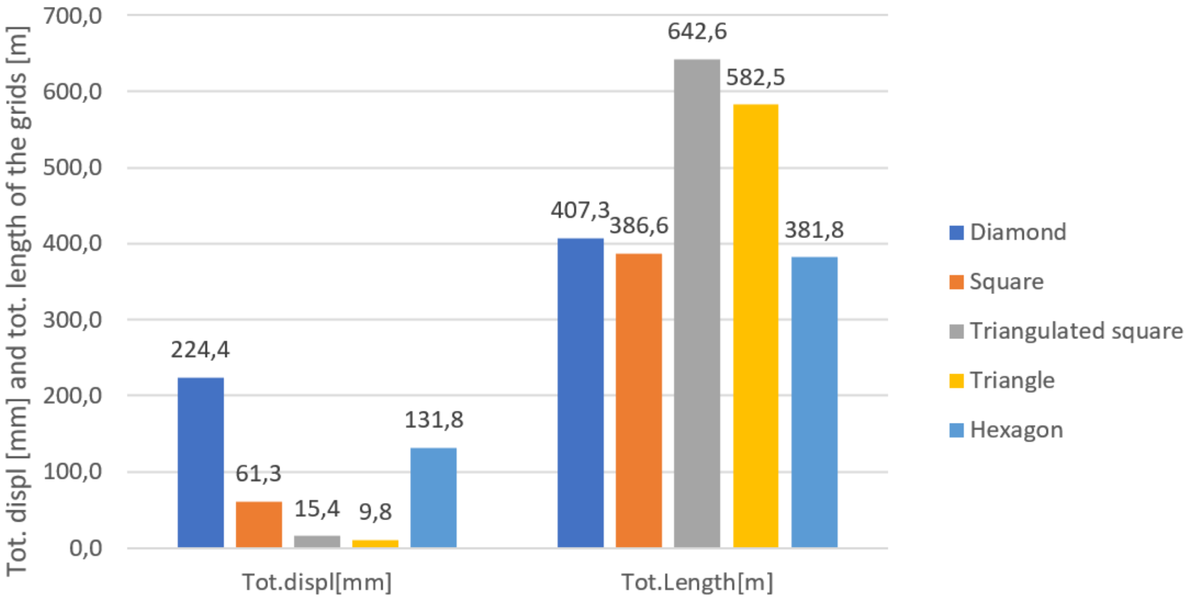
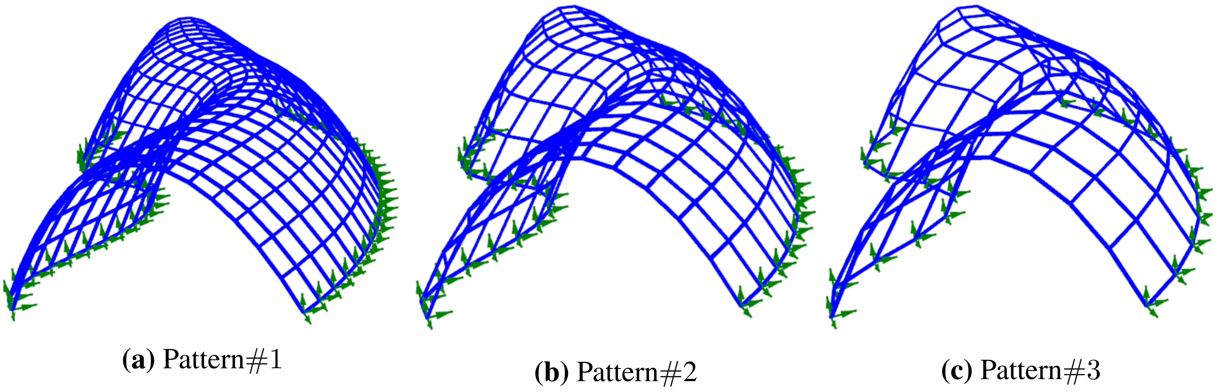


Figure 5.5: Maximum displacement and total length of material used to create the grid patterns presented in Fig. 5.1

As seen from Figure 5.5, the displacement is found to be largest for the diamond and hexagon grids. In other words they seem to display less stiffer structures compared to the three others. The triangular grid seems to have the lowest displacement with a maximum value of $9.8mm$. The total length of material used depends on the pattern, and it shows that the triangular patterns (Triangulated square and Triangle) investigated consist of more material than the others.

Different spacing lengths of the square grid pattern for the freeform grid shell



(a) Pattern#1

(b) Pattern#2

(c) Pattern#3

Figure 5.6: Square grid pattern presented in Fig. 5.1 with different spacing lengths

To compare the effect the beam lengths may have on the shell behaviour, the square grid pattern is analysed for different spacing lengths (see Fig. 5.6). Figure 5.6b portrays the shape already analysed in the previous sections with beam lengths of approximately $1m$, and Figures 5.6a and 5.6c represent the same pattern but with average beam lengths of approximately $0.7m$ and $1.4m$, respectively.

The results obtained are presented in Table 5.1 and illustrated in Figure 5.7.

Table 5.1: Normal forces $N_{compression}$ $[kN]$ and tot.displacement $[mm]$ for varying spacing lengths

	Beam length $[m]$	$N_{compression}$ $[kN]$	Tot.displ. $[mm]$
Pattern #1	0.7	-21.9	43.6
Pattern #2	1.0	-30.2	61.3
Pattern #3	1.4	-44.2	115.8

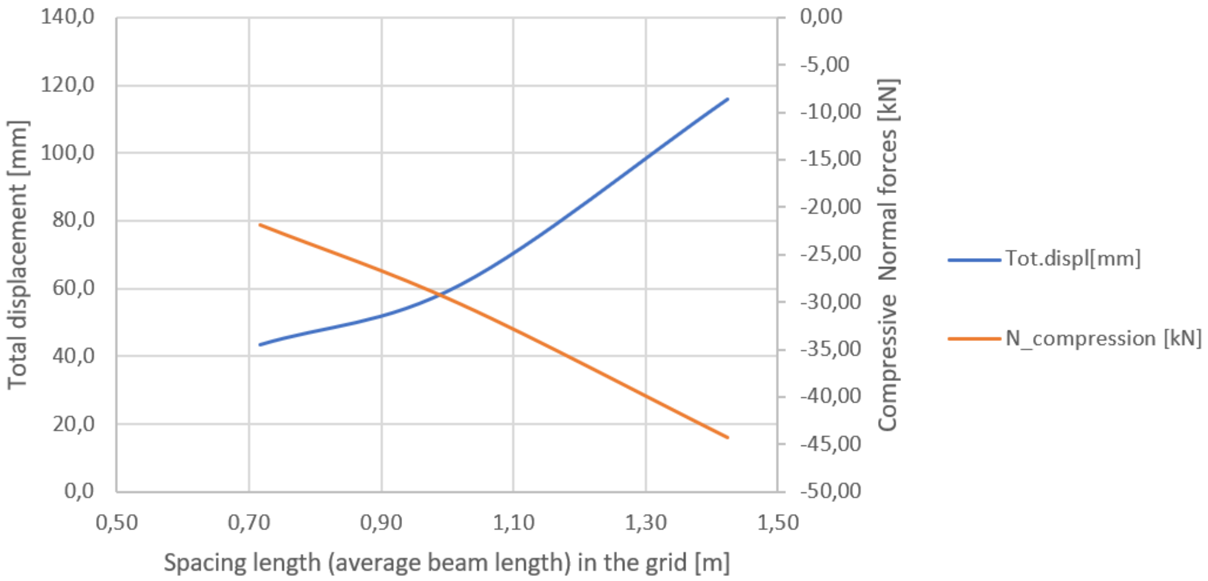


Figure 5.7: Normal forces $N_{compression}$ $[kN]$ and tot.displ. $[mm]$ for varying spacing lengths (x-axis)

As seen from the graphs in Figure 5.7, both the displacement and normal force seem to show approximate linear behaviours when the spacing length is increased. The compression forces increase for larger beam lengths, as do the displacements. In other words, a denser grid will give lower displacements and normal forces.

5.3 Principle stress lines for an equivalent continuous shell

The principle stress lines indicate the trajectories of internal forces. In other words it provides information for the ideal paths of where the material should be arranged, namely the pattern for the optimal grid topology. By creating a stress-based grid pattern, it enhances the possibility of achieving complex structures both satisfying architectural design and structural performance and efficiency.

Figure 5.8 illustrates the principle stress lines for the freeform structure. From the figure one can see how the blue lines (second principle stress lines) is shaped like arches along the surface, starting from one side and ending on the other supported side. The path of the red lines will vary more. However, the general direction of them will be parallel to the supports. The stress lines resembles the square grid (Fig. 5.1b) however, the ideal pattern would in fact be a grid oriented approximately $45°$ on the principle lines. This to ensure a more smooth distribution of stresses, and to avoid highly utilised beam elements.

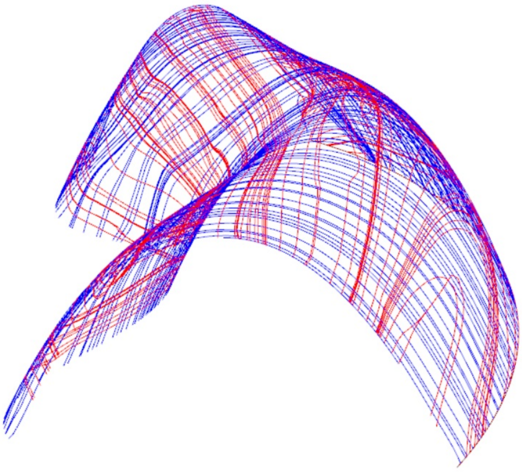


Figure 5.8: Principle stress lines for the freeform structure modelled as a continuous shell. Red lines illustrates the first principle stresses, blue lines the second principle stresses.

5.4 Discussion

With respect to the figures presented above (Fig. 5.3-5.5), the triangle and triangulated square grid seem to show the lowest overall values. However, the total length of material used in these patterns is higher than for the other patterns, which indicate higher costs. The square grid is also showing relatively low values of normal forces, moments and displacement, and consist of only 60 $\%$ of the material used in the triangulated square.

The total amount of beam elements is a relevant measure not only for costs, but with regards to comparing the respective normal forces and displacement. As the snow load applied will be the same for all grids, the expected normal forces in each beam will be smaller if the forces can be distributed to more beams. This gives an advantage to the triangulated square and triangles, which consist of more beam elements compared to the three other grids.

With respect to displacements, more material could be beneficial as it results in a stiffer structures more suited against deformations. For instance the displacement of the square grid is found to be $61.3mm$, while for an equivalent structure with bracing (triangulated square grid) the displacement is only $15.4mm$.

Hence, the discussion is how to reduce costs (low usage of material) and be able to decrease the displacement as much as possible.

To properly consider the squared, diamond and hexagon grid patterns, the rigidity in the beam joints needs to be considered, as they have a crucial influence on the behaviour and strength of the grids. As they do not have bracing, the joints need to be as rigid as possible for the grids to reach their structural integrity.

6 | Case Study: NTNUI cabin

Through the preceding chapters, we have managed to collect and sort information about shell structures and shapes, and have portrayed some of their advantages and drawbacks. Moving into this chapter, we would like to merge our learnings in the seek of designing a grid shell roof for a cabin, both structurally and architecturally pleasing. The parametric code established for the grid shell in Chapter 4 will be employed also in this chapter. The modifications and/or additions to the code will be emphasised (see Appendix C *Grasshopper code for Case Study* for details).

The cabin is planned to be built by the end of 2019, and should be a part of several small cabins (koier) managed by NTNUI. The project is done in close cooperation with the PhD candidates Marcin Luczkcowski (Department of Structural Engineering) and Steinar Hillersøy Dyvik, both working in the conceptual structural design group (CSDG) at NTNU.

In addition we have worked together with two other NTNU student, Helle Stam Faugstad and Øyvind Sunnvoll Rognes. We have been focusing on the grid shell structure as a whole, and they have been focusing on the joints connecting the beam elements in the grid shell. The intent is to merge the results from our theses and together develop a code (design procedure) taking into account all the necessary concerns with regards to the structural system and capacity of the grid shell cabin.

Note that the shape in question is under development as the project of the NTNUI cabin is still at an early stage. Hence it is very likely that other concepts will be considered, and that the final shape will not be similar to the shape investigated in this thesis. This is however where the beauty of parametric design will come in handy, as it is flexible and not fixed to one shape. The parametric model can hence be adopted also for future conceptual shapes, as the process presented in this chapter is a recipe for structural optimisation and design of a grid shell.

6.1 Information

As mentioned previously in this thesis, the location for the cabin is across the fjords of Trondheim in Indre Fosen Kommune (see Ch. 3). Necessary information of the location is presented in Table 6.1, and the site of the cabin can be seen in Figures 6.1 and 6.2.

Table 6.1: Information about the building site of the NTNUI cabin

Name of area	Mevassetran
Coordinates	63.5898110, 10.3258360
Approx. floor area of the cabin	115 m^2

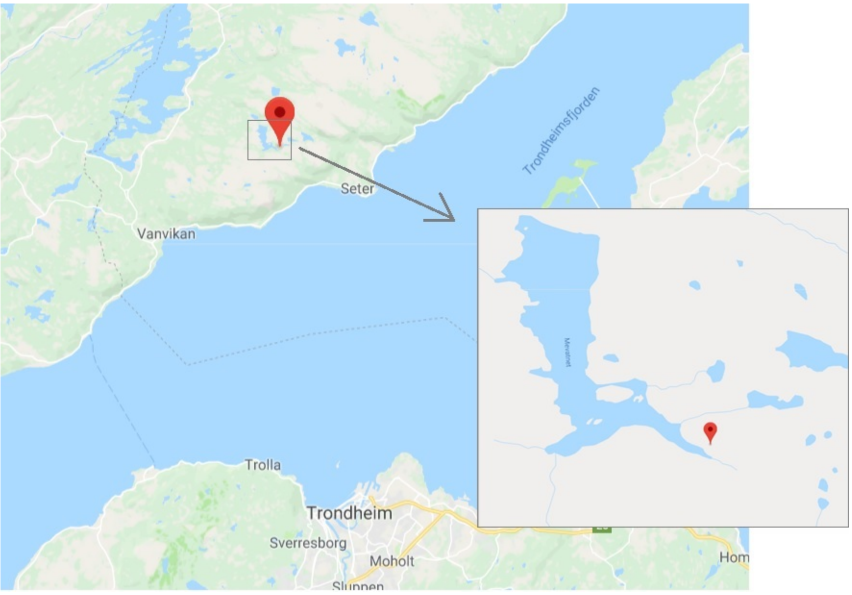


Figure 6.1: Map showing where the location of the cabin is with respect to Trondheim. The cabin is located at Mevassetran, next to a lake. [43]

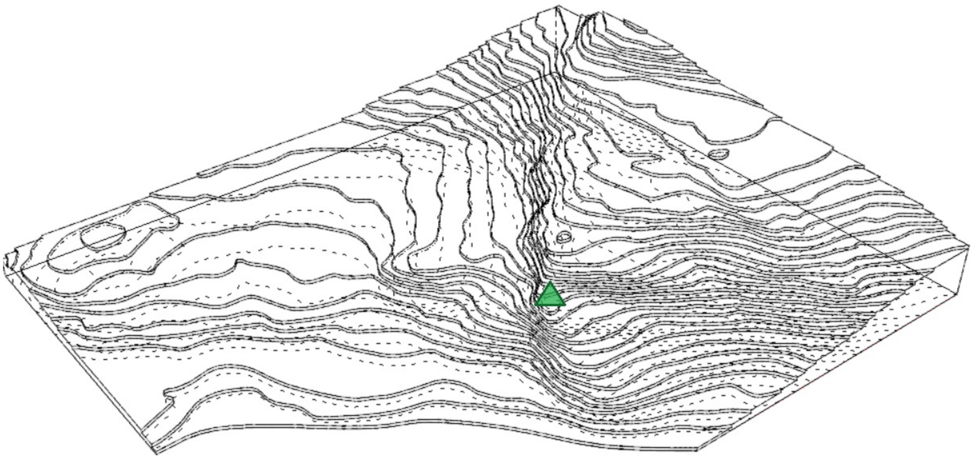


Figure 6.2: 3D topographic map of the cabin location at Mevassetran

6.2 Concept of shape and function

The reasons for constructing a grid shell (with the use of parametric modelling) in the Norwegian mountains can be seen as the following:

- Light weight structure (emphasised in Sec. 4.4.4) - minimal use of materials, reduced costs, reduce amount of heavy lifts during the construction phase.
- Easy assembly with prefabricated precision, where the parametric software handles the modelling complexity and the manufacturing is done with Computer Numeric Controller (CNC) machines. The simple assembly will be positive for the constructors of the cabin, which will most likely be volunteers and not professionals.
- Unique structure, architecturally and structurally pleasing
- Reflects and represents the university NTNU, and the students' competence and creativity

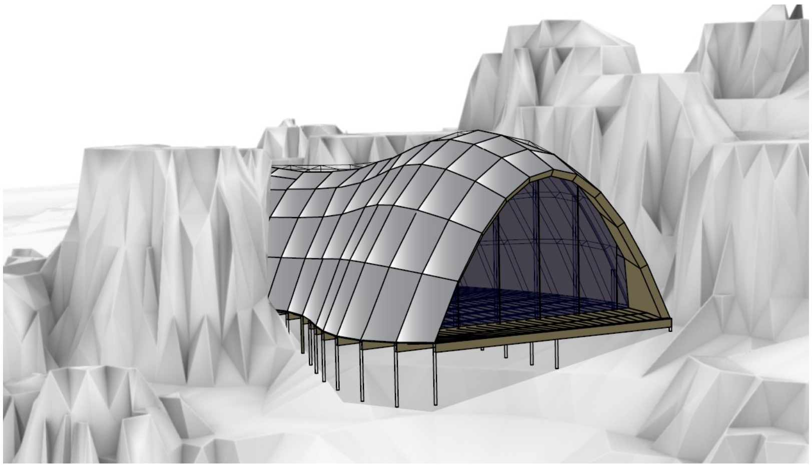


Figure 6.3: 3D illustration of the first concept of the cabin presented by the architect, Steinar Hillersøy Dyvik

The concept for the cabin is that it should be simple in terms of function, mainly offering a bright and open space and a shelter for the students/guests visiting. The visual appearance however, should be that it portrays a fully functioning cabin. The focus is hence on the structural integrity of the grid shell as well as the architectural appearance of the shell and its adaptability in the Norwegian mountains. Figures 6.3 and 6.4 illustrates the first concept of the exterior design and shape of the cabin.

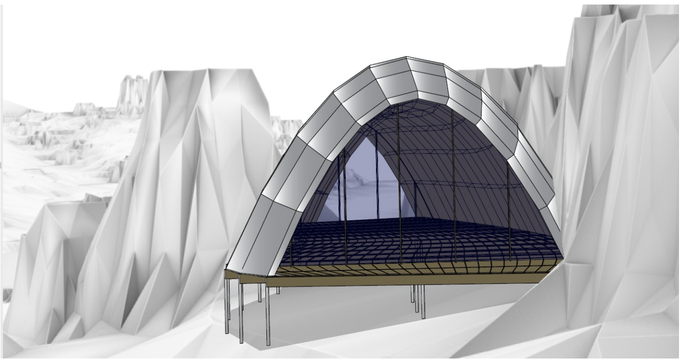


Figure 6.4: 3D illustration (front view) of the first concept of the cabin presented by the architect, Steinar Hillersøy Dyvik

The vision of the shape in question, is to create a space which expresses an open-closed-open arrangement. Visually the shape resembles the Landesgartenschau Exhibition Hall presented in 1.3 *Inspiring shell structures*. Figure 6.5 shows the floor plan of the cabin, with the simplicity of function and interior design. The cabin consists of one room functioning both as a living room, kitchen and bedroom.

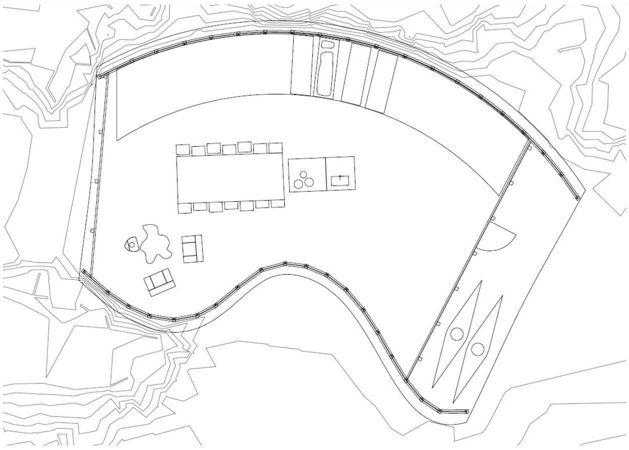


Figure 6.5: Draft of the floor plan for the first concept of the cabin presented by the architect, Steinar Hillersøy Dyvik

The conceptual shape presented by Steinar Hillersøy Dyvik (the architect) will be investigated in this chapter, and alternative changes to the shape will be examined with regards to optimising the structural performance.

To keep a structured arrangement of the shapes in discussion, the initial shape will from here on be spoken of as *Shape 0*.

6.3 Materials and constraints

The shell structure will be built as a timber grid, with straight beam elements connected by aluminum joints. Ergo the grid shell will be non-kinematic, and have the same configuration as the Pods Sports Academy, presented in Section 1.3 *Inspiring shell structures*.

Aluminum joints

The joints are designed in aluminum and are assumed to be rigid. This means all the translational and rotational degrees of freedom on the beam elements are constrained, and moments can be transferred between the beam elements.

Timber beam elements

The material chosen for the beam elements is GL32c. Glulam (combined laminated timber) consist of wood laminates glued together. The laminates on the top and bottom have a higher strength class than the laminates in the rest of the cross section. This is because the greatest tensile and compressive stresses generally occur at the top and the bottom grains of the cross section [44].

The cross sections need to have sufficient bending and compressive resistance. The supplier of Glulam beams is assumed to be Moelven and the type of tree is assumed spruce. Hence, standard dimensions supplied by Moelven are used as a reference for selecting cross sections [45]. The following dimensions in $[mm]$ are considered as standard:

Width 90, 115, 140

Height 90, 115, 135, 180, 225, 270, 315, 360, 405, 450, 495, 540, 585, 630

Shape optimisation

The roof shape received from architect, *Shape 0* will be adjusted with the attempt of improving its load resistance. This will however only include small tweaks, i.e.ensuring the original shape intended by the architect is retained. The available floor area at the construction site will be limited, thus also restricting the possibility of adjusting the shape. There needs to be enough space available for the structure to stand, as well as sufficient space for the assembling of the structure during the construction phase. It is assumed that the architect will consider this and that the small adjustments to the positioning of the supports (affecting the size of the floor area) during optimisation process will be negligible.

6.4 Load combinations

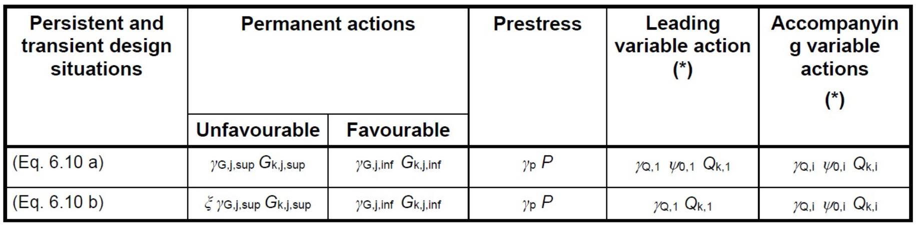
Action on structures is classified by their variation in time [39, 4.1.1(1)]. The classes are as follows:

- *Permanent actions (G)*, e.g. self-weight of the structure.
- *Variable actions (Q)*, e.g. wind and snow action.
- *Accidental actions (A)*, e.g. explosions.

Accidental actions will not be considered for the cabin roof. The individual actions for the critical load cases should be combined with regards to the selected design situation and the relevant limit states. If actions cannot occur similatniously, they should not be considered together in combination [39, 6.1(2)].

Ultimate Limit States (ULS)

The Ultimate limit state concern the safety of people and structures ([39, 3.3(1)]). The ultimate limit state verified in this thesis will be STR, i.e. internal failure or excessive deformation of structure or structural members [6.4.1(1)]. The design should be verified using the design values of actions in Table NA.A2.4(B) (see Fig. 6.6 below).



Persistent and transient design situations	Permanent actions		Prestress	Leading variable action (*)	Accompanying variable actions (*)
	Unfavourable	Favourable			
(Eq. 6.10 a)	$\gamma_{G,j,sup}\ G_{k,j,sup}$	$\gamma_{G,j,inf}\ G_{k,j,inf}$	$\gamma_p\ P$	$\gamma_{Q,1}\ \psi_{0,1}\ Q_{k,1}$	$\gamma_{Q,i}\ \psi_{0,i}\ Q_{k,i}$
(Eq. 6.10 b)	$\xi\ \gamma_{G,j,sup}\ G_{k,j,sup}$	$\gamma_{G,j,inf}\ G_{k,j,inf}$	$\gamma_p\ P$	$\gamma_{Q,1}\ Q_{k,1}$	$\gamma_{Q,i}\ \psi_{0,i}\ Q_{k,i}$

Figure 6.6: Part of Table NA.A2.4(B) in [39]: Design values of actions (STR/GEO)

The following values are used for γ and ξ and ψ:

$\gamma_{G,sup}$ 1.35 (is used when the resulting total load effect is unfavourable)

$\gamma_{G,inf}$ 1.0 (is used when the resulting total load effect is favourable)

ξ 0.89

γ_Q 1.5 for variable actions, when unfavourable (1.6 for wind)

γ_Q 0 for variable actions, when favourable

ψ_0 0.7 (snow load) and 0.6 (wind load) [39, Table NA.A1.1]

When permanent action is unfavourable and snow load is the leading variable action and wind is accompanying variable actions, the design values for effects of actions E_d become:

Eq. 6.10 a):

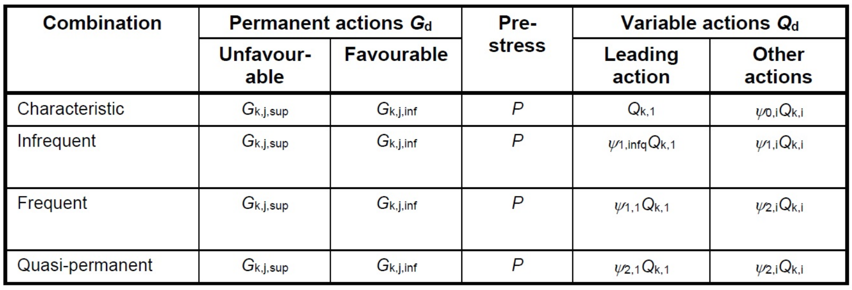
$$E_d = 1.35 * G_{k,j,sup} + 0.7 * 1.5 * Q_{k,1} + 0.6 * 1.6 * Q_{k,2} \tag{6.1}$$

6.10 b):

$$E_d = 0.89 * 1.35 * G_{k,j,sup} + 1.5 * Q_{k,1} + 0.6 * 1.6 * Q_{k,2} \tag{6.2}$$

Serviceability Limit States (SLS)

The Servicability limit state concern the functioning of the structure or structural members under normal use, the comfort of people and the appearnce of construction works ([39, 3.4(1)]). Design values of actions in the SLS are defined in Table NA.A2.6 in [39].



Combination	Permanent actions G_d		Pre-stress	Variable actions Q_d	
	Unfavour-able	Favourable		Leading action	Other actions
Characteristic	$G_{k,j,sup}$	$G_{k,j,inf}$	P	$Q_{k,1}$	$\psi_{0,i}Q_{k,i}$
Infrequent	$G_{k,j,sup}$	$G_{k,j,inf}$	P	$\psi_{1,infq}Q_{k,1}$	$\psi_{1,i}Q_{k,i}$
Frequent	$G_{k,j,sup}$	$G_{k,j,inf}$	P	$\psi_{1,1}Q_{k,1}$	$\psi_{2,i}Q_{k,i}$
Quasi-permanent	$G_{k,j,sup}$	$G_{k,j,inf}$	P	$\psi_{2,1}Q_{k,1}$	$\psi_{2,i}Q_{k,i}$

Figure 6.7: Table NA.A2.6 in [39]: Design vaules of actions for use in the combinations of actions

Section NA.A1 4.2 (3) states that deflection can normally be calculated for the action situation *quasi-permanent*, when the time-dependent effects are important. Whereas the additional action for the design situation *characteristic* or *frequent* is regarded as short-term load. We assume that the snow load will be the leading variable action and the main contributor to the deflection. Hence, we conservatively consider the *characteristic* design situation when looking at the deformation of the roof, as this is the most critical situation. The following will be the design value:

$$E_d = G_{k,j,sup} + Q_{k,1} + Q_{k,i} \tag{6.3}$$

6.5 Actions on the roof

The loads which needs to be considered when designing the roof will be presented in the following. An investigation will be done in the seek for the worst load distribution of the variable loads, snow and wind. This will be divided into two approaches. The first one will be to draw the concepts from Chapter 3 *Actions according to Eurocode* and determine a suitable load distribution on the shell roof both for a drifted and undrifted load case. Secondly, the Galapagos evolutionary solver in GH will be used to find the most critical placement of load with regards to what load configuration gives the largest displacement and moments.

6.5.1 Self-weight

The self-weight of the building will involve the following:

- The weight of the Glulam beams. Karamba will automatically take into account the weight of the timber beams
- The weight of the aluminum joints connecting the beam elements. This is taken in as point loads in the end nodes of the beam elements. The load will depend on the size of the beam cross section and the necessary thickness of aluminum plates and amount of bolts to transfer the forces. Thus, as a first assumption the load is conservatively considered to be $0.20kN$. After the cross section is optimised the weight of the joints will again be examined to check whether the assumption still stands.
- The weight of the insulation, roof plates and roof covering (asphalt). This is assumed to be a uniform load of $0.5kN/m^2$. The load is applied as a Mesh load with global orientation. The vertices of the mesh faces corresponds to the connecting nodes in the grid, i.e. all the load is transferred to the end nodes of the beam elements.

6.5.2 Live Load

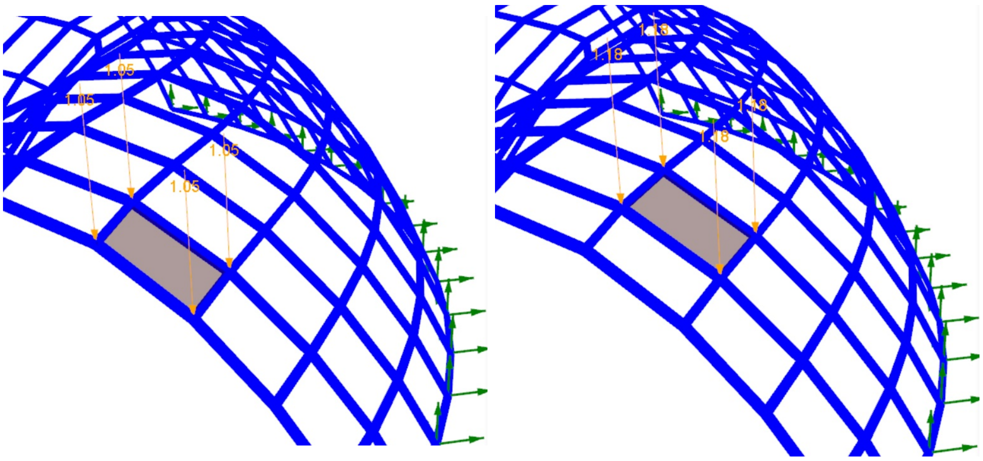
When designing a roof structure, it is necessary to ensure that the design resistance is considering all plausible loads. This would include maintenance work on the roof, meaning the load of at least one person climbing and/or standing on top of the roof should be considered. For the early stage of the design process of the NTNUI cabin presented in this Chapter, live load will though not be considered.

6.5.3 Snow Load

Snow is an environmental load caused by nature, thus a lot of variables will influence how and to what degree the snow affects the structure. The difficulty in predicting how the snow is distributed leads to the requirement of considering several load situations. To determine the magnitude of the snow load on the roof the characteristic snow load and shape factors given in Eurocode ([35]) are used (same procedure as found in Ch. 3, Sec. 3.1). The snow load distributions are also found in accordance with Eurocode (further explained in the following subsection).

To investigate which load distributions actually gives the most critical displacements and moments, Galapagos will be adopted to parametrically generate the distribution. In addition a load case where the snow load is applied evenly over the entire roof will be considered.

In Chapter 4 the snow load was applied as a point load with equal magnitude to all relevant nodes (see Sec. 4.2.4). In this Chapter the snow load is globally projected and applied using *Mesh load* in the $Loads(Karamba)$-component. The surface is meshed with the same uv-count as the grid pattern. As a result the mesh faces is corresponding to the rectangular grid faces (see Figure 6.8). This way the load is transferred to the beam element end-nodes via the vertices of the fictitious mesh faces. The resulting load transferred to the nodes will vary depending on the location of the load, hence this load distribution will be a better approximation in comparison to the one adopted in Chapter 4.



(a) Mesh face 1

(b) Mesh face 2

Figure 6.8: Loads applied on the corners of the fictitious mesh faces corresponding to the nodes of the grid. Two arbitrary mesh faces are illustrated with the distinctive loads applied in the corners.

Snow Load distribution according to Eurocode

In Chapter 3 *Actions according to Eurocode* it was found that Eurocode provides limited information on load distribution on irregular roof shapes. The cylindrical shape will be the geometric shape having the most resemblance with the freeform roof received from the architect, *Shape 0*. Thus an assumption is made that the loads on *Shape 0* will coincide with the load distribution on the cylindrical shape presented in Section 3.1 *Snow calculations according to NS-EN 1991-1-3* in Figure 3.4. Some simplifications will however be necessary to adopt EC on the freeform shaped cabin. In Section 3.1 it was explained how Eurocode considers both a drifted and undrifted load configuration, depending on the wind effects.

Undrifted Load Case

The *Undrifted* load case was adopted in Chapter 4, when the basic geometric shapes were studied. However, the load distribution now becomes more irregular as the shape is not symmetric. Figure 6.9 shows how the load will be applied, where the marked areas represent the mesh faces where the snow will be applied, i.e. $\beta \leq 60°$. The load values are presented in Table 6.2.

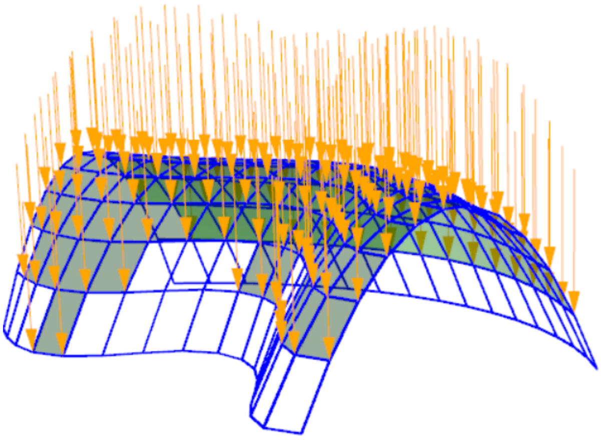


Figure 6.9: Undrifted load snow load according to Eurocode distributed on *Shape 0*. The green area highlights where the load is applied

Table 6.2: Shape coefficient (μ_3) and undrifted snow load for *Shape 0*. Characteristic snow load is $s_k = 7kN/m^2$. The snow load is found with Eq. (3.1)

	Shape factor	Undrifted snow load $[kN/m^2]$
Shape 0	μ_3 0.8	s(μ_3) 5.6

Drifted Load Case

The *Drifted* load case will give an asymmetric pressure on the reticulated shell structure. Figure 6.10 shows the basis on which the load area is selected. The strip of grid faces in the entrance/opening of the structure (highlighted faces in Fig. 6.10) is similar to the shape of a cylindrical shell. The load is hence applied in accordance with the cylindrical shape presented in Chapter 3 *Actions according to Eurocode* (Sec. 3.1.1 and Fig. 3.4).

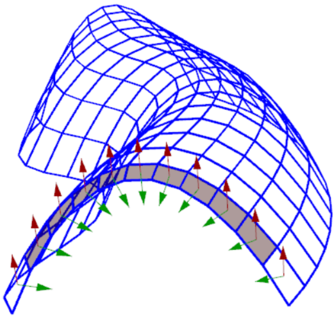


Figure 6.10: Area where the snow load is applied on the first strip of grid faces. The green arrows display the normal vectors of the grid faces, which is used to determine the curvature based on the angle with the unit z-vector (red arrows). The curvature of the marked faces is $\beta \leq 60°$.

In GH, this curve is analysed to find where the angle is less than $60°$, by finding the normal to all faces, and comparing this vector with a unit vector in z-direction (see Fig. 6.10). Where the absolute value of the angle is smaller or equal to $60°$, the faces are selected. The curvature and the faces with $\beta \leq 60°$ will change for every strip of grid faces over the length of the structure. However to simplify this, the distribution on the first strip is repeated for all strips over the length of the surface.

The concept of the *Drifted* load case from Figure 3.4 is adopted. The concept being that the snow will be drifted by the wind, and accumulate more on one side than the other. However the triangular snow distribution with shape factor μ_3 in Figure 3.4 is replaced by a rectangular distribution increasing in magnitude towards the right (see Fig. 6.11). The corresponding shape factor is named μ_4.

The magnitude of the shape factor μ_4 is found by requiring the same resulting amount of snow on the freeform roof as for the cylindrical. Table 6.3 presents the resulting snow loads in the three zones considered (see Fig. 6.11) for the cabin, and two zones for the cylindrical roof, respectively.

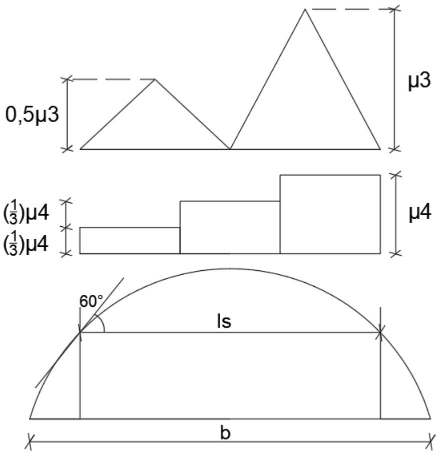


Figure 6.11: The triangular snow distribution with shape factor μ_3 given in Eurocode and the approximated snow distribution used for *Shape 0* with shape factor μ_4. l_s illustrates the load width where the curvature is less than $60°$.

Table 6.3: Shape coefficients (μ_3 and μ_4) and drifted snow loads for cylindrical and *Shape 0*. Zone 1, 2 and 3 corresponds to the three levels of magnitude in Fig. 6.11. The characteristic snow load is $s_k = 7kN/m^2$, found with Eq. (3.1)

	Shape factor	Drifted snow load $[kN/m^2]$		
		Zone 1	Zone 2	Zone 3
Cylindrical shape	μ_3	s(0.5μ_3)	s(μ_3)	
	2.0	7	14	
Shape 0	μ_4	s($\frac{1}{3}\mu_4$)	s($\frac{2}{3}\mu_4$)	s(μ_4)
	1.13	2.63	5.25	7.88

Thus by separating the load area into three zones (illustrated on the left shape in Fig. 6.12), the loads in the given zones are applied with magnitudes in accordance with Table 6.3 (*Shape 0* in 2_{nd} row).

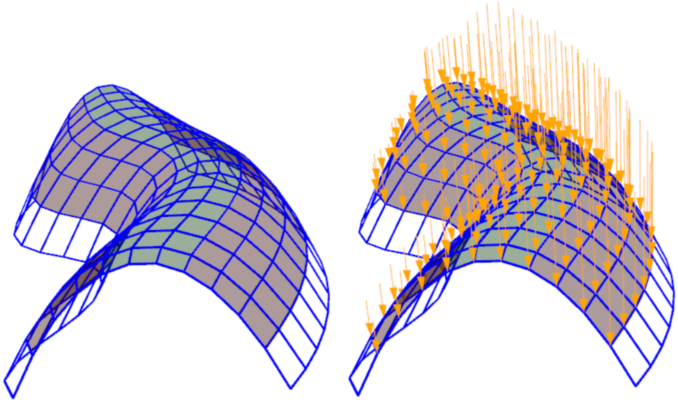


Figure 6.12: Illustration of the three zones assumed to have a different magnitude of snow load (left ill.) and the drifted load distribution in Karamba (right ill.). Increasing magnitude of the snow load from left to right. The load distribution is based on the concept of *Drifted* snow load from Eurocode

Snow Load distribution found with Galapagos

The following assumptions were made when finding the worst snow load situation for the roof with Galapagos (see Sec. 2.6 for details about software and *Galapagos*):

- Uniform snow load with the same load value as presented in Table 6.2 (*Undrifted*).
- No restrictions to where the load can be applied, i.e. the $\beta \leq 60°$ from Section 3.1 is no longer accounted for.

Galapagos was used to find the 4 load distributions maximising the following (see Fig. 6.13 for directions):

1. Displacement in x-direction
2. Displacement in y-direction
3. Displacement in z-direction
4. Moment $M_{res} = \sqrt{M_y^2 + M_z^2}$

The meshed surface was first exploded into its faces. Where each mesh face corresponds to the rectangular grid face and represents a possible load area. The load applied on the mesh face was transferred to the end-nodes of the beam elements. A gene pool with as many *genes* (i.e. variables) as there were mesh faces, was then created. Where the *genes* in the gene pool is a collection of sliders where each can alternate between 0 and 1, i.e. *false* and *true*. The sliders in the gene pool controls which one of the mesh faces are "turned on"(1) and which ones are "turned off"(0). If the load on the mesh face leads to a larger displacement in the given direction, the mesh face in question is "turned on". These mesh faces (*true*) were then taken as inputs into the $Loads(Karamba)$-component and applied the snow load.

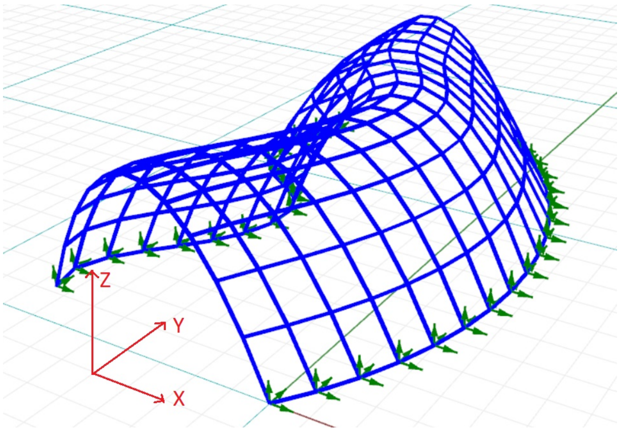
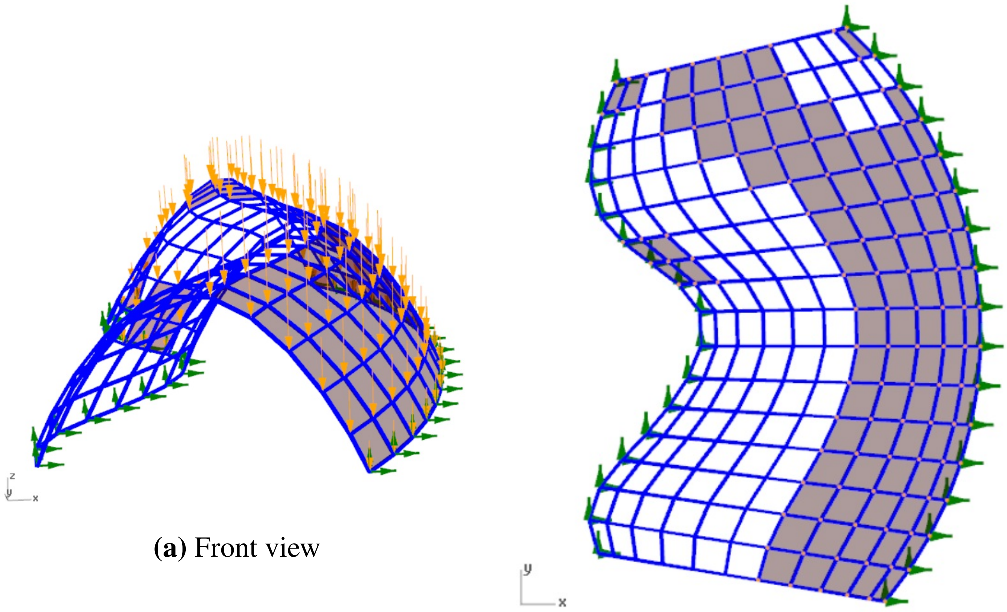


Figure 6.13: *Shape 0* with global coordinate system

The maximisation process with Galapagos gave the following load distributions for the four cases:

1. Maximise displacement in x-direction

Load distribution for load case: *Max x-dir.*



(a) Front view

(b) Aerial view

Figure 6.14: Snow load area found by maximising the displacement in x-direction with Galapagos

The displacement in x-direction was found to be maximised when the snow load was applied to the right side of the structure. Figure 6.15 illustrates how the structure deforms by being pushed to the side by the snow load.

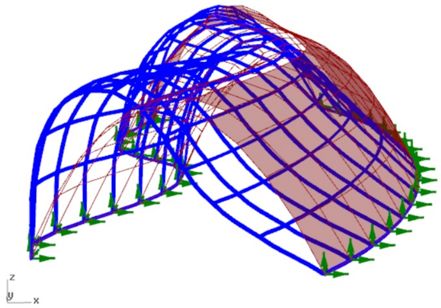
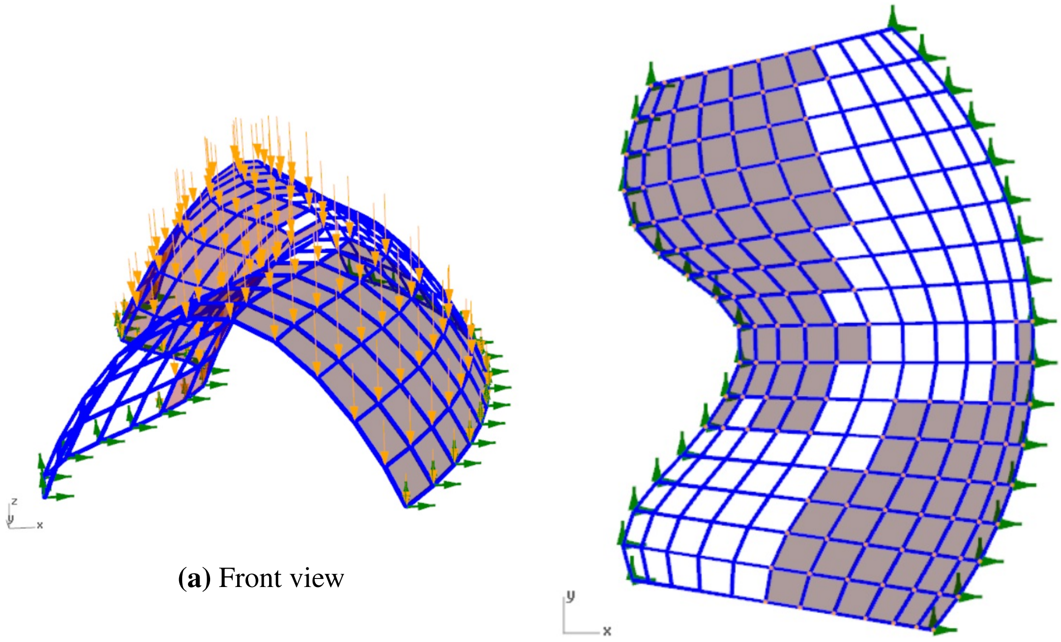


Figure 6.15: Deformed structure with snow load case *Max x-dir.*

2. Maximise displacement in y-direction
Load distribution for load case: *Max y-dir.*



(a) Front view

(b) Aerial view

Figure 6.16: Snow load area found by maximising the displacement in y-direction with Galapagos

As seen in Figure 6.16, the critical load case will be when the snow is distributed in the two opposite quadrants, similar to the chessboard pattern previously mentioned for the spherical shaped roof in Chapter 3 (Sec. 3.1.2). From Figure 6.17 one can observe how the the load will twist the structure which leads to a large horizontal displacement along the length of the structure (y-direction).

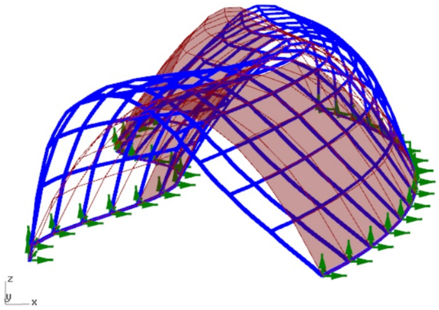
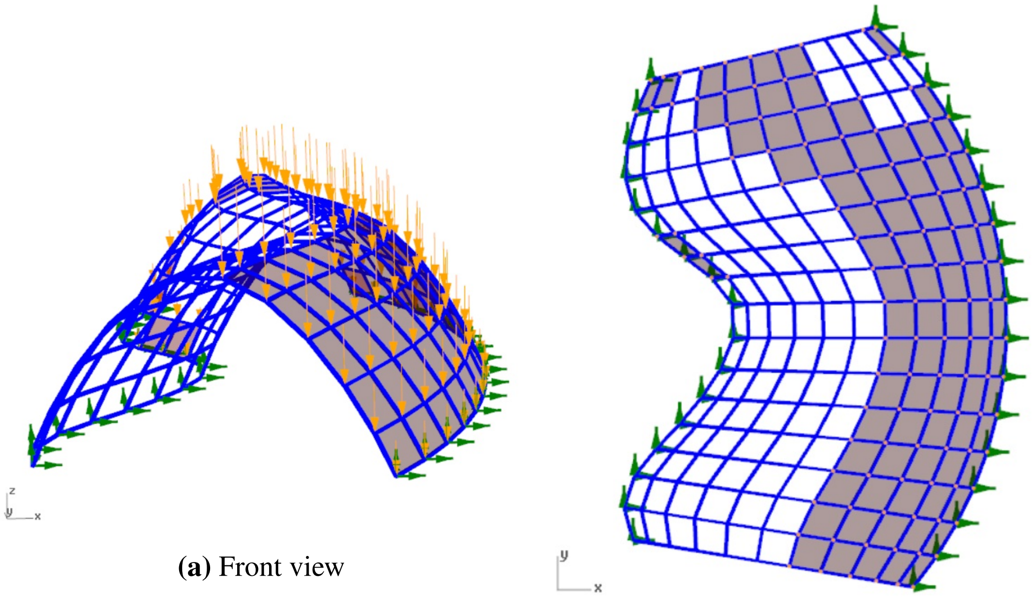


Figure 6.17: Deformed structure with snow load case *Max y-dir.*

3. Maximise displacement in z-direction

Load distribution for load case: *Max z-dir.*



(a) Front view

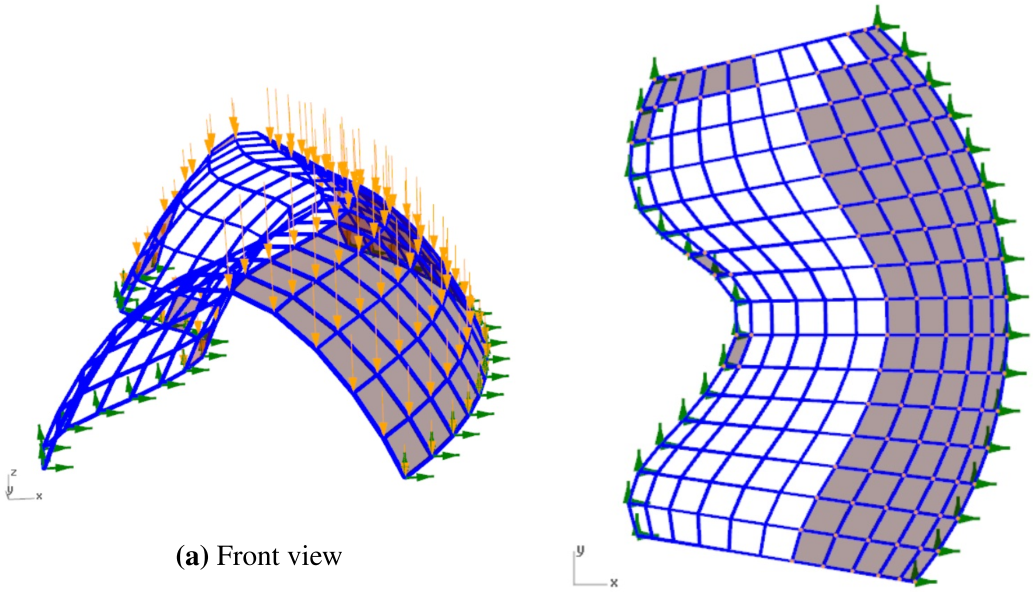
(b) Aerial view

Figure 6.18: Snow load area found by maximising the displacement in z-direction with Galapagos

A natural assumption could be that when the snow load is distributed over the entire structure and the structure is applied maximum snow load, the displacements in z-direction would be largest. However, from Figure 6.18 it can be observed how the load distribution causing the highest displacement in z-direction is when the load is applied asymmetrically on the surface. The load distribution is very similar to the load case for *Max x-dir.*, which results in almost identically deformed structures for both load cases. The deformation of *Max z-dir.* is hence similar to Figure 6.15.

4. Maximise resulting moment

Load distribution for load case: *Max moment*



(a) Front view

(b) Aerial view

Figure 6.19: Snow load area found by maximising the occurring moment M_{res} $[kNm]$ in the beam elements with Galapagos

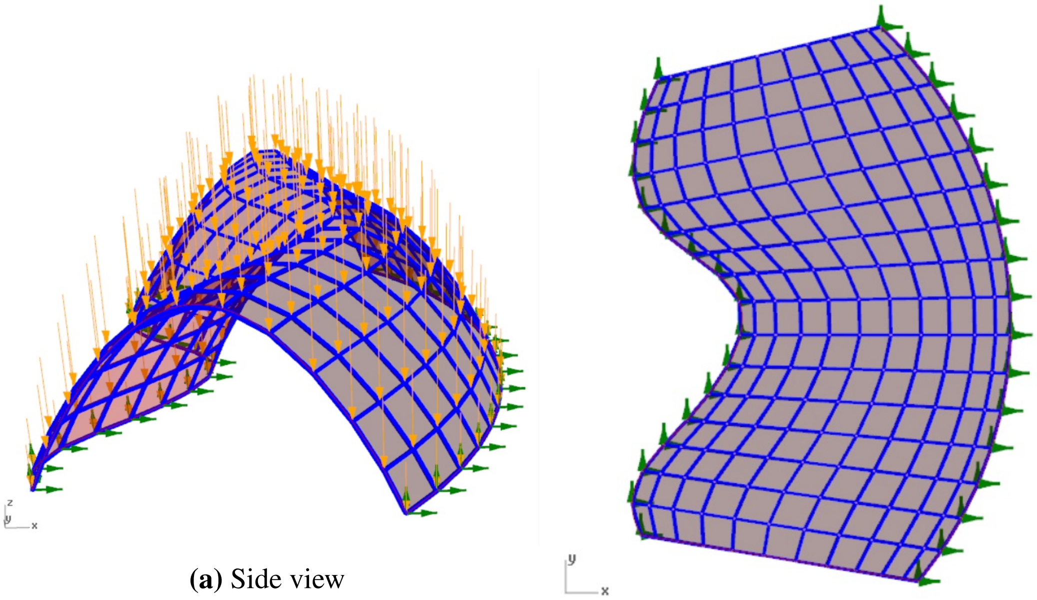
The worst load situation giving the maximum resulting moment, $M_{res} = \sqrt{{M_y}^2 + {M_z}^2}$, is when the snow load is applied along the right side of the structure as shown in Figure 6.19. This is similar to the load cases *Max x-dir.* and *Max z-dir.* presented in the load cases No.1 and No.3 above.

Brief recap regarding the loads found with Galapagos

Unlike the load distributions according to Eurocode, the load distribution found with Galapagos will not be a result of empirical data based on real situations. Thus it is necessary to be extra careful in adopting the load distributions, and discuss the plausibility of the loads occurring. Fortunately for all the load situations, the snow loads were found to accumulate in the same area and were not spread out randomly over the surface. The snow will not naturally gather in only certain areas of the roof unless wind, shape of the roof or other factors affects the snow. A large wind load coming in perpendicular to the side of the structure, could lead to the snow accumulating on one side, like the load cases *Max x-dir*, *Max z-dir* and *Max moment* illustrates. The load case *Max y-dir.* where the snow accumulates in two opposite quadrants, seems less likely.

Uniform snow load over the entire roof

Loadcase *Uniform*:



(a) Side view

(b) Aerial view

Figure 6.20: Snow load distributed over the entire surface area (Load case *Uniform*)

Directional displacements and moments for all the snow load cases

The displacements and the moments from the different snow load cases are presented in Table 6.4 below.

Table 6.4: Resulting directional displacements $[mm]$ and moments M_y and M_z in $[kNm]$ for all the snow load cases introduced in this section

Loadcase	Displ.x $[mm]$	Displ.y $[mm]$	Displ.z $[mm]$	M_y $[kNm]$	M_z $[kNm]$
Undrifted	35.8	10.3	19.4	0.85	1.95
Drifted	118.2	27.5	81.6	2.12	4.35
Max x-dir.	214.5	45.1	134.4	2.84	6.09
Max y-dir.	152.5	83	97.4	2.52	6.16
Max z-dir.	214.5	44.3	134.5	2.84	6.1
Max Moment	211.5	38.8	132.7	2.79	6.24
Uniform	18.7	4.0	14.8	0.41	1.01

In Table 6.4 the results from the load distributions found with Galapagos, and the Eurocode loads are presented. The values for load cases *Max x-dir.*, *Max z-dir.* and *Max moment* are shown to be approximately similar. This is due to the load distributions obtained with Galapagos, where the load is applied along the right side of the structure for all three cases (see Fig. 6.14, 6.18 and 6.19).

As seen from Table 6.4 the *Undrifted* and *Drifted* load cases found on the basis of Eurocode are not the load distributions giving the most critical displacements and moments. The load case *Uniform* where the load is applied over the whole surface is displaying even lower values, illustrating how the shell is efficient in withstanding symmetric loads. Considering the distributions obtained with Galapagos, the displacements are found to be quite large compared to the ones EC suggest - even though the *Undrifted* and *Drifted* load cases in question are limited to the extent EC provides information. However, it is evident that it could be risky to solely rely on the load distributions given by EC. The discussion turns into whether the load cases found with Galapagos are probable or not, and ideally, the loads should be verified experimentally for the given shapes.

As the results from Galapagos contradict with some of the guidelines given in Eurocode (e.g. accumulations doe not occur where $\beta \geq 60°$), the load cases found in accordance with EC will be load situations considered further on in this thesis. They will be referred to as the *Undrifted* and *Drifted* load case.

6.5.4 Wind Load

It is assumed that the wind load distribution on *Shape 0* will resemble the wind load distribution on the cylindrical roof as presented in Section 3.2 *Wind calculations according to NS-EN 1991-1-4*.

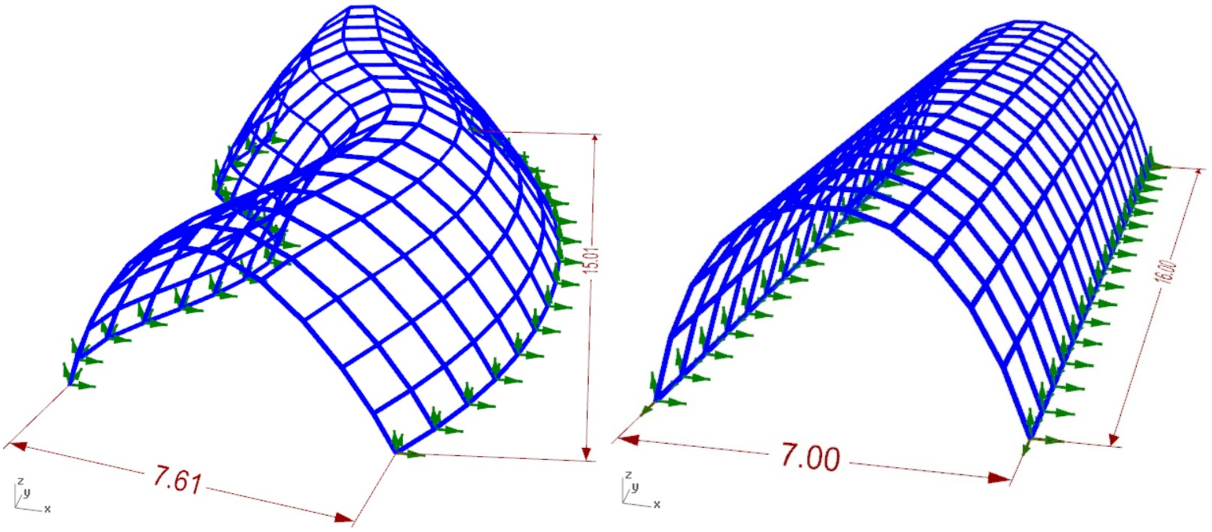
The wind pressure values obtained for *Case 1 - Cylindrical roof shape* in 3.2.1 were all low. Negative values were found for both wind blowing parallel ($\theta = 0°$) and perpendicular ($\theta = 90°$) to the eaves, see Table 3.3 and 3.4, respectively. Negative values imply suction over the entire roof area, which in fact makes the combination of snow and wind favourable (opposite load directions).

A load situation that could occur is wind acting on the structure without snow load present. The wind could in worst-case lead to a lifting of the roof if the suction is high enough. Due to the light-weight characteristic of the structure, this scenario should be carefully considered. In this thesis it is assumed that snow load is always present. As a result of this the presence of wind will always be favourable, and thus wind loads are not accounted for in this thesis.

6.6 Initial study of the shape

Initially it appears that *Shape 0* resembles the barrel vault investigated in Chapter 4 *Parametric Study*. Both have openings on the ends of the structure, and they both have shapes defined by arched sections along the structure. Figure 6.21 illustrates the two structures in question.

As seen from Figure 6.21a, *Shape 0* also shows to have the same effective feature as the dome, namely the double curvature. As presented in Chapter 2 *Governing equations* and illustrated in Chapter 4 *Parametric Study* (summary of the Parametric Study), a shell structure which is doubly curved will be more efficient than a singly curved shell. This increases the rigidity of the shape and will cause it to better withstand imposed loading with regards to deformation.



(a) *Shape 0*

(b) Barrel Vault

Figure 6.21: The freeform *Shape 0* vs. a regular shaped barrel vault of similar size

To check if the double curvature will have a significant effect on the resulting displacements, a comparison is made between *Shape 0* and the generic barrel vault (otherwise rather similar shapes). The aim is to understand the structural intention behind creating a shell with an unsual shape.

The dimensions for the floor area of the barrel vault are selected based on reaching a similar floor area as for *Shape 0*. The height of the two shapes are both $4.14m$. The cross section of the beam elements and grid pattern are equal for both shapes. As a result they are comparable with regards to amount of material used and their boundary conditions are similar in that only two sides are supported. Both shapes are applied the *Undrifted* snow load as shown in Figure 6.9, with values presented in Table 6.2.

Results from the initial study of *Shape 0*

The directional displacements and maximum moment M_{max} for the two shapes was found, and the results are presented in the following table (Tab. 6.5):

Table 6.5: Directional displacements $[mm]$ and moment $M_{max} = max(M_y, M_z)$ obtained for *Shape 0* and the barrel vault

	Floor area $[m^2]$	Displ. x $[mm]$	Displ. y $[mm]$	Displ. z $[mm]$	M_{max} $[kNm]$
Shape 0	114.7	35.7	10.2	19.3	1.94
Barrel vault	112	115.9	0.18	105.3	3.05

The directional displacements of the barrel vault were found to be approximately 4 times larger (or more) than for *Shape 0*. The exception is the horizontal displacement parallel to the structure's length (displ.y) where the barrel vault seems to barely move.

The results obtained in Table 6.5 suggest that enabling the generic barrel shape to be adjusted slightly could reduce the displacements and the moments in the structure. By allowing the supports to be generated on a curved rather than a straight line, the shape can be generated with a double curvature, which repeats to show a more rigid structure.

6.7 Optimisation of the shape

In this section the aim is to optimise the *Shape 0* using the genetic algorithm solver, Galapagos, with respect to minimising the displacements (see Sec. 2.6 *Software tools* for details about *Galapagos*). The intention is to illustrate how small changes to the geometry can improve the structural performance. The loads accounted for in Section 6.7 will be the *Undrifted* and *Drifted* snow loads according to EC, as presented above in Section 6.5. Self-weight, live load and wind load will not be considered in the optimisation of the shape, but the first become included in the design proposal given in Section 6.8 (wind load and live load not considered, see Sec. 6.5.2 and 6.5.4).

Optimisation process

The changes of the geometry include the following:

- Right rail: change the span, the overall shape remains the same
- Section: Height and shape of the section, floor area remains the same
- Left rail: Curvature where the shape is doubly curved, section remains the same

Note that the left rail, right rail and section mentioned above refer to the curves in which define the surface of the structure, see Fig 6.22 (explained further in Sec. 6.7.1).

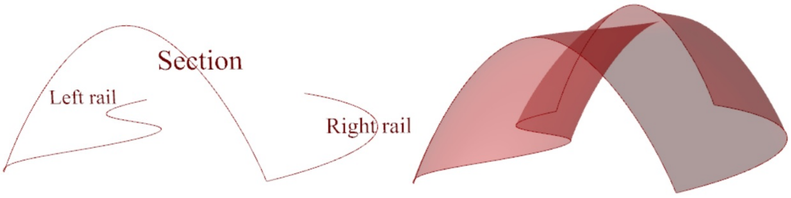


Figure 6.22: Curves defining the surface of *Shape 0*. The curves are named left rail, right rail and section

The following figure 6.23 illustrates the optimisation process of finding a more robust shape with lower displacements (*Shape 1*).

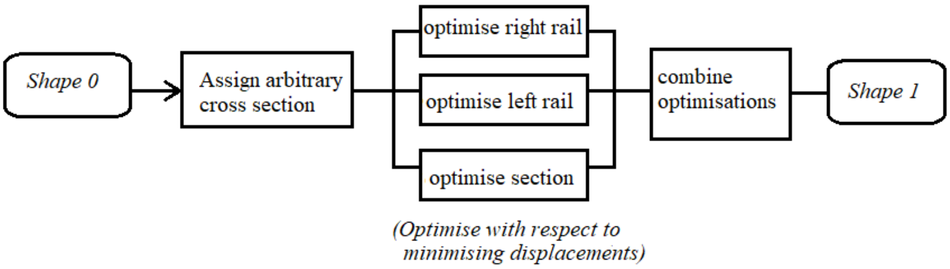


Figure 6.23: Flow diagram for the optimisation process of the shape, from *Shape 0* to *Shape 1*

In addition to the optimisation of the shape and minimising the displacement, an additional analysis is done where the displacements are maximised. This is added to get an idea of what features of the shape should be avoided. The results for both the optimisation (minimisation) and the maximisation will be presented in the same table.

6.7.1 Rebuilding surface in Grasshopper

Unlike the geometric shapes used in Chapter 4 which were defined within the GH environment, the shape is now defined by a surface created in Rhino. The surface was originally made by the component $Sweep2$ using three defining curves. To be able to control the shape of the surface in GH and perform a structural optimisation, the surface needs to be redefined in GH. To achieve this, the curves used to establish the sweep-surface in Rhino are extracted in GH. These curves (NURBS curves) include the two rails in the XY-plane, defining the shape along the length of the structure, and the section which defines the sectional shape (see Fig. 6.22). The control points for the three curves are found, and separated to be able to individually adjust their positions.

The $Move$-component is connected to the control points, and a vector $\{x, y, z\}$ is taken as input to determine the movement of the points ($z = 0$ for the control points on the rails/supports). By connecting sliders to the x, y, and z-components of the vector, the points can change positions. The parametric control points replace the original control points of the three curves and the two rails and the section can be reestablished. The new curves are then fed into the $Sweep2(old)$-component to rebuild the surface in GH (see Fig. 6.24). Now the shape of the surface can be controlled from within GH.

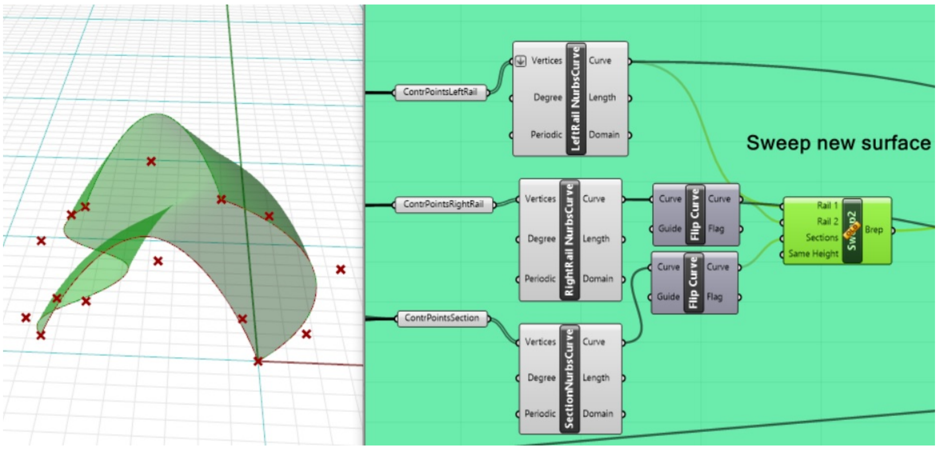


Figure 6.24: Process of rebuilding the surface with the Sweep2(old)-component in GH. The illustration shown is the finishing part where the optimised curves are taken as inputs to rebuild the surface.

6.7.2 Optimisation approach with Galapagos

The shape optimisation is, as mentioned in the beginning of Section 6.7, performed with the genetic algorithm approach using Galapagos. The three curves defining the surface (left rail, right rail and section) will be optimised separately. Each curve is assigned individual gene pools containing sliders that controls the movement-interval of the control points. The gene pool defines the *Genomes* that Galapagos should change with respect to a certain fitness.

The two snow load distributions presented in Section 6.5.3 under *Snow Load distribution according to Eurocode* are used to optimise the shape. A $C\#$-code is generated to find the maximum resulting displacement from the load distributions:

$$d_{resultant} = \sqrt{d_x^2 + d_y^2 + d_z^2}.$$

For each configuration of the control points (genome), the C#-code loops over the load cases, finds the resulting displacement for all nodes and stores them in a list. The largest resulting displacement of the two load cases is determined by finding the maximum of the list. The displacement is the fitness function which Galapagos seeks to minimise, by changing the *Genomes*. This way the structure will be optimised with regards to the load case in which gives the largest displacements. Figure 6.25 illustrates the process explained above, where the right rail is the curve being optimised.

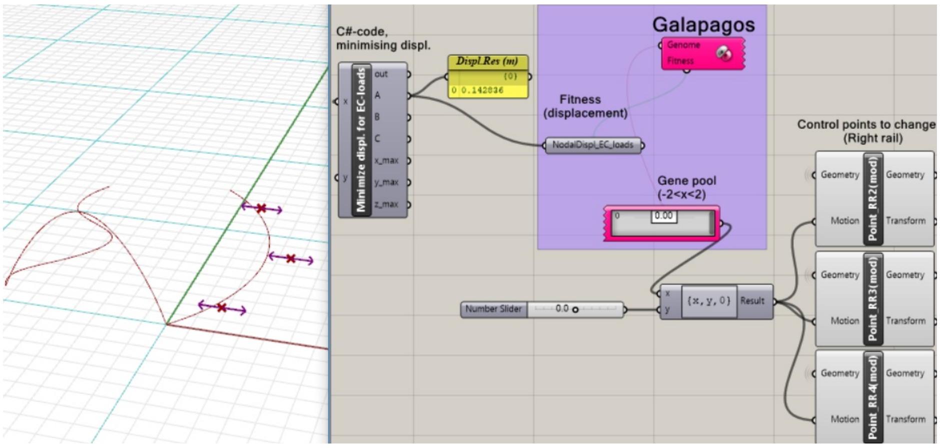


Figure 6.25: Optimisation process with Galapagos. The *Genome* is the parametric control points for the curve in question, and the *Fitness* is minimising the resulting displacement $d_{resultant}$. The example shown is the optimisation of the right rail.

Interpreting results

The optimised shapes will be compared to the *Shape 0* with respect to the results presented in Table 6.6. The optimised shapes will be analysed for the same arbitrary cross section $80mmx80mm$ as selected for *Shape 0*. The *difference* in results for the two shapes will be displayed, where a negative difference means decreasing forces and displacements (green), and positive means increasing values (red) (see Tables 6.7-6.10)

Table 6.6: Structural behaviour of *Shape 0* with the arbitrary cross section $80mmx80mm$

	Displ.Res $[mm]$	*Load case*	Displ.x $[mm]$	Displ.y $[mm]$	Displ.z $[mm]$	M_y $[kNm]$	M_z $[kNm]$
Shape 0	142.8	Undrifted	35.8	10.2	19.4	0.85	1.95
		Drifted	118.2	27.5	81.6	2.11	4.32

6.7.3 Optimise right rail

Figure 6.26 illustrates the parameters in question for the changes done to the right rail. The middle three control points are locked from moving separately, but can move in the x-direction within the interval $x[m] =< -2, 2 >$ with a step of $0.01m$.

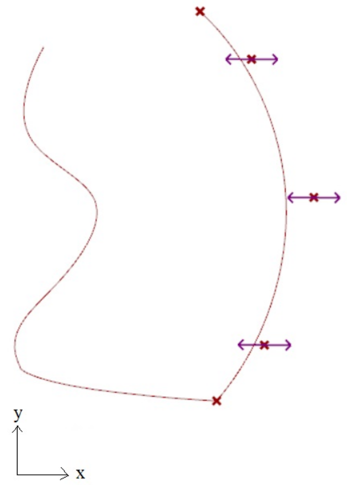


Figure 6.26: Shape optimisation: Three control points for the right rail are free to move in the x-direction, within the interval $x =< -2, 2 >$ meter.

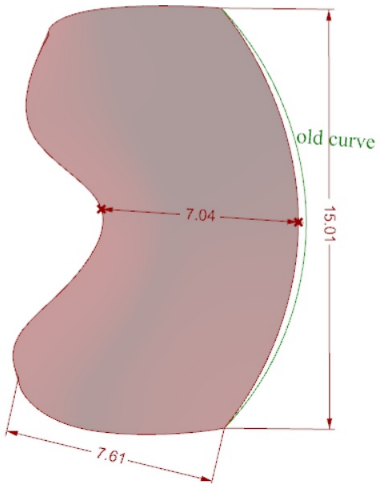


Figure 6.27: Shape after optimisation of the right rail. Green curve represents the curve belonging to *Shape 0*

The optimised position of the right rail was found by moving the control points approximately $0.3m$ to the left. Figure 6.27 illustrates the small movement from the old curve to the new position of the right rail. In Table 6.7 the new values for the optimised shape are presented.

Table 6.7: Structural behaviour of shape with optimised right rail, for both *Undrifted* and *Drifted* snow load case. Red values indicate increase and green values indicate decrease compared to *Shape 0*

	Displ.Res	*Load case*	Displ.x	Displ.y	Displ.z	M_y	M_z
	$[mm]$		$[mm]$	$[mm]$	$[mm]$	$[kNm]$	$[kNm]$
Right rail *optimise*	137.6	Undrifted	32.3	9.4	17.0	0.8	1.78
		Difference	-3.4	-0,8	-2.4	-0.05	-0.17
		Drifted	115.7	27.1	78.2	2.07	4.09
		Difference	-2.5	-0.4	-3.4	-0.04	-0.23

The rows for the *Undrifted* and *Drifted* load case in Table 6.7 represent the new values obtained for the optimised shape. The *Difference* between the optimised shape and *Shape 0* are marked with green values, which indicates improvements for the optimised shapes. The changes are though small both for the *Undrifted* and *Drifted* load case.

Figures 6.28 and 6.29 illustrate the total displacements of the two load cases. The legend on the right hand side describe the displacements, where red is the maximum value, and blue the minimum. The red area seems to be slightly smaller on the optimised shapes, which is the same as indicated in Table 6.7

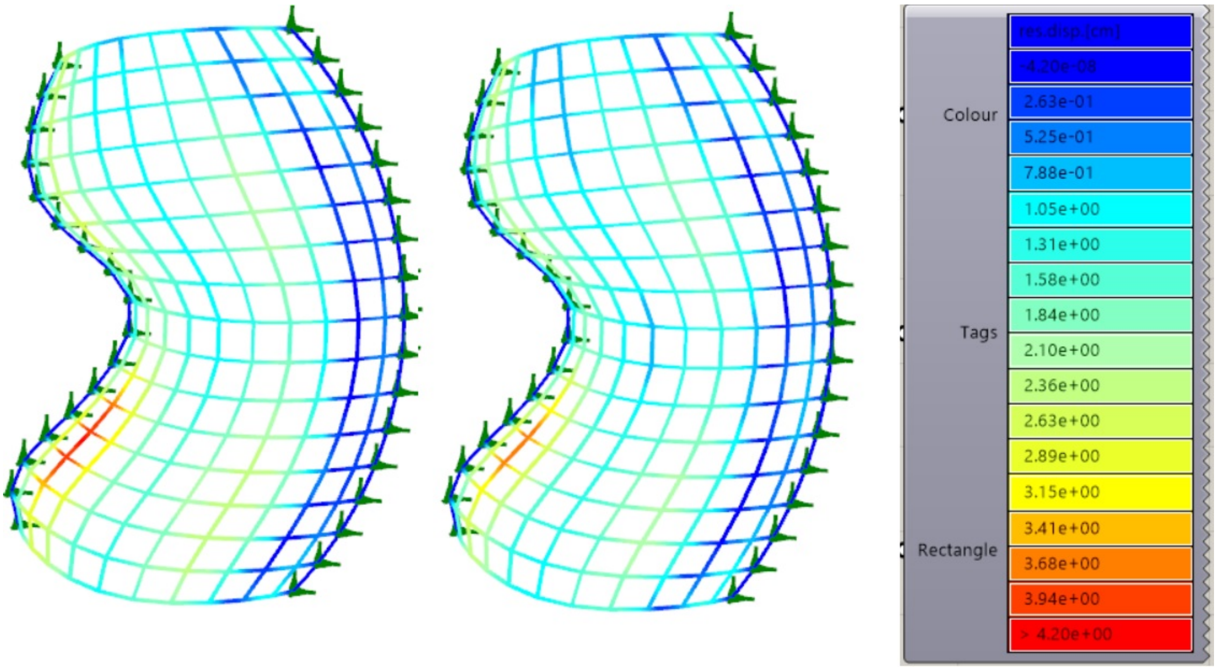


Figure 6.28: Right rail: Displacement of the structure when applied the *undrifted* snow load case. Left: *Shape 0*. Right: Optimised right rail

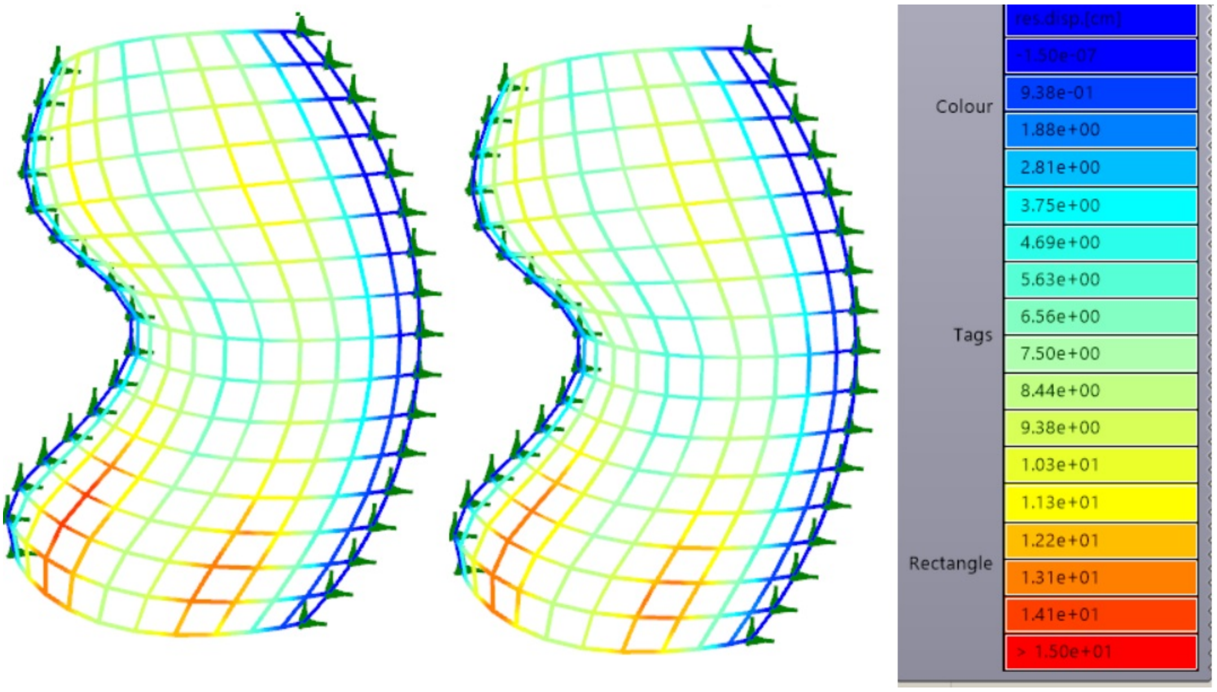
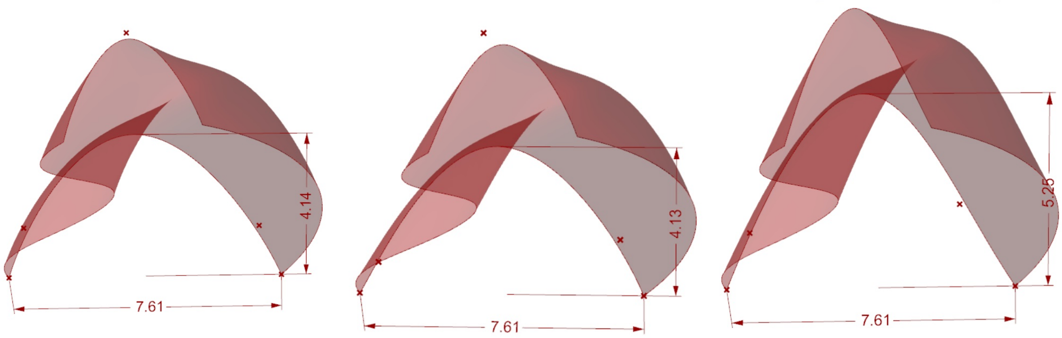


Figure 6.29: Right rail: Displacement of the structure when applied the *drifted* snow load case- Left: *Shape 0*. Right: Optimised right rail

6.7.4 Optimise section

Two optimisation analyses where done when considering the section curve. First the control points were restricted to move in the z-direction within the interval $z[m] =< -2, 2 >$ (see Fig. 6.30a). For the second analysis the control point could move both in the x- and z-direction (see Fig. 6.30b), with the same interval for z and $x[m] =< -1, 1 >$.

Figure 6.31 shows the optimised shapes (see Fig. 6.31b and 6.31c) compared to the original shape, *Shape 0*. When restricting the control points to only move in z-direction, the optimised shape ends up resembling *Shape 0* (see Fig. 6.31b). However, when the control points are free to move in both x- and z-direction the resulting shape appear more different (see Fig. 6.31c). Table 6.8 presents the displacements and moments for the two optimised shapes and the difference compared to *Shape 0*.

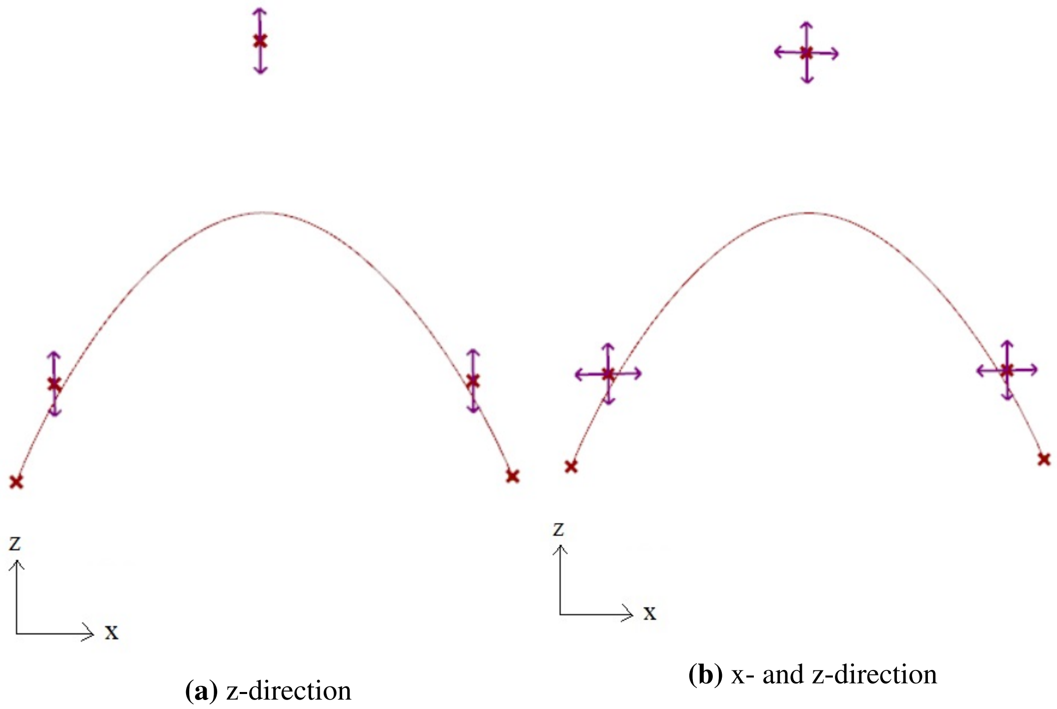


(a) *Shape 0*

(b) z-direction

(c) x- and z-direction

Figure 6.31: *Shape 0* vs. the shapes after optimising the section



(a) z-direction

(b) x- and z-direction

Figure 6.30: Shape optimisation: Three control points for the section free to move in the directions indicated. In x-direction the points can move within the $x =< -1, 1 >$ meter. In z-direction; $z =< -2, 2 >$ meter

Table 6.8: Structural behaviour of the shape with optimised section when parametric control points are free to move in either z-direction or x-and z-direction. Both *Undrifted* and *Drifted* snow load case is presented. Red values indicate increase and green values indicate decrease compared to *Shape 0*

	Displ.Res $[mm]$	*Load case*	Displ.x $[mm]$	Displ.y $[mm]$	Displ.z $[mm]$	M_y $[kNm]$	M_z $[kNm]$
z-dir. *optimise*	68.3	Undrifted	54.3	22.5	32.9	1.12	1.53
		Difference	+18.5	+12.3	+13.5	+0.28	-0.42
		Drifted	42.2	10.7	25.4	0.77	1.21
		Difference	-76	-16.8	-56.2	-1.34	-3.11
x-& z-dir. *optimise*	68.9	Undrifted	50.1	15.1	37.6	0.79	2.29
		Difference	+14.3	-4.9	+18.2	-0.06	+0.34
		Drifted	50.7	11.4	40.5	1.47	1.59
		Difference	-67.5	-16.2	-41.2	-0.64	-2.73

From Table 6.8 it is observed that both of the optimised shapes will improve the values in terms of lower displacements and moment for the *Drifted* load case. However, for the *Undrifted* load case, some of the displacements and moments show a small increase in values.

Figures 6.32 and 6.33 (representing the optimised shape for movement only in z-direction) substantiates this. The first (Fig 6.32) show larger red areas on the optimised shape indicating larger displacements for the *Undrifted* case, while the latter (Fig. 6.33) show the red areas disappear meaning the displacements decrease in the optimised shape for the *Drifted* case.

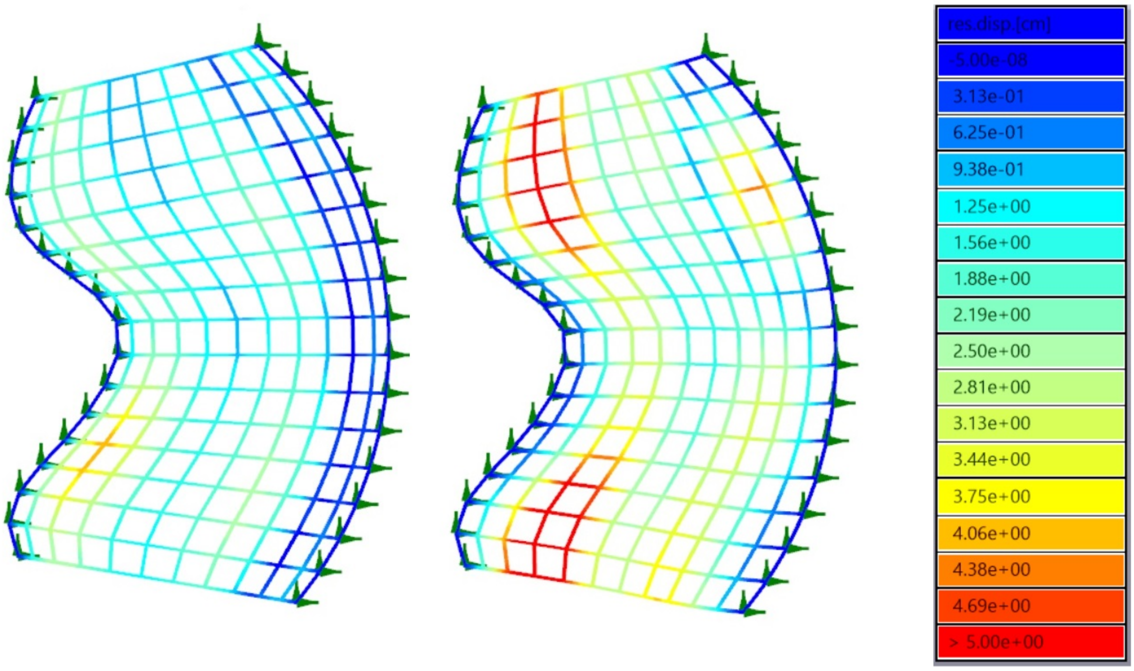


Figure 6.32: Section: Displacement of the structure when applied the *undrifted* snow load case. Left: *Shape 0*. Right: Optimised section (z-direction)

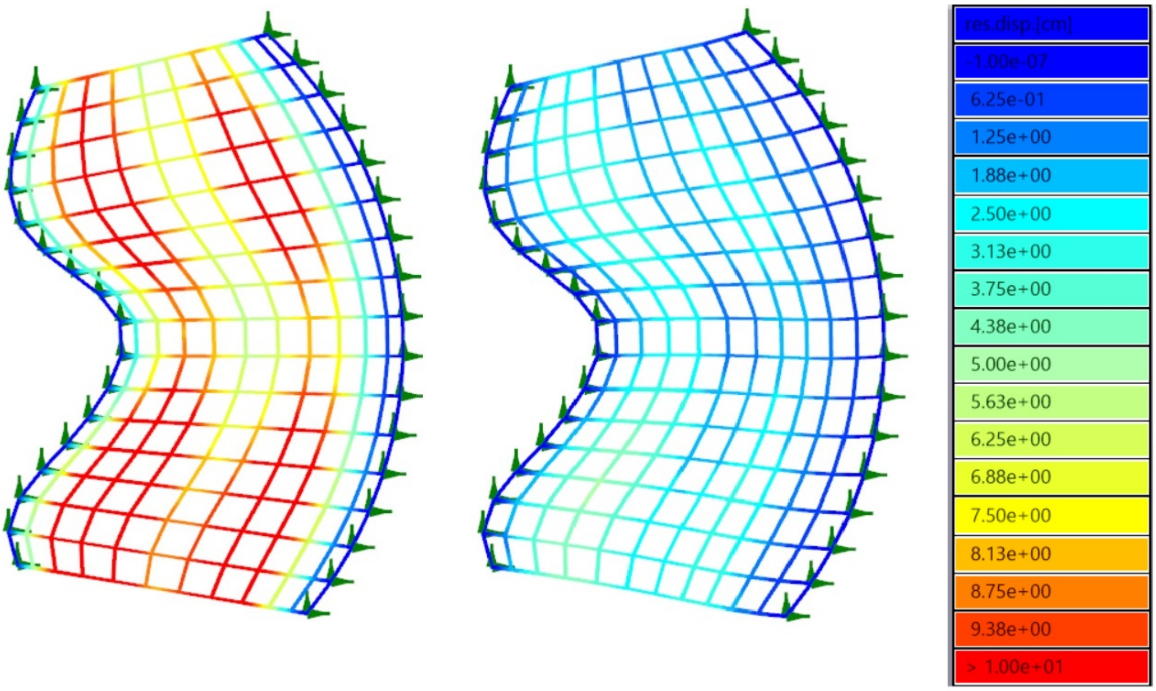


Figure 6.33: Section: Displacement of the structure when applied the *drifted* snow load case. Left: *Shape 0*. Right: Optimised section (z-direction)

6.7.5 Optimise left rail

Two shapes were found when considering the left rail, one for maximising displacements and one for minimising. The control points for the left rail were free to move separately in the x-direction (sideways) between the interval $x[m] =< -1, 1 >$ with a step of $0.01m$, except for the middle point which was assigned a bigger interval of $x[m] =< -2, 2 >$ (see Fig. 6.34).

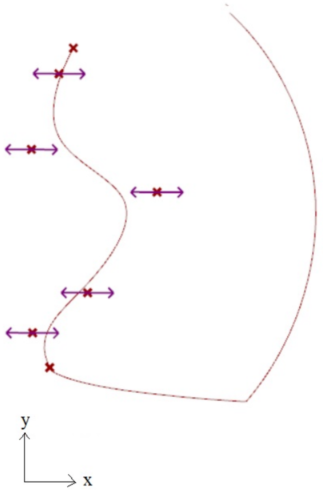
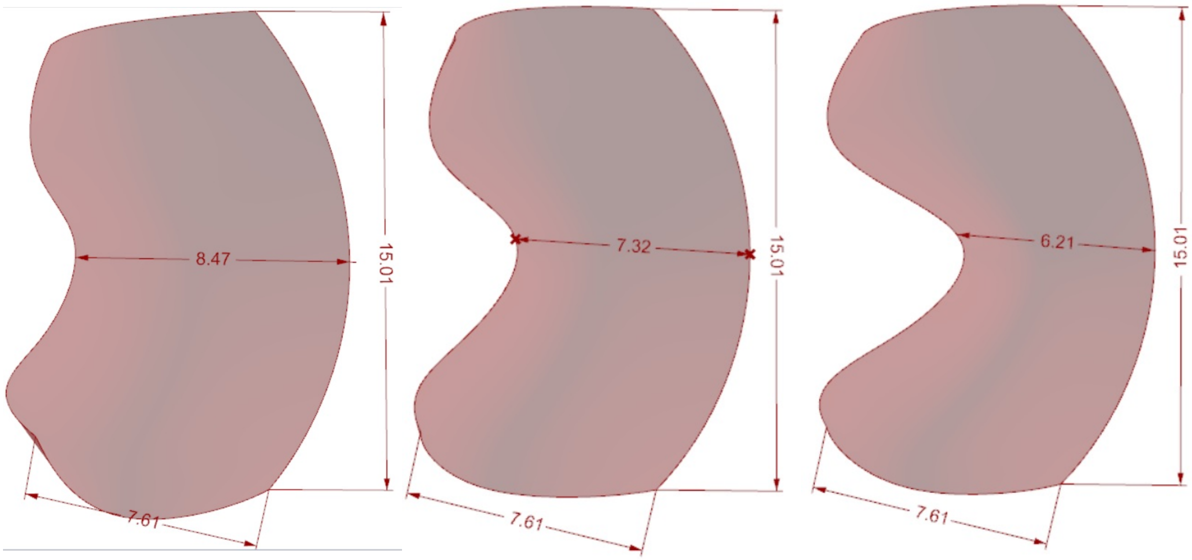


Figure 6.34: Shape optimisation: Five control points for the left rail free to move in the x-direction, within $x[m] =< -1, 1 >$. Except mid-point which can move within $x[m] =< -2, 2 >$



(a) maximised/weakened left rail

(b) *Shape 0*

(c) Optimised left rail

Figure 6.35: *Shape 0* vs. optimised and maximised(weakened) shape with respect to left rail

Figure 6.35 shows the shapes obtained when moving the control points of the left rail to maximising and minimising the displacements with Galapagos. The optimised shape (Fig. 6.35c) displays a distinct increase in the curvature of the shape, i.e. it becomes more doubly curved. Note that the increase in curvature also leads to a shortening of the span. The weakened shape (Fig. 6.35a), on the other hand, shows a lesser curvature and an increase in span, where the left rail seems to straighten up.

Table 6.8 presents the displacements and moments for the optimised shape and the difference compared to *Shape 0*. The shape found with maximising displacements is also presented in the table for comparison.

Table 6.9: Structural behaviour of the shape with optimised left rail and for the shape found by maximising displacements. Both *Undrifted* and *Drifted* snow load case is presented. Red values indicate increase and green values indicate decrease compared to *Shape 0*

	Displ.Res $[mm]$	*Load case*	Displ.x $[mm]$	Displ.y $[mm]$	Displ.z $[mm]$	M_y $[kNm]$	M_z $[kNm]$
Left rail	224.5	Undrifted	30.57	9.17	23.61	0.77	1.82
		Difference	-5.13	-1.03	+4.22	-0.08	-0.13
maximise		Drifted	184.6	12.7	135	2.86	6.87
		Difference	+66.4	-14.8	+53.4	+0.75	+2.55
		Undrifted	32.8	14.7	17.4	0.79	1.83
Left rail	108.3	Difference	-2.9	4.5	-2.0	-0.06	-0.12
optimise		Drifted	88.2	34.2	62.8	1.85	3.46
		Difference	-30	6.7	-18.8	-0.26	-0.86

As presented in Table 6.7, the values found after optimising the left rail show improvements both for the *Undrifted* and *Drifted* load case. The exception is the displacement in y-direction, which seems to have increased a few millimeters. The largest changes are seen for the *Drifted* case for the displacements in x- and z-direction, where the values have decreased with $3cm$ for the first and approximately $2cm$ for the latter.

The maximised shape show lower values for the displacement in y-direction, implying that a shape similar to a barrel vault is more stable in the given direction (see Sec. 6.6 *Initial study of the shape*), compared to a shape with more double curvature (*Shape 0* and optimised left rail). The rest of the values will increase for the *Drifted* load case.

Figure 6.36 shows how the *Undrifted* load case only show small changes in the total displacement for the three shapes (the colour shades are quite similar). For the *Drifted* case (Fig. 6.37 , the decreasing displacement is apparent with the red zones in the maximised shape, and the blue zones in the two other.

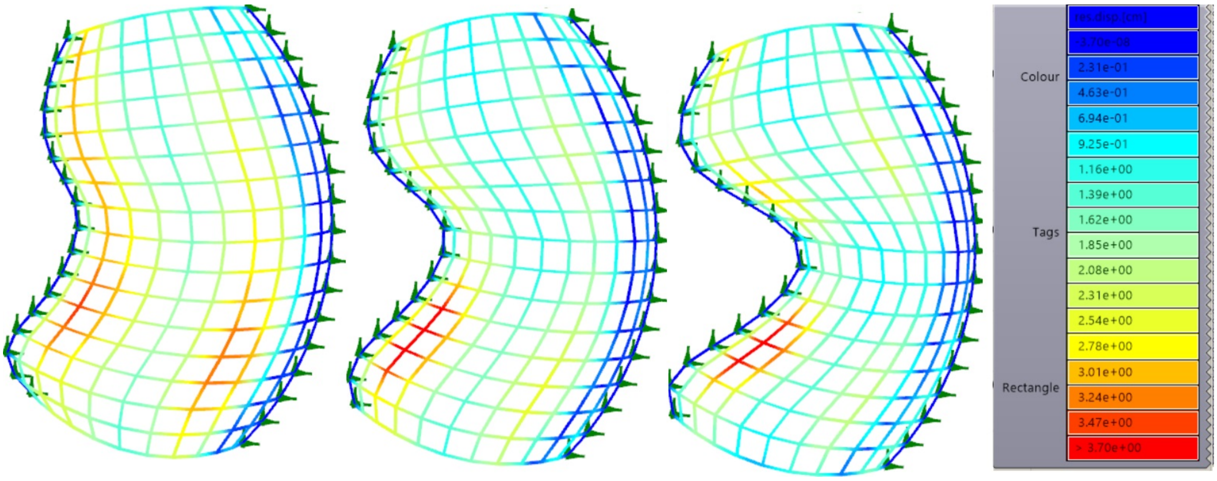


Figure 6.36: Left rail: Displacement of the structure when applied the *undrifted* snow load case, Left: Maximised left rail, middle: *Shape 0* right: Optimised left rail

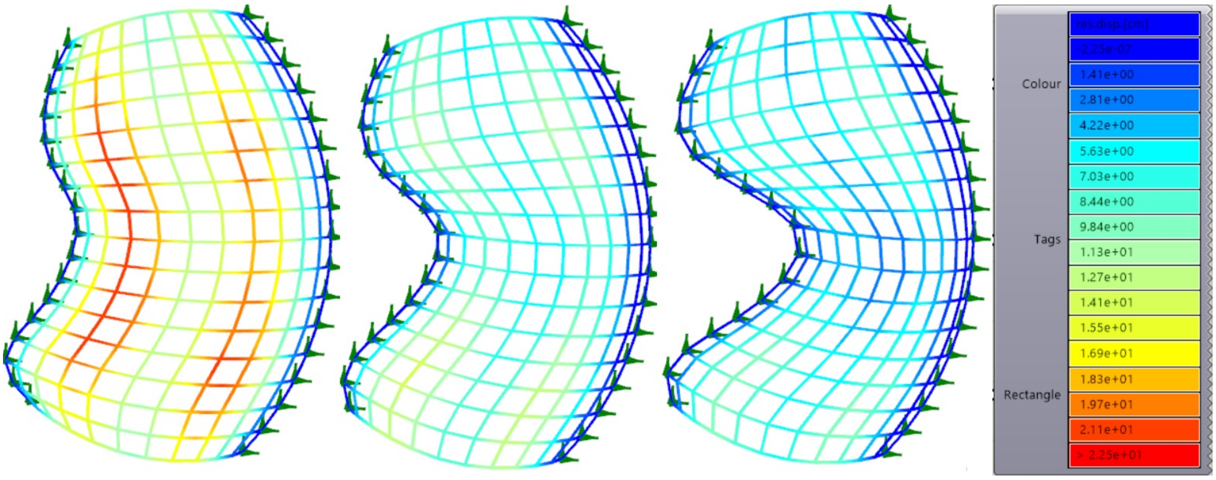
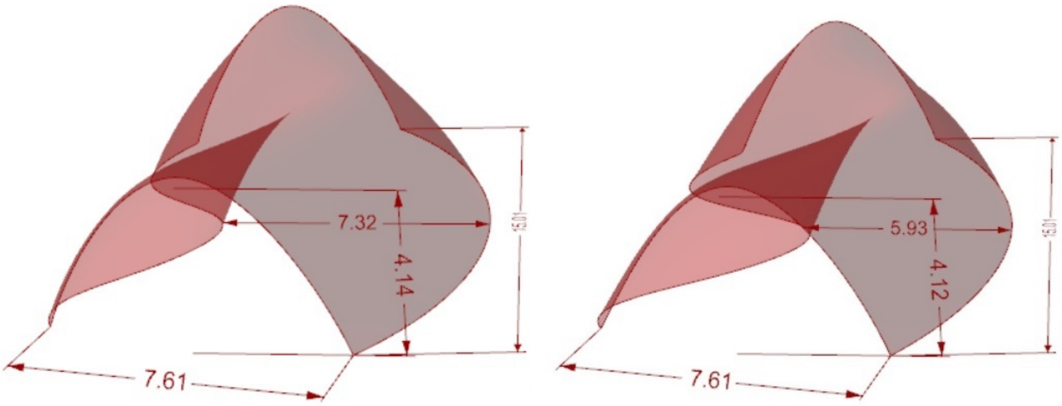


Figure 6.37: Left rail: Displacement of the structure when applied the *drifted* snow load case, Left: Maximised left rail, middle: *Shape 0* right: Optimised left rail

6.7.6 *Shape 1* - Combine optimisations

Figure 6.38 shows the resulting shape (*Shape 1*), and Table 6.10 presents the resulting values when combining the optimised section and rails found in Sections 6.7.3-6.7.5 above.



(a) *Shape 0*

(b) *Shape 1*: combined optimisations

Figure 6.38: *Shape 0* and the resulting *Shape 1* when combining the different optimisations (right rail, left rail and section)

Table 6.10: Structural behaviour of the shape with combined optimisations - *Shape 1*, for *Undrifted* and *Drifted* snow load case is presented. Red values indicate increase and green values indicate decrease compared to *Shape 0*

	Displ.Res $[mm]$	*Load case*	Displ.x $[mm]$	Displ.y $[mm]$	Displ.z $[mm]$	M_y $[kNm]$	M_z $[kNm]$
Shape 1 combined results	64.8	Undrifted	56.7	20.4	28.8	1.1	1.93
		Difference	21	10.2	9.4	0.25	-0.02
		Drifted	32.8	13.2	32.4	1.62	1.57
		Difference	-85.4	-14.3	-49.2	-0.49	-2.75

From Table 6.10 it is evident how the optimisation process has improved the *Drifted* load case rather than the *Undrifted* case. This is due to the optimisation approach using Galapagos (see Sec. 6.7.2), where it is the maximum nodal displacement of the two load cases being minimised. As seen in the initial results of *Shape 0* (see Tab. 6.6), the *Drifted* load case gives the largest values, i.e. the optimisation with Galapagos is hence minimising the displacements of this load situation, which does not necessarily improve the *Undrifted* load case.

Figures 6.39 and 6.40 illustrates the total displacement for the two load cases applied to *Shape 0* (left) and *Shape 1* (right). The *Undrifted* case shows larger displacements in *Shape 1*, while the *Drifted* case is improved.

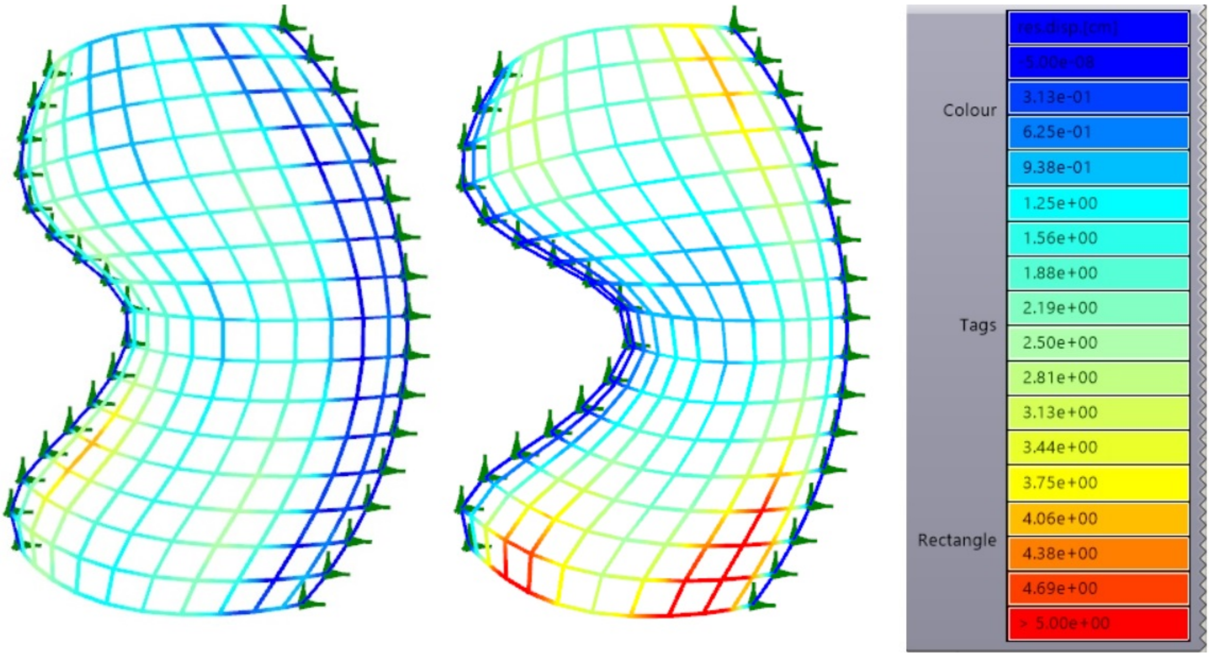


Figure 6.39: Combined optimisation: Displacement of the structure when applied the *undrifted* snow load case, Left: *Shape 0* right: *Shape 1*

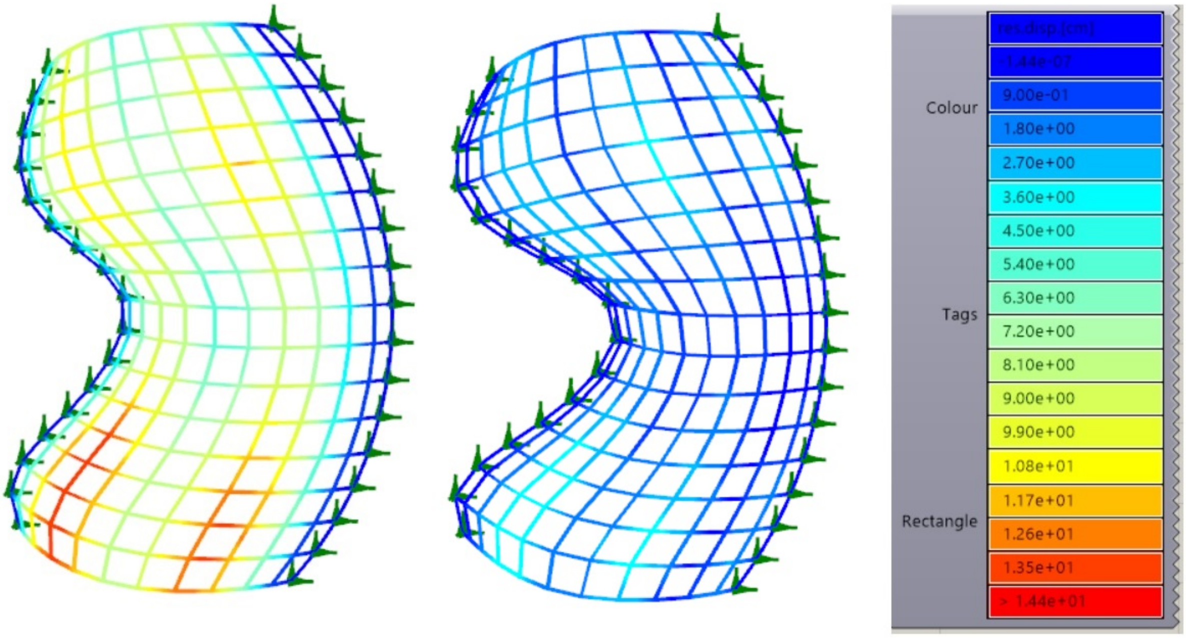


Figure 6.40: Combined optimisation: Displacement of the structure when applied the *drifted* snow load case, Left: *Shape 0* right: *Shape 1*

Comparison of *Shape 0* and *Shape 1*

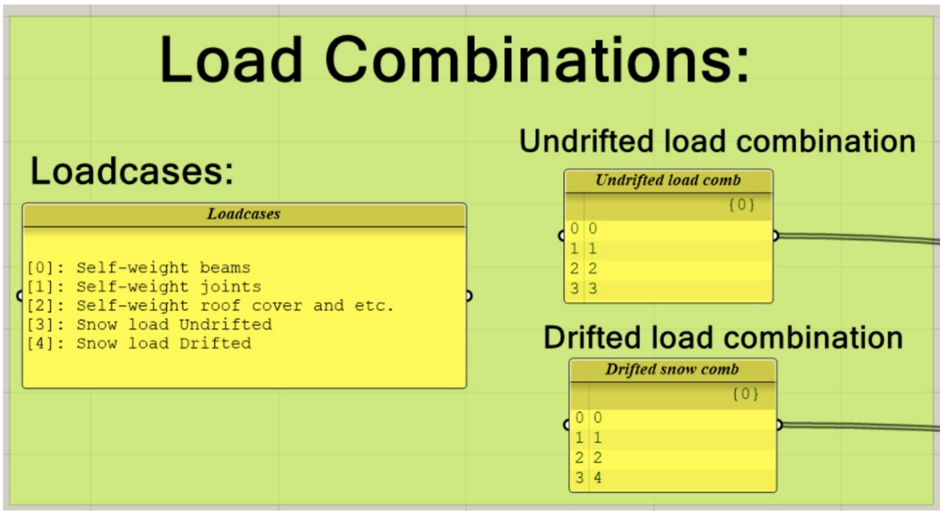
A stable shape or structure can be thought of as a structure which has the ability to handle multiple load situations. Given that the values for *Shape 0* displayed quite large displacements for the *Drifted* load case compared to the *Undrifted*, the results in Table 6.10 are in fact an improvement seen in total.

The combination of the positive (red) difference for the *Undrifted* case and negative (green) for the *Drifted* (shown in Table 6.10) gives a shape which is more robust compared to *Shape 0*, as the displacements and moments for the two load cases are more equal. *Shape 1* will in other words have a similar structural response for the two different load cases, which is a positive feature.

The optimised shape will appear more narrow in the middle of the structure. The floor area of *Shape 1* is approximately 106 m^2 compared to 115 m^2 for *Shape 0*. Thus, the optimisation will decrease the available space inside the cabin.

6.8 Design of the grid shell structure, *Shape 1* - cross section

Tables 6.6-6.10 illustrate the displacements and moments before the design process of the appropriate cross section, and the structure is applied either *Undrifted* or *Drifted* snow load. When designing the structure, which will be the aim of this section, all relevant loads need to be considered. This includes the following loads and load combinations presented in Figure 6.41:



Load Combinations:

Loadcases:

Loadcases

```
[0]: Self-weight beams
[1]: Self-weight joints
[2]: Self-weight roof cover and etc.
[3]: Snow load Undrifted
[4]: Snow load Drifted
```

Undrifted load combination

Undrifted load comb

```
{0}
0 0
1 1
2 2
3 3
```

Drifted load combination

Drifted snow comb

```
{0}
0 0
1 1
2 2
3 4
```

Figure 6.41: Loads and load combinations used in the design of the structure.

Table 6.11 presents the resulting maximum forces and displacements for *Shape 0* and *Shape 1* for both load combinations with cross section $80mmx80mm$.

Table 6.11: Structural behaviour for *Shape 0* and *Shape 1* with cross section $80mmx80mm$ for *Undrifted* and *Drifted* snow combination. Red values indicate increase and green values indicate decrease compared to *Shape 0*

	Shape 0		Shape 1 (optimised)			
Load combinations:	*Undrifted*	*Drifted*	*Undrifted*	Diff.	*Drifted*	Diff.
$N_{compression}[kN]$	-27.90	-37.60	-29.09	1.19	-29.23	-8.37
$N_{tension}[kN]$	5.60	10.70	8.93	3.33	10.15	-8.37
$M_y[kNm]$	0.88	2.18	1.12	0.24	1.66	-0.52
$M_z[kNm]$	2.08	4.45	2.32	0.24	1.66	-2.79
$Displ.x[mm]$	-38.2	-125.0	62.8	24.6	29.6	-95.4
$Displ.y$ [mm]	10.9	28.9	-24.9	14.0	10.6	-18.3
$Displ.z$ [mm]	-20.9	-87.1	-35.5	14.6	-32.6	- 54.5
$Totaldispl.[mm]$	46.7	151.9	71.3	24.6	53.4	-98.5

The necessary cross section for *Shape 1* will be determined based on the following requirements:

- Minimum displacement of (L/250) in all directions (SLS)
- Utilisation of the beam elements < 1 (ULS)

- Minimising mass $[kg]$ of structure

The utilisation is checked for load combinations according to Ultimate Limit State (ULS), and the displacement for combinations according to Service Limit State (SLS) (see Sec 6.4). The flow diagram for the design process is presented in Figure 6.42, below:

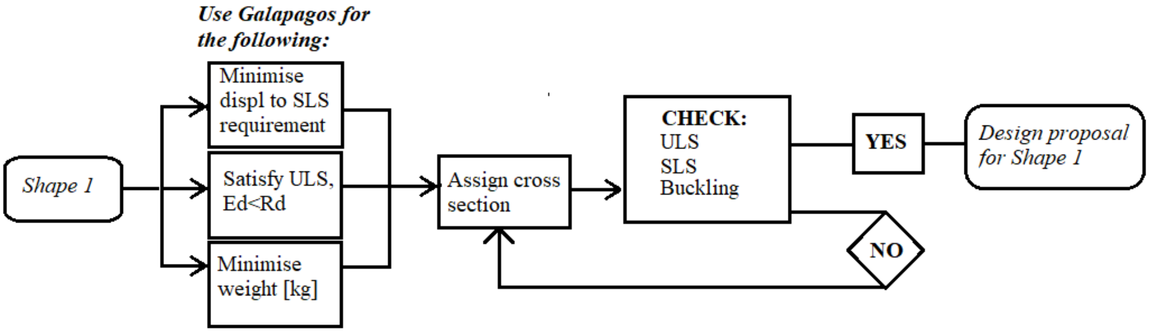


Figure 6.42: Flow diagram for the design process of the cross section for *Shape 1*, from arbitrary cross section to design cross section satisfying ULS, SLS and buckling.

According to Table NA.A1 (904) in [39] the recommended value for the highest permitted deflection is between L/200 and L/250, where L is the span. The span is measured at three places and conservatively assumed to be the smallest of the measured values. For the cabin roof (*Shape 1*) the span is assumed to be approximately $6m$.

$$L/250 = 6000/250 = 24mm$$

A deflection (displacement) of $24mm$ is hence chosen as the maximum requirement. The utilisation of all beam elements will be found according to Eurocode 5 ([46]), and limited to 1. See Appendix D for detailed utilisation calculations. The following checks in [46] is performed (see Fig. 6.43):

- (6.1.2) Tension parallel to grain direction
- (6.1.4) Compression parallel to grain direction
- (6.1.6) Combined bending about local y- and z-axis
- (6.1.7) Shear
- (6.2.3) Combined bending about local y- and z-axis, and axial tension
- (6.2.4) Combined bending about local y- and z-axis, and axial compression
- (6.3.2) Local buckling of members. Assumed buckling length: $0.5L$, where L is the length of each beam member.

The following example illustrates the requirements for combined bending about local y- and z-axis, and axial compression (Eq. 19 and 20 in [46]):

$$(\frac{\sigma_{c,0,d}}{f_{c,0,d}})^2 + \frac{\sigma_{m,y,d}}{f_{m,y,d}} + k_m \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \tag{6.4}$$

$$(\frac{\sigma_{c,0,d}}{f_{c,0,d}})^2 + k_m \frac{\sigma_{m,y,d}}{f_{m,y,d}} + \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \tag{6.5}$$

- k_m is a factor that is set to 0.7 for rectangular cross sections [6.1.6(2)]
- $\sigma_{m,y,d}$ & $\sigma_{m,z,d}$ are the design bending stresses about the principle axes shown in Figure 4.23, where: $\sigma_{m,y,d} = \frac{M_{y,d}}{W_z}$ and $\sigma_{m,z,d} = \frac{M_{z,d}}{W_y}$
- $\sigma_{c,0,d}$ is the design compressive stress along the grain: $\sigma_{c,0,d} = \frac{N_d}{A}$

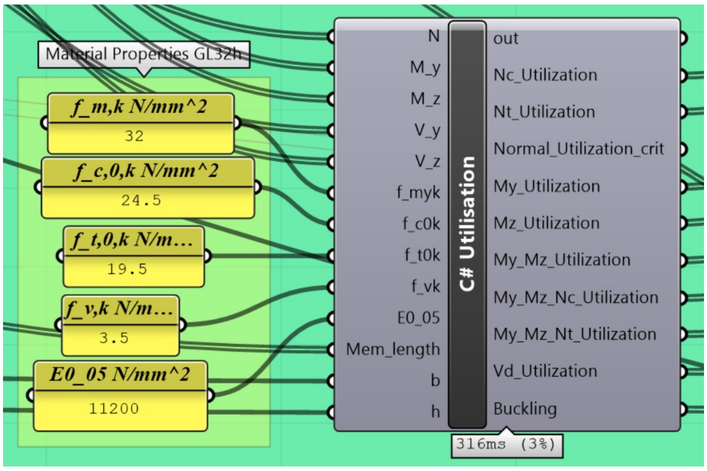
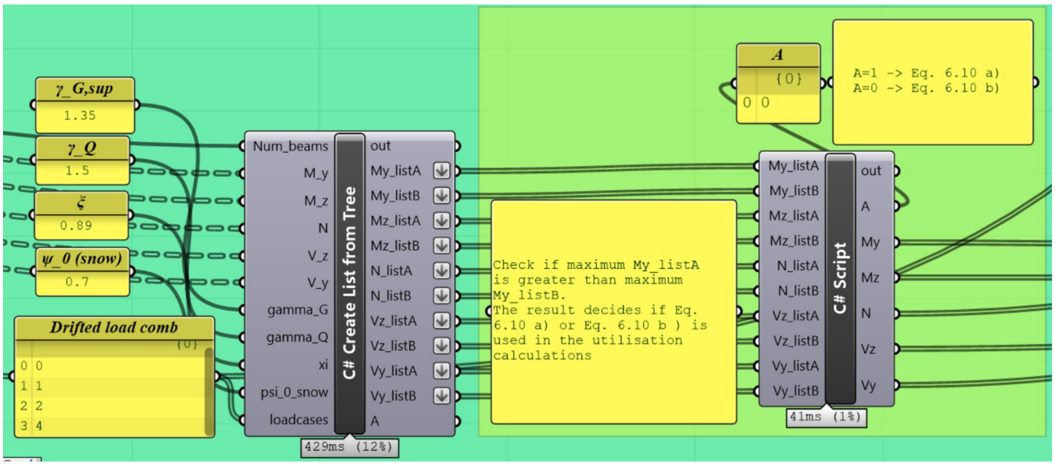


Figure 6.43: C#-component for calculation the utilisation of each beam member

The $BeamForces(Karamba)$-component and C# is used to extract the forces in the beams, which includes N, M_y, M_z, V_z and V_y. Karamba gives two measured values per beam element, and thus to be on the conservative side the the maximum of the two is evaluated for each beam. This means that the resulting design force is not guaranteed to be occurring in the same section of the beam element. A C# code is used for establishing the correct load combinations and calculating the utilisation of each beam element (see Figure 6.44).



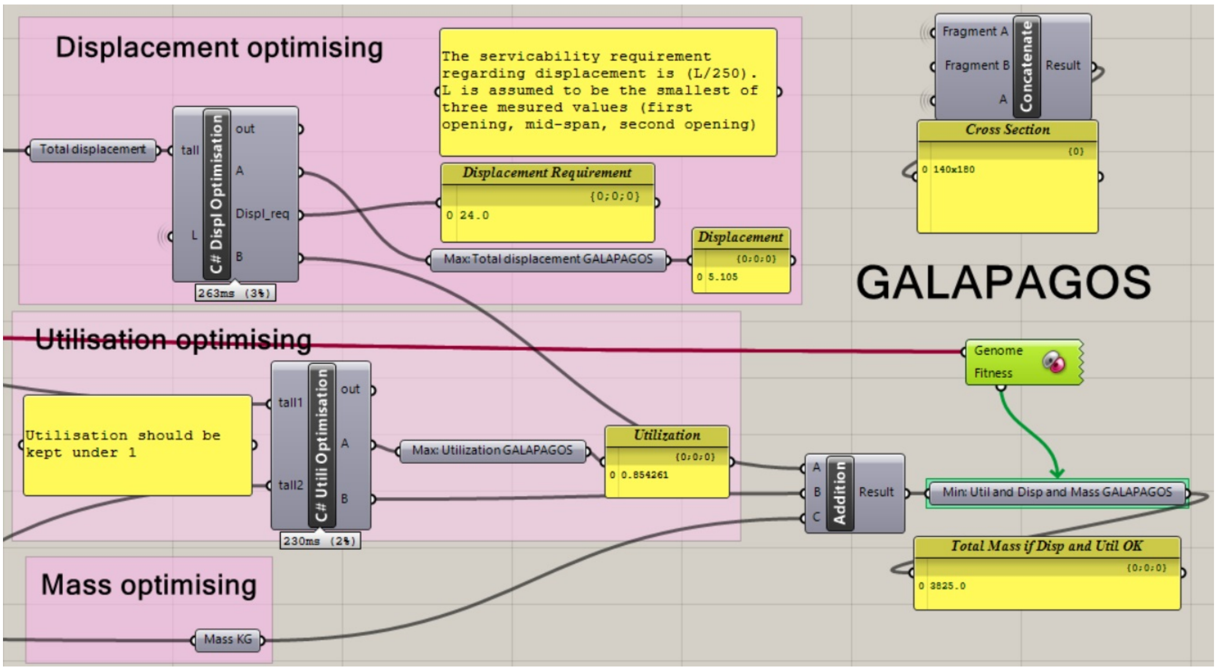
```
A=1 -> Eq. 6.10 a)
A=0 -> Eq. 6.10 b)
```

```
Check if maximum My_listA
is greater than maximum
My_listB.
The result decides if Eq.
6.10 a) or Eq. 6.10 b ) is
used in the utilisation
calculations
```

Figure 6.44: C# -components for establishing ULS load combinations (Eq. 6.10a) and 6.10b)) and hence determining the design loads. The C#-code (see the component on the right) returns the ULS combination giving the highest design load.

Galapagos is used to find the necessary height and width of the cross section satisfying both SLS and ULS, i.e. the displacement is below $24mm$ and the utilisation is below 1. The

Galapagos solver is allowed to switch between the available heights and widths presented in 6.3. In addition the mass of the structure is minimised. The fitness selected for Galapagos to minimise is a combination of numbers representing displacement, utilisation and mass (see Fig. 6.45). For both the utilisation and displacement the number is set to 0 if the result is within the limits and 1000000 if the result is above the limit. This way Galapagos will always avoid choosing cross sections that does not satisfy the requirements. Minimising the mass ensures that the solver does not end up with a massive cross section (see Appendix C for more details from the optimisation code).



The servicability requirement regarding displacement is (L/250). L is assumed to be the smallest of three mesured values (first opening, mid-span, second opening)

Utilisation should be kept under 1

Figure 6.45: Illustration of the optimisation code established in GH. Galapagos is used to optimise the cross section for *Shape 1* with respect to satisfying ULS and SLS requirements as well as minimising mass $[kg]$.

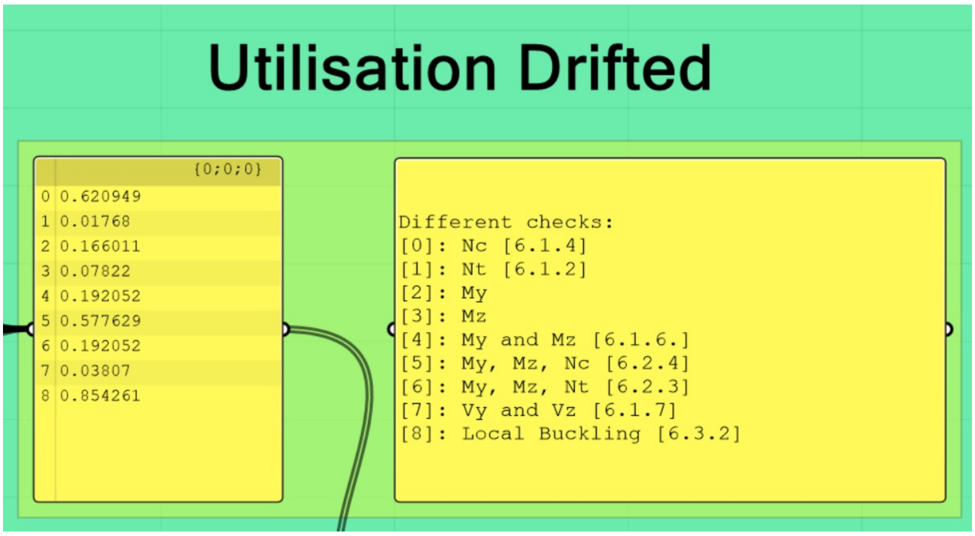
6.8.1 Results from the design of the grid shell, *Shape 1* - cross section

From an initial stage it was established that the total displacement will always be below $2.4cm$ when the utilisation check is OK. Secondly it was found that the utilisation check which gave the most critical results was in general local buckling (Eq. 6.23 and Eq. 6.24 in [46]). In the local buckling check the beam elements were assumed to have a buckling length equal to $0.5L$ as the joints were assumed to be fixed.

Table 6.12 shows the cross section optimised and designed for *Shape 1*, and the resulting maximum forces and displacements for *Shape 0* and *Shape 1* for both load combinations.

Table 6.12: Structural behaviour for *Shape 0* and *Shape 1* with the designed cross section $140mmx180mm$ when applied *Undrifted* and *Drifted* snow load combinations. Red values indicate increase and green values indicate decrease compared to *Shape 0*

	Shape 0		*Shape 1* (optimised)			
Load combinations:	*Undrifted*	*Drifted*	*Undrifted*	Diff.	*Drifted*	Diff.
Cross section	$140mmx180mm$		$140mmx180mm$		$140mmx180mm$	
Utilisation	0.74	0.88	0.76	0.02	0.80	-0.08
$N_{compression}[kN]$	-28.43	-31.22	-29.36	0.93	-29.47	-1.75
$N_{tension}[kN]$	2.41	6.44	5.34	2.93	5.02	-1.42
$M_y[kNm]$	1.03	2.47	1.26	0.23	1.80	-0.67
$M_z[kNm]$	1.75	3.94	2.16	0.41	1.17	-2.77
$Displ.x[mm]$	-2.4	-8.4	4.1	1.7	1.5	-6.9
$Displ.y[mm]$	0.6	1.7	-1.9	1.3	0.5	-1.2
$Displ.z[mm]$	-1.7	-5.9	-2.9	1.2	-2.4	-3.5
$Totaldispl.[mm]$	3	10.1	5.1	2.1	3.7	-6.4



Utilisation Drifted

```
{0;0;0}
0 0.620949
1 0.01768
2 0.166011
3 0.07822
4 0.192052
5 0.577629
6 0.192052
7 0.03807
8 0.854261
```

```
Different checks:
[0]: Nc [6.1.4]
[1]: Nt [6.1.2]
[2]: My
[3]: Mz
[4]: My and Mz [6.1.6.]
[5]: My, Mz, Nc [6.2.4]
[6]: My, Mz, Nt [6.2.3]
[7]: Vy and Vz [6.1.7]
[8]: Local Buckling [6.3.2]
```

Figure 6.46: Utilisation results obtained in GH with Galapagos. Example shows the most critical beam member in *Shape 1* for the *Drifted* load combination.

Table 6.12 shows how the displacements, internal beam forces and thus utilisation will decrease for *Shape 1* compared to *Shape 0* when the *Drifted* load combination is applied (see green numbers). The decrease for the *Drifted* load combination for *Shape 1* illustrates how a smaller cross section could in principle be adopted for this shape compared to the initial one to obtain the same degree of utilisation. As there are limited alternatives for the beam sizes and cross sections on the market however, the same cross sections are still adopted. The utilisation was below 1 and hence OK for both shapes and both load combinations.

6.8.2 Buckling check

A buckling analysis was done for the grid shell with the resulting cross section $140mmx180mm$ obtained through optimisation with Galapagos (see Fig. 6.45). The buckling load factors for the *Undrifted* and *Drifted* load combinations (see Fig. 6.41) were found, and the results are presented in Table 6.13.

Table 6.13: Resulting buckling load factors for the timber grid shell with cross section $140mmx180mm$ and the corresponding maximum compression force $N_{compression}$ $[kN]$ for the different load combinations.

	Undrifted load comb	*Drifted load comb.*
$N_{compression}$ $[kN]$	-29.7	-29.3
BLFac	18.0	20.6

As seen from Table 6.13 the structure will have a high buckling load factor and will therefore be considered safe with regards to buckling under the given load combinations. Figure 6.47 illustrates the 1st buckling mode for the *Undrifted* load combination.

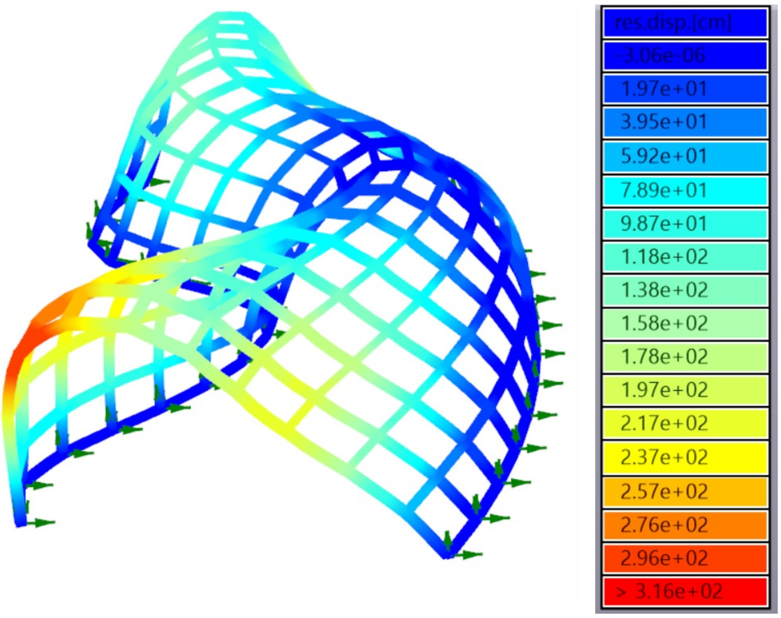
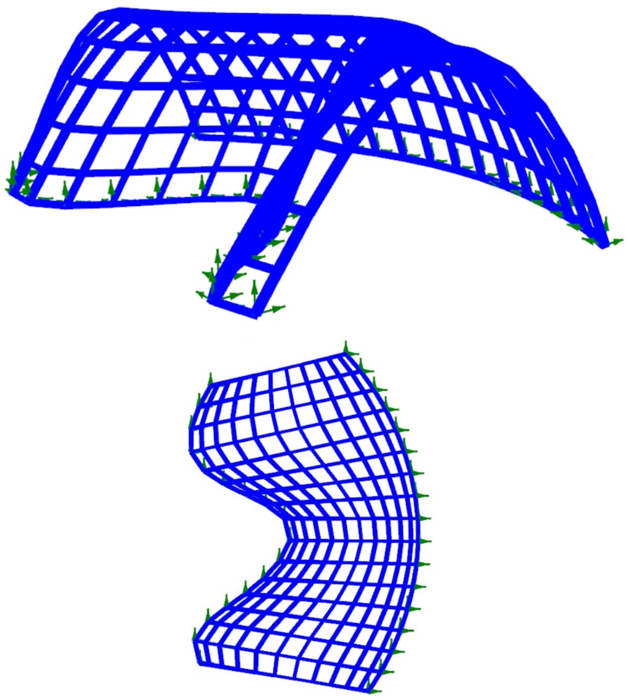


Figure 6.47: 1st buckling mode for the timber grid shell (*Shape 1*) under the *Undrifted* load combination. The legend on the right illustrates the corresponding displacements.

6.9 Design proposal for the NTNUI cabin

After optimising the initial *Shape 0* in Section 6.7 resulting in *Shape 1*, a suitable cross section was obtained in Section 6.8 and hence the following design proposal in Table 6.14 is given.

Table 6.14: Design proposal for *Shape 1* for the NTNUI timber grid shell. With the selected cross section, the ULS, SLS and buckling requirements are satisfactory for both the *Drifted* and *Undrifted* load combinations.

Shape 1:		
Material	GL32c	
Cross Section	$140mmx180mm$	
Load combinations	*Undrifted*	*Drifted*
Requirements:		
ULS Utilisation	$0.77(OK!)$	$0.85(OK!)$
SLS Deflection	$5.1mm(OK!)$	$3.7mm(OK!)$
Buckling	$27.1(OK!)$	$23.4(OK!)$

6.10 Discussion

The design proposal in Table 6.14 displays a structure with a design resistance satisfying the requirements of ULS, SLS and buckling. Nevertheless there are some observations regarding the design and optimisation process in this chapter which should be mentioned.

- For *Shape 0* the most critical load situation with regards to displacements and internal beam forces was the asymmetric *Drifted* load combination, and the *Undrifted* combination was a lot less critical. The optimised *Shape 1* displayed improvements in structural performance for this load combination, but an increase in forces and displacements for the *Undrifted* one. This is however favourable as the configuration of *Shape 1* leads to a smaller gap between the structural response for the two load combinations.
- The changes in structural performance between *Shape 0* and *Shape 1* are quite small for cross section $140mmx180mm$ (see Tab. 6.12). For the *Drifted* load combination the displacements improved by a maximum of approximately $7mm$ (sideways displacement in x-direction), and the moments with 2.8 kNm (M_z). It must however be noted how these values do display noticeable changes if the percentage of decrease is considered. The maximum total displacement due to the *Drifted* load combination decreased with 63% compared to the displacement for *Shape 0*. The maximum moments M_y and M_z decreased with 27% and 70%, respectively.
- The displacements for *Shape 1* were displayed visually in GH, and the structure seemed to have some local weaknesses in one of the openings (see red area for *Shape 1* in Fig. 6.39 and Fig. 4.51). This could have been prevented by manipulating the control points of the left rail also in the direction parallel with the structure (y-direction, see Fig 6.34). The result would most likely be an increase in the double curvature in the entrance part of the structure. This would have been beneficial as it would resemble the back of the structure, which showed to the more stable part of the structure with respect to displacements (see Fig. 6.47).
- The control points in the optimisation process were only allowed to move within a small interval to maintain the same size of the cabin. The changes did however result in a *Shape 1* with a noticeably smaller floor area than the initial one and should thus be checked to ensure the available floor area is sufficient for the cabin's type of use.

In conclusion, it would have been preferred to optimise the shape multiple times in an iterative manner before the final design proposal was set. The optimisation process should be a loop of improved shapes to be able to consider all components, and build a structure both globally and locally robust.

7 | Concluding remarks

Through the work of this master thesis our knowledge about parametric modelling and the design of shell structures has grown. We are left with a better understanding of the process of designing a structure with the use of parametric software, as well as having learned some of the opportunities and challenges connected with it. In the following sections below, we will present some lessons learned and further thoughts.

Design actions on shell structures according to Eurocode

- EC provides limited information regarding shell roofs - difficult to determine the distribution of wind and snow loads.
- The information provided in EC lacks detailed guidelines, hence deriving information cannot be done with great reliability.
- Snow: Parallels from the cylindrical shape presented in [35] can be drawn to similar shell shapes, however upper limits and details about the snow distribution on the surface of the shape are lacking.
- Wind: Only one figure for each roof shape (barrel vault and dome) is given in [36] to obtain the pressure coefficient and describe the wind distribution. The information lacks detail and clarity.
- Experiments and/or simulations are needed to verify the design loads and their distribution on shell roofs. This could be carried out using drifted artificial snow in a wind tunnel and wind simulations such as CFD (Computational Fluid Dynamics).

Shell shapes

- A dome is a distinctly efficient structure. This is as a result of its doubly curved shape facilitating a smooth flow of forces, the aforementioned being the main feature of a great shell structure.
- The singularly curved barrel vault displayed a structure more prone to deflect compared to the dome. This led to higher moments, normal forces and displacements.
- If the geometric barrel vault is substituted by a free-form structure resembling the barrel vault with a double curvature, the structural capacity will increase noticeably coupled with decreasing moments and displacements.
- A curved shape structure can withstand vertical load with primarily membrane forces, consequently it is efficient in resisting deformation. A flat surface will resist the vertical load solely with bending action, deducing that shorter spans and heavier structures are necessary to limit the moments and displacements.

Grid shell structures

- The reticulated shell can be treated as a shell, equivalent to the continuous concrete shell. Their structural behaviours regarding displacement and moments were found to be relatively similar.
- Environmental friendly structure: Timber is a green material, efficiency of a grid shell leads to low use of material. In addition, wood from local forests allows for shorter transport distances.
- Light-weight structure: Important factor to consider when the structure is located in secluded locations, with challenges related to transportation.
- The combination of an unrestrained timber grid shell and the possibility of prefabricated precision connected to the parametric software, would facilitate a manageable assembly of the grid shell.

Asymmetric loads on shell structures

- A load configuration-algorithm was established in GH to find the load distribution displaying maximum displacements and moments. The Galapagos evolutionary solver was implemented, where the result gave asymmetric load distributions for all cases.
- The load distributions obtained with Galapagos result in a more critical load situation for the grid shell compared with the EC load distributions. However, the plausibility of the given load situations is questionable.
- Implementing the loads from EC, the asymmetric *Drifted* load case is generally worse for the structure when compared to the symmetric *Undrifted* load case.

Optimisation with Galapagos evolutionary solver

- It was found to be challenging to optimise a structure to handle various load situations in order to increase its overall structural capacity.
- The optimisation process with Galapagos performed in this thesis has been carried out based on minimising displacement. Minimising the displacements caused a decrease in moments and normal forces.
- Optimisation of the shape with regards to minimising displacements improved the shape's capability to handle the load case with highest displacements. When minimising displacements for load combinations with the *Undrifted* and *Drifted* snow, the shape was directly optimised for the latter load situation with the *Drifted* case initially giving the largest displacements. The *Undrifted* load combination, on the other hand, displayed an increase in displacement after optimisation.
- The optimisation approach did however decrease the gap between the structural response for the two load combinations, i.e. more equal displacements and moments after the optimisation.

Parametric design workflow

- We learned how it is more time consuming and complicated to generate a parametric model when compared to a traditional CAD design. Instead of simply drawing the geometry, programming skills are required to establish the geometry for the model to become parametric. Adopting parametric software for structural design requires interest in learning and developing these skills.

- You give and you gain. Once the parametric model is created, the design is flexible and iterative steps can easily be made. By modifying a parameter the model will adapt accordingly, whereas a traditional CAD software will demand redrawing and remaking the model. Thus making modifications more time consuming for this type of workflow.

- When creating complex structures such as shell structures, where the shape and structural performance go hand in hand, parametric modelling is highly beneficial due to its flexible nature. The investigation of the shape requires careful consideration and the design should be continuously updated until the solution is satisfactory. Through the course of this thesis, benefits of the parametric design were found to be abundant. Both with regards to obtaining good results and optimising the structure, but also how small changes really do affect a shell structure.

- Conventional methods would be very inefficient to adopt for the design of shell structures. However, when the structure is more classic and the design process is well-known or straight forward, the traditional design methods may be as good or even better than parametric design.

7.1 Considerations regarding design proposals of the NTNUI cabin

Note that the design proposal is based on the **assumptions**:

1. Rigid joints.
2. Buckling length $L_k = 0.5 * L_{beam}$ (Based on assumption No.1. above).
3. Shear forces are small, hence not studied in detail. Only accounted for in ULS checks.
4. Displacements are considered the most crucial to minimise and therefore the shape is optimised accordingly.
5. The same UV-count of the grid (UV defined based on the length of the curves for the left rail (U) and the section(V)) can be adopted for *Shape 1* as for *Shape 0* due to the small modifications of the shape.
6. Approximate value of the cladding is set to $0.5kN/m^2$.
7. The *Drifted* load case is deduced from EC, but is not divided into the same zones, i.e. not triangular snow distribution etc.
8. Snow will always be present, thus suction of wind is not investigated as a case.
9. Live load not considered due to early stage of design process.

Based on these assumptions, the following needs to be taken into **consideration**.

1. The joints are designed by Helle and Øyvind in a different case study of the NTNUI cabin, where they are seeking to design the joints as rigid as possible. However 100% rigidity does not exist. As the grid pattern until now has been considered as squared, sufficient rigidity of the joints is crucial. We want to avoid planar movement of the grid due to lacking shear stiffness - squares as structural elements has the possibility of generating a mechanism. This invites the question of adding stiffeners to avert the mechanism, for instance diagonal stiffeners and creating a triangular pattern.
2. This is a non-conservative assumption which needs thorough consideration based on what is resolved with regards to consideration No.1 above.
3. Going forward this ought to be validated. Need to consider shear forces in the design of the joints.
4. Alternative approaches of optimising the shape should be considered. It was observed how the SLS requirements (displacements) were satisfied with a good margin when the cross section was designed according to ULS. It could therefore be thought that the more optimised shape could have been found with minimising the combination of normal forces and moment, to increase the buckling safety for instance, and in general reduce the most critical checks for utilisation.

5. This was found to not be correct as the length of the left rail and section curve for *Shape 1* would result in a higher UV-count and a denser structure. *Shape 1* was accordingly analysed with longer beam elements than *Shape 0* and thus more prone to local buckling. Had the UV- count been updated the structure would have needed a lower cross section. Consequently, the optimisation should been done with a changing UV-count.

6. This can easily be changed in the code, magnitude of load is controlled by a slider for the load of interest.

7. Should look into adopting wind tunnels and/or other experiments to find how the wind affects the cabin, and how the snow would realistically accumulate on the shape in question.

8. Need to consider wind actions without snow, as suction/lifting of the roof can be a potential problem.

9. Live load should be considered for the final design of the grid shell cabin. As point loads are especially dangerous for shell structures (local action), the live load should be considered as a point load equivalent to the weight of a person, approximately $1kN$.

In addition, the structural details of the cabin needs further study. For instance, the connection details of how the cladding is connected to the grid. Given a sufficient rigidity of the joints in the grid shell, the cladding should simply offer coverage, and not be a part of the structural framework. This implies that the panels/cladding should be connected to the grid directly to the nodes, to ensure all forces are being transferred to the nodes and through the beams to the ground (and not through the surface). Hence some relative movement of the plates should be allowed.

7.2 Further work

The exploration of shell shapes should be continued with parametric approaches of form-finding and procedures of optimising shapes. Throughout this thesis it was found how small adjustments to the shape can contribute to substantially increasing the structure's capacity. The shape optimisation in Chapter 6 is solely a suggestion and only the beginning of exploring the complexity and details rooted in the parametric form-optimisation procedure. Further research could be to continue adjusting the shape as the optimisation process should be an iterative process.

It could be interesting to perform a shape study with the inclusion of practical constraints. For instance by initiating the form finding procedure of the shell with a specific floor area, and implement various methods to see how the shape is optimised for different boundary conditions. This could be a parametric study of arbitrary freeform shapes or geometric shapes like the dome but with openings to make the shape more realistic to construct. Alternatively, the form could be found with numeric approaches such as the Force Density Method, to obtain a structure working primarily in membrane action.

The opportunities attached to the use of parametric software deserves to be exploited and investigated further. A variant of load design on shell structures was sought out in this thesis, and the load algorithm gave interesting results. However, its adoptability was questionable due to some contradictions with the guidelines in EC. Hence, an interesting topic would be the development of a parametric load algorithm for shell structures based on a combination of experimental data and/or EC. A code receiving the input of the shape of a shell, and automatically generating an appropriate/several load distribution(s) on the shape in question.

Standards

[35] Standard Norge. *Eurocode 1: Actions on structures, Part 1-3: General actions, Snow actions*. Brussels, July 2003. URL: `http://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=345414` (visited on 22/04/2018).

[36] Standard Norge. *Eurocode 1: Actions on structures, Part 1-4: General actions, Wind actions*. Brussels, Apr. 2005. URL: `http://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=436137` (visited on 22/04/2018).

[37] Standards Australia. *Structural design actions, Part 2: Wind Actions*. Australia, 2011. URL: `https://www.scribd.com/document/290190540/as-nzs-1170-2-2011-pdf` (visited on 15/03/2018).

[39] Standard Norge. *Eurocode*. Brussels, July 2003. URL: `http://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=345414` (visited on 22/04/2018).

[41] Standard Norge. *Timber structures, Glued laminated timber and glued solid timber, Requirements*. Brussels, June 2013. URL: `http://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=866055` (visited on 05/05/2018).

[46] Standard Norge. *Eurocode 5: Design of timber structures, Part 1-1: General Common rules and rules for buildings*. Brussels, Nov. 2004. URL: `http://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=437401` (visited on 02/05/2018).

Pictures

[4] G. Marcus. *Pantheon [Internet].* marcuslink.com. URL: https://marcuslink.com (visited on 20/02/2018).

[5] The History Hub. *Pantheon Historical Facts and Pictures [Internet].* The History Hub. URL: https://www.thehistoryhub.com/ (visited on 20/02/2018).

[9] S. Adriaenssens et al. *Shell structures for architecture: Form finding and optimization.* 2014.

[11] I. Liddell. *Frei Otto and the development of gridshells.* 2015.

[12] Wilkhahn. *Frei Otto. Thinking in Models – Wilkhahn recommends architecture exhibition at the ZKM Karlsruhe [Internet].* Wilkhahn Blog, 2017. URL: https://blog.wilkhahn.com/frei-otto-thinking-models-wilkhahn-recommends-architecture-exhibition-zkm-karlsruhe/ (visited on 20/03/2018).

[14] R. Harris, B. Gusinde and J. Roynon. *Design and construction of the Pods Sports Academy, Scunthorpe, England.* Auckland, New zealand, 2012.

[17] designboom. *Robotically fabricated Landesgartenschau Exhibition Hall in Germany [Internet].* designboom. URL: https://www.designboom.com/architecture/robotically-fabricated-landesgartenschau-exhibition-hall-06-25-2014/ (visited on 29/05/2018).

[19] A. Menges, T. Schwinn and Krieg O. *Landesgartenschau Exhibition Hall.* 2015.

[20] ArchDaily. *Landesgartenschau Exhibition Hall/ICD/ITKE/IIGS University of Stuttgart [Internet].* ArchDaily, 2014. URL: https://www.archdaily.com/520897/landesgartenschau-exhibition-hall-icd-itke-iigs-university-of-stuttgart (visited on 29/05/2018).

[22] scp. *Construction Principles: The clou with the Mesh [Internet].* schlaich bergermann partner, 2015. URL: https://www.sbp.de/en/themes/construction-principles/6 (visited on 30/05/2018).

[23] pxhere. *Spiderweb [Internet].* pxhere, 2017. URL: https://pxhere.com (visited on 30/05/2018).

[28] J. Heyman. *The stone skeleton: structural engineering of masonry.* Cambridge, 1995.

[34] hoydedata.no. *Laserinnsyn (topo4) [Internet].* Kartverket. URL: https://hoydedata.no/LaserInnsyn/ (visited on 15/03/2018).

[43] google.no/maps. *Location of site [63.5898110, 10.3258360] [Internet].* Kartdata Google. URL: https://www.google.no/maps (visited on 03/04/2018).

Other References

[1] S. Adriaenssens et al. *Shell structures for architecture: Form finding and optimization*. Routledge, 2014. ISBN: 9780415840606.

[2] E. Hjort-Hansen et al. *Snow Engineering 2000: Recent Advances and Development*. Balkema, A.A., 2000. ISBN: 9058091481.

[3] B. Addis. *3000 years of Design Engineering and Construction*. Phaidon Press Ltd, 2007. ISBN: 9780714869391.

[6] O.P. Larsen. *Conceptual Structural Design : Bridging the Gap Between Architects and Engineers*. London: Thomas Telford Publishing, Dec. 2003. ISBN: 9780727738899. URL: `http://site.ebrary.com/id/10868592`.

[7] D. Holzer, R. Hough and M. Burry. "Parametric Design and Structural Optimisation for Early Design Exploration". In: *International Journal of Architectural Computing* 5.1-4 (2007), pp. 625–44. ISSN: 1478-0771.

[8] C.R.B. Hernandez. "Thinking parametric design: introducing parametric Gaudi". In: *Design Studies* 27.3 (2006), pp. 309–24. ISSN: 0142-694X.

[10] I. Liddell. "Frei Otto and the development of gridshells". In: *Case Studies in Structural Engineering* 4 (2015), pp. 39–49. ISSN: 2214-3998.

[13] R. Harris, B. Gusinde and J. Roynon. "Design and construction of the Pods Sports Academy, Scunthorpe, England". In: Auckland, New zealand: World Conference on Timber Engineering 2012, 2012, pp. 510–517.

[15] Buro Happold Engineering. *The Pods Sports Academy Scunthoupe, UK. A complex design with sustainability at its heart [Internet]*. Buro Happold Engineering. 2012. URL: `https://www.burohappold.com/wp-content/uploads/2018/03/Scunthorpe-Sports-Academy-Yorkshire.pdf` (visited on 03/03/2018).

[16] A. Menges, T. Schwinn and O. Krieg. "Landesgartenschau Exhibition Hall". In: *Interlocking Digital and Material Cultures* (2015), pp. 56–71.

[18] J.M. Li and J. Knippers. "Segmental Timber Plate Shell for the Landesgartenschau Exhibition Hall in Schwabisch Gmund - the Application of Finger Joints in Plate Structures". In: *International Journal of Space Structures* 30.2 (2015), pp. 123–39. ISSN: 0266-3511.

[21] W. Flügge. *Stresses in Shells*. Second Edition. Berlin: Springer, 1973. ISBN: 3-642-88291-9.

[24] R. Harris. "Design of timber gridded shell structures". In: *Proceedings of the Institution of Civil Engineers* 164.2 (2011), pp. 105–116. ISSN: 0965-0911.

[25] J. Blaauwendraad and J.H Hoefakker. *Structural Shell Analysis: Understanding and Application*. Springer Netherlands, 2014. ISBN: 9789400767003.

[26] K. Bell. *An engineering approach to Finite Element Analysis of linear structural mechanics problems*. Trondheim: Akademika Publishing, 2013. ISBN: 9788232102686.

[27] J. Heyman. *The stone skeleton: structural engineering of masonry*. Cambridge: Cambridge University Press, 1995. ISBN: 0521629632.

[29] Mode Lab. *The Grasshopper Primer (EN)*. Mode Lab. 2015. URL: `http://grasshopperprimer.com/en/index.html?index.html` (visited on 18/05/2018).

[30] Karamba 3D. *Karamba Parametric Engineering [Internet]*. Karamba 3D. URL: `https://www.karamba3d.com/` (visited on 30/05/2018).

[31] C. Preisinger. *FEM-definition for the shell element of Karamba*. Karamba 3D. 2015. URL: `http://www.grasshopper3d.com/group/karamba/forum/topics/fem-definition-for-the-shell-element-of-karamba` (visited on 30/04/2018).

[32] C. Preisinger. *Beam element interpolation in Karamba*. Karamba 3D. 2016. URL: `http://www.grasshopper3d.com/group/karamba/forum/topics/beam-element-interpolation-in-karamba-1` (visited on 30/04/2018).

[33] D. Rutten. *Galapagos Evolutionary solver [Internet]*. Karamba 3D. 2010. URL: `http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles` (visited on 19/05/2018).

[38] P.A. Blackmore and E. Tsokri. “Wind loads on curved roofs”. In: *Journal of wind engineering and industrial aerodynamics* 94.11 (2006), pp. 833–844. ISSN: 0167-6105.

[40] R. & Associates McNeel. *What are NURBS? [Internet]*. Rhinoceros. 2018. URL: `https://www.rhino3d.com/nurbs` (visited on 05/03/2018).

[42] C. Preisinger. *Parametric Structural Modeling Karamba, User Manual for Version 1.2.2*. Karamba 3D. Oct. 2016. (Visited on 22/04/2018).

[44] E. Serano et al. *Limtreboka*. Norske Limtreprodusenters Forening, 2015. ISBN: 978-82-7129-324-6.

[45] Moelven. *Standard Limtre [Internett]*. Moelven Limtre AS. 4th May 2018. URL: `https://www.moelven.com/no/Produkter-og-tjenester/Limtre-og-Kerto/Standard-Limtre/`.

Appendices

A | Calculations of Actions according to Eurocode

Snow calculations according to NS-EN 1991-1-3

NA.4.1(901) gives the characteristic values on ground for Leksvik kommune The values in Eurocode accounts for heights lower or equal to the limit height, $H_g = 150m$. Thus, since $H = 361m > H_g = 150m$, the expression (A.1) should be determined [NA.4.1(1)]:

$$s_k = s_{k,0} + n\Delta s_k, \tag{A.1}$$

where

$s_{k,0}$ is the characteristic snow load on ground for the given municipal. $s_{k,0} = 4.0$ for Leksvik kommune.

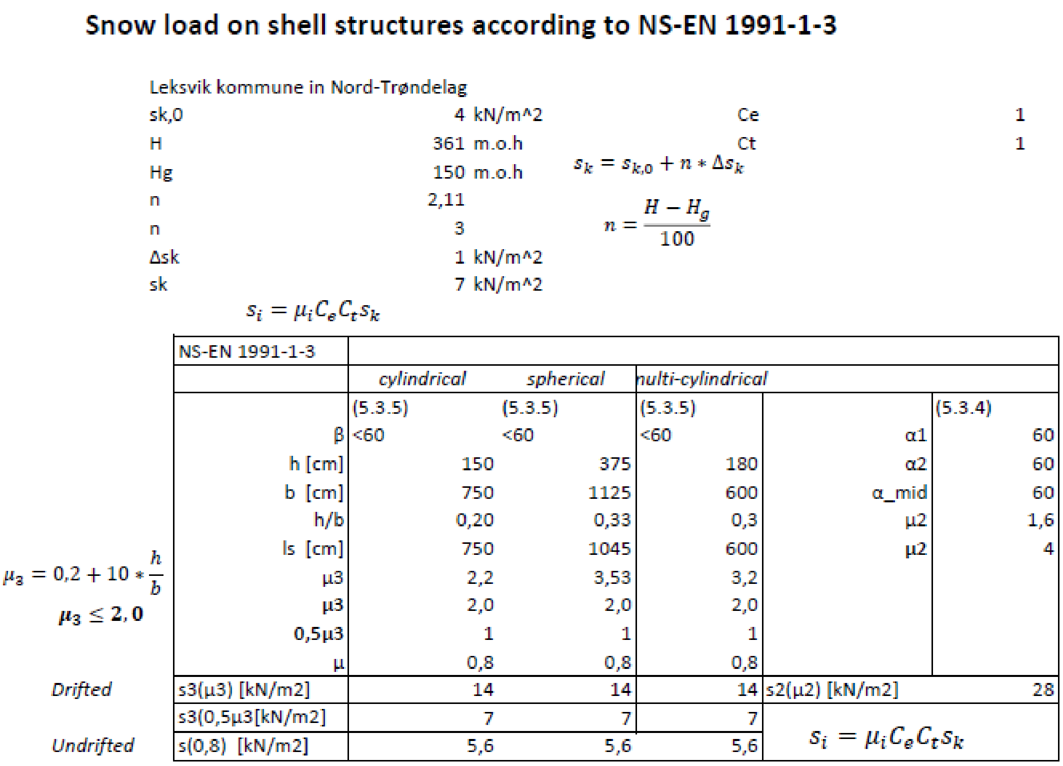
n = $(H - H_g)/100$, and n should be rounded up to the nearest integer, n=3.0.

Δs_k takes into account the increase in snow load for when $H > H_g$. $\Delta s_k = 1.0$

As a result, given the structure is situated at a height approximately $200meters$ above the reference height of Leksvik kommune, the characteristic value of snow load on ground increases:

$$s_k = 4.0 + 3 * 1.0 = 7.0kN/m^2.$$

The shape coefficients for the cylindrical, spherical and mulit-span cylindrical roof shapes, and the relevant values for the corresponding snow loads (according to Sec. 5.3 in NS-EN 1991-1-3) are presented in Figure A.1.



Snow load on shell structures according to NS-EN 1991-1-3

Leksvik kommune in Nord-Trøndelag

sk,0	4	kN/m^2	Ce	1
H	361	m.o.h	Ct	1
Hg	150	m.o.h		
n	2,11			
n	3			
Δsk	1	kN/m^2		
sk	7	kN/m^2		

$s_k = s_{k,0} + n * \Delta s_k$

$n = \frac{H - H_g}{100}$

$s_i = \mu_i C_e C_t s_k$

$\mu_3 = 0{,}2 + 10 * \frac{h}{b}$

$\boldsymbol{\mu_3 \leq 2{,}0}$

	NS-EN 1991-1-3					
		cylindrical	*spherical*	*multi-cylindrical*		
		(5.3.5)	(5.3.5)	(5.3.5)		(5.3.4)
	β	<60	<60	<60	α1	60
	h [cm]	150	375	180	α2	60
	b [cm]	750	1125	600	α_mid	60
	h/b	0,20	0,33	0,3	μ2	1,6
	ls [cm]	750	1045	600	μ2	4
	μ3	2,2	3,53	3,2		
	μ3	2,0	2,0	2,0		
	0,5μ3	1	1	1		
	μ	0,8	0,8	0,8		
Drifted	s3(μ3) [kN/m2]	14	14	14	s2(μ2) [kN/m2]	28
	s3(0,5μ3[kN/m2]	7	7	7	$s_i = \mu_i C_e C_t s_k$	
Undrifted	s(0,8) [kN/m2]	5,6	5,6	5,6		

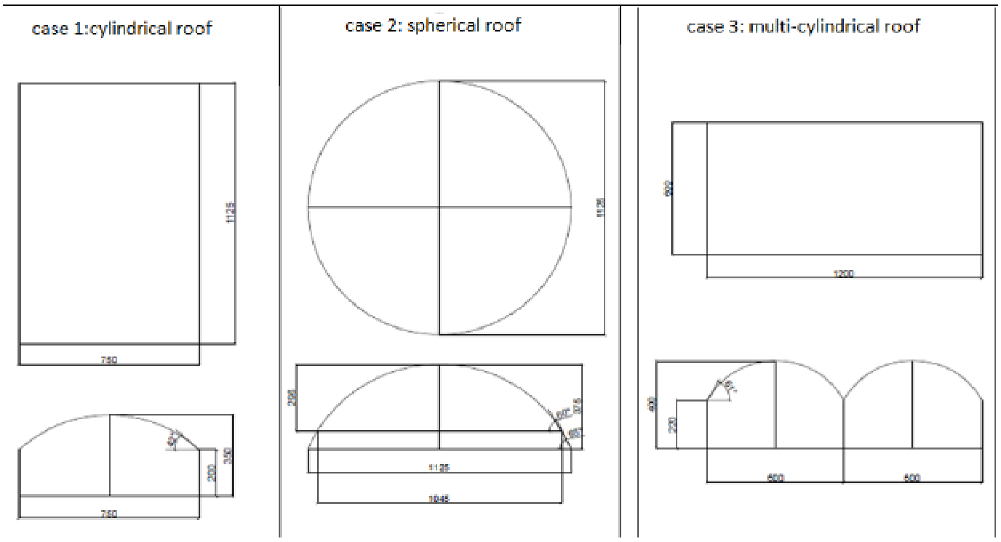


Figure A.1: Shapes investigated in Ch. 3 *Actions according to Eurocode* with shape factors and corresponding snow loads, *Drifted*: s(μ_i), i=2,3. and *Undrifted*: s(0.8).

Wind calculations according to NS-EN 1991-1-4

Calculation of wind pressure

The basic wind velocity is calculated from the following expression given in NA.4.2(2) expression (NA.4.1):

$$v_b = c_{dir} * c_{season} * c_{alt} * c_{prob} * v_{b,0} \tag{A.2}$$

c_{dir} The value of the directional factor for various wind directions may be found in Table NA.4(901.4). The recommended value is 1.0, which coincides with the value for wind coming from North-West direction in Region Trøndelag. Due to the surrounding topography, this is assumed to be the most recurring wind direction and is therefore considered in these calculations (see Fig. A.3).

c_{season} The value for the season factor is given in Table NA.4 (901.5) and is recommended to 1.0.

c_{alt} Above the tree line level, H_0, the basic wind velocity will increase [NA.4.2(2)P(901.1)]. In Table NA.4 (901.2) H_0 = 700 m for the district of the location evaluated. The structure is located 361 m.a.s.l, thus there will be no increase and the altitude factor, c_{alt}, is set to 1.0.

c_{prob} The probability factor is used if the return period differ from 50 years. Here 50 years is assumed, and c_{prob} is set to 1.0.

$v_{b,0}$ In the National Annex Table NA.4 (901.1) the fundamental basic wind $v_{b,0} = 26m/s$ is found for Leksvik Kommune.

As a result:

$$v_b = 1.0 * 1.0 * 1.0 * 1.0 * 26m/s = 26m/s$$

The peak velocity pressure given in expression NA. 4.8 (see Ex. (A.3)), which includes mean and short-term fluctuations, should then be determined:

$$q_p(z) = 0,5 * \rho * v_m(z)^2 * [1 + 2 * k_p * I_v(z)], \tag{A.3}$$

where

ρ is the air density with recommended value of $1.25m/s^3$

$v_m(z)$ is the mean velocity defined later in ex. (A.4).

$I_v(z)$ is the turbulence intensity defined later in ex. (A.6).

k_p a peak factor equal to 3.5

The mean velocity $v_m(z)$ is determined with expression (4.3) in Eurocode:

$$v_m(z) = c_r(z) * c_o(z) * v_b \tag{A.4}$$

The roughness factor, $c_r(z)$ accounts for variations in the mean wind velocity due to height above sea level and ground roughness of the terrain upwind the structure in the wind direction considered [4.3.2(1)].

To determine this factor Table NA.4.1 is used to define the terrain category and the values z_0, z_{min} and k_r. The terrain category is assumed to be 2, which is defined as: *an area with low vegetation such as grass and isolated obstacles with separations of at least 20 obstacle heights*. The values $z_0 = 0,05m$, $z_{min} = 4m$ and $k_r = 0,19$ are given in the table.

Expression (4.4) in [36] (see ex. (A.5)) is then used to determine the roughness factor $c_r(z)$, where z is the height above terrain. The structures is assumed to have a height of approximately 4 m. When $z_{min} \leq z \leq z_{max}$, where z_{max} is to be taken as 200 m, the following equation should be used:

$$c_r(z) = k_r * ln(z/z_0) = 0,19 * ln(4/0,05) = 0,83 \tag{A.5}$$

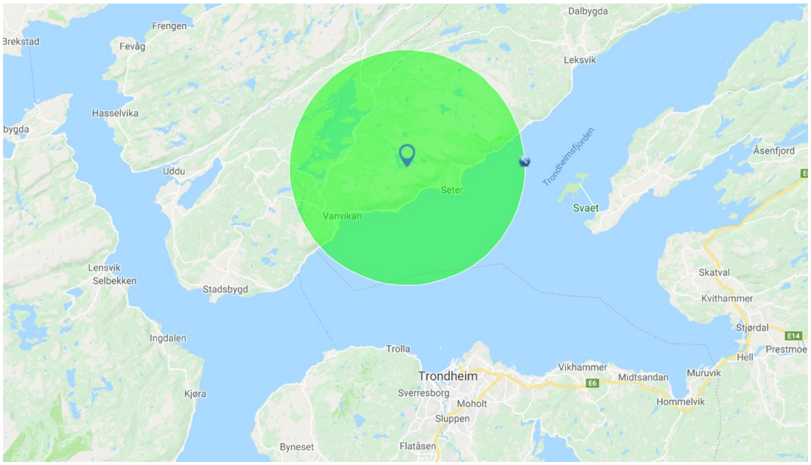


Figure A.2: 10 km radius circle drawn around the cabin location

In Figure A.2 a circle with 10 km radius is drawn around the location. NA.4.3.2(2) states that if the construction area is closer than 10 km from areas with deviating roughness, both v_m and I_v should be adjusted. As it appears from Figure A.2, the terrain will vary and have a lower category level around the sea and bigger lakes. However, here the wind direction considered is from north- west (see Fig. A.3), and it is assumed that the upstream distance with uniform terrain roughness is long enough to stabilize the profile sufficiently.

The mean wind velocity $v_m(z)$ (see eq. (A.4)), also depends on the topography. The topography factor $c_0(z)$ takes into account an increase in wind velocity due to e.g. hills and cliffs. The structure is not located near any slope tops, and it is assumed that the surrounding terrain resembles a valley (see Fig. A.3). Annex A, Section A.3(4) in [36] states that in a valley, $c_0(z)$ may be set to 1.0. The resulting mean wind velocity, defined in (A.4), is then:

$$v_m(z) = c_r(z) * c_0(z) * v_b = 0.83 * 1.0 * 26m/s = 21.65m/s$$

Then in order to calculate the peak velocity pressure (see eq. (A.3)), the wind turbulence intensity $I_v(z)$ need to be determined. Section 4.4 defines the turbulence intensity as:

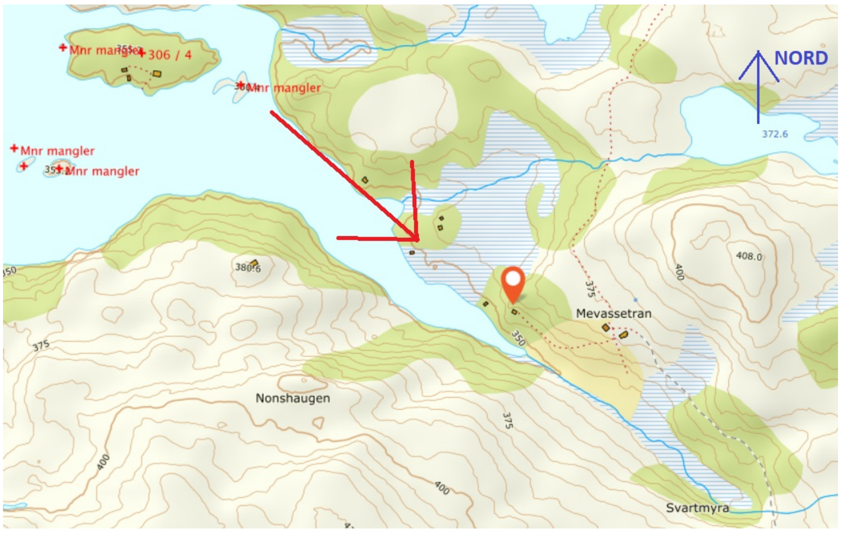


Figure A.3: Wind direction considered indicated with red arrow

$$I_v(z) = \sigma_v / v_m(z) = \frac{k_l}{c_o(z) * ln(\frac{z}{z_0})} \tag{A.6}$$

where

$$\sigma_v = k_r * v_b * k_l \tag{A.7}$$

According to Note 2 in Section 4.4(1) the turbulence factor k_l has the recommended value 1.0, and k_r is previously defined in Table NA.4.1. Hence,

$$\sigma_v = k_r * v_b * k_l = 0.19 * 1 * 26 = 4.94$$

and as a result:

$$I_v(z) = \frac{\sigma_v}{v_m(z)} = \frac{4.94}{21.65} = 0.228$$

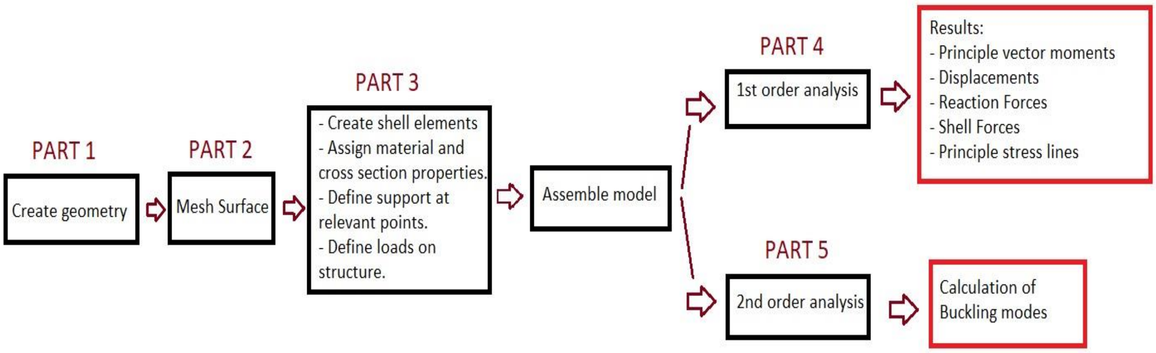
Now the peak velocity pressure for z=4 (see eq. (A.3)), can be determined.

$$q_p(z) = 0,5 * \rho * v_m(z)^2 * [1 + 2 * k_p * I_v(z)]$$
$$q_p(z) = 0.5 * 1.25 * 21.65^2 * [1 + 2 * 3,5 * 0,228]$$
$$q_p(z) = 694 N/m^2$$

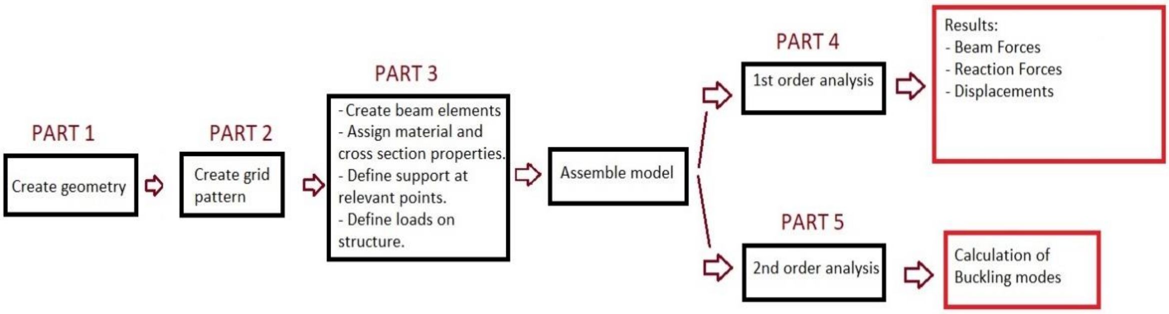
B | Grasshopper code for Parametric Study

Grasshopper codes for:

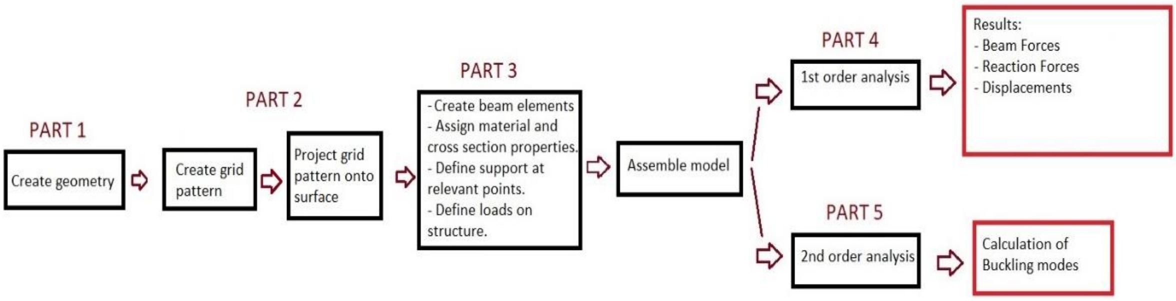
1. Concrete model



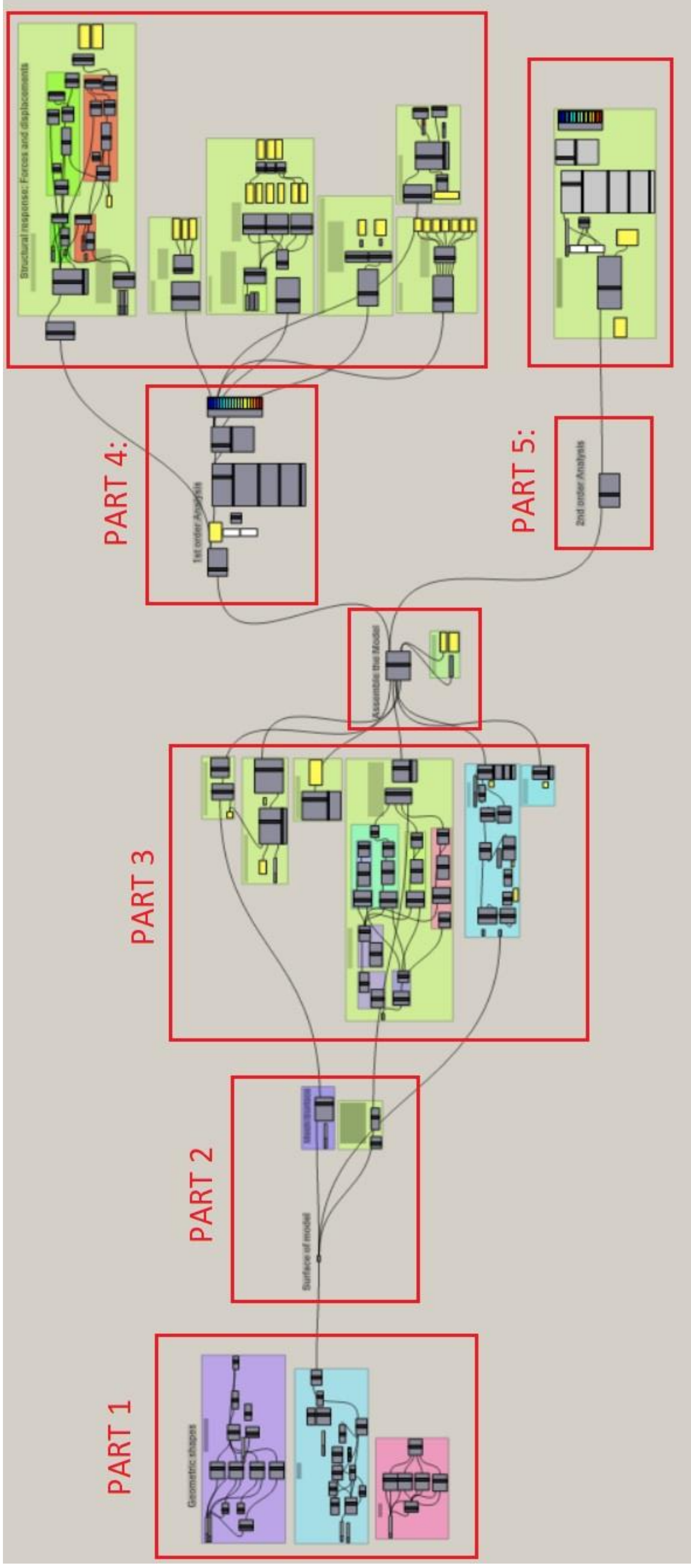
2. Grid shell Barrel Vault/ Plate



3. Grid shell Dome

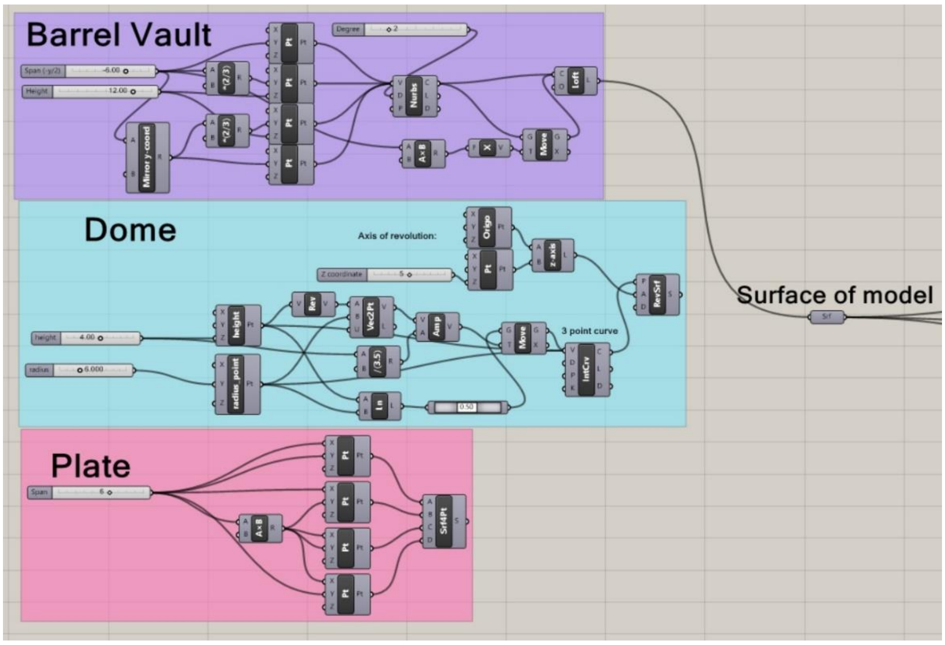


1. Concrete model



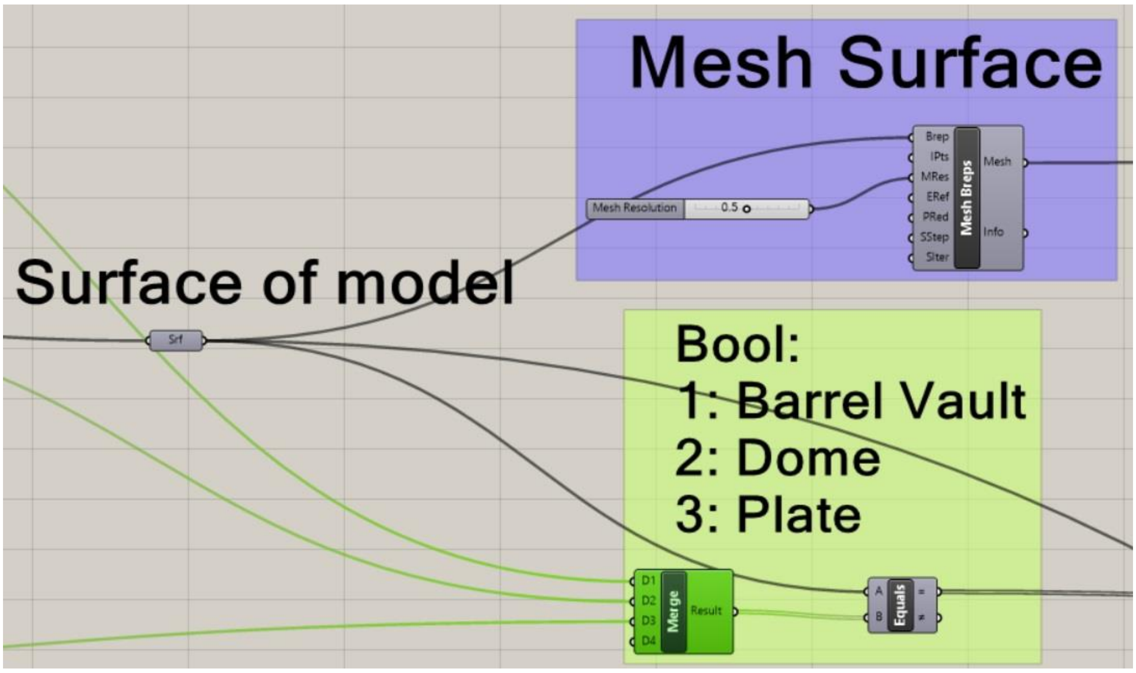
PART 1: Concrete model

***Create geometry**



PART 2: Concrete model

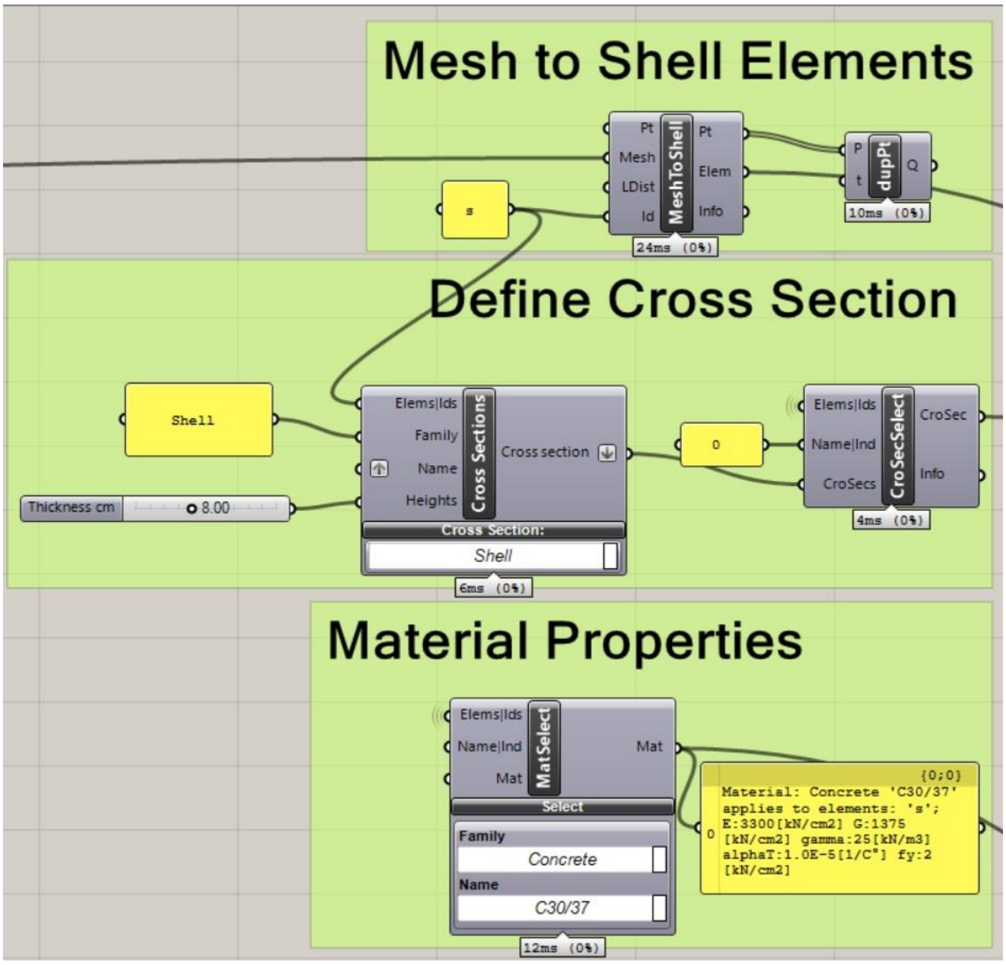
***Mesh Surface**



PART 3: Concrete model

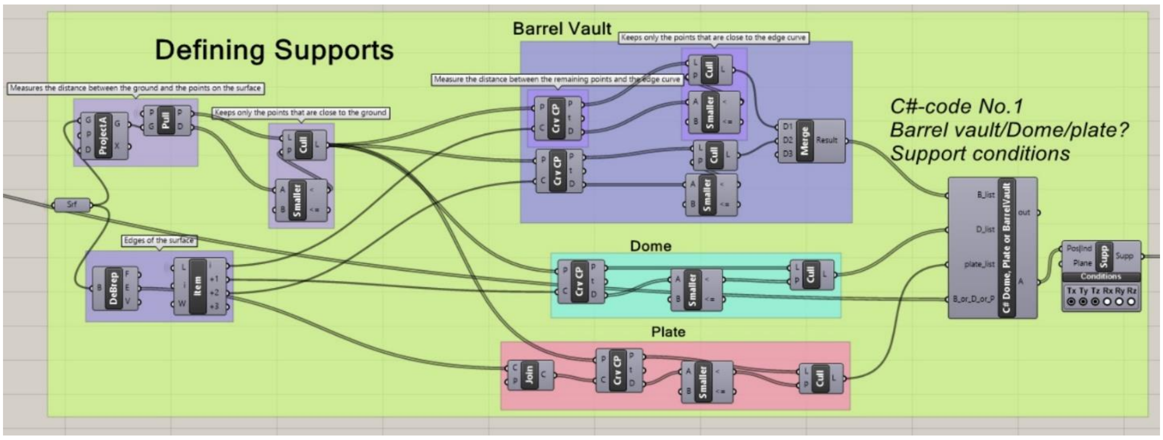
***Create shell elements**

***Assign material and cross section preperties**



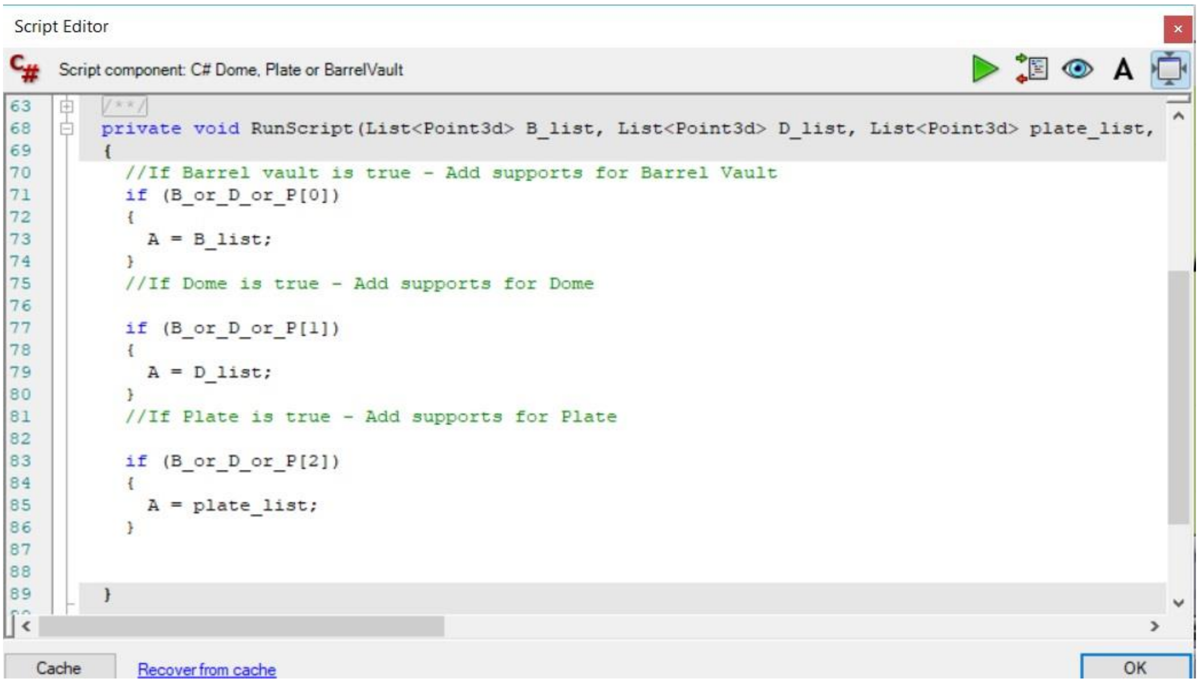
```
Material: Concrete 'C30/37'
applies to elements: 's';
E:3300[kN/cm2] G:1375
[kN/cm2] gamma:25[kN/m3]
alphaT:1.0E-5[1/C°] fy:2
[kN/cm2]
```

***Define supports at relevant points**


Defining Supports
Measures the distance between the ground and the points on the surface
ProjectA
Pull
Keeps only the points that are close to the ground
Cull
Smaller
Srf
Edges of the surface
DeBrep
Item
Barrel Vault
Keeps only the points that are close to the edge curve
Measure the distance between the remaining points and the edge curve
Crv CP
Cull
Smaller
Merge
Result
Dome
Plate
Join
C#-code No.1
Barrel vault/Dome/plate?
Support conditions
B_list
D_list
plate_list
B_or_D_or_P
C# Dome, Plate or BarrelVault
out
A
PosInd
Plane
Supp
Conditions
Tx Ty Tz Rx Ry Rz

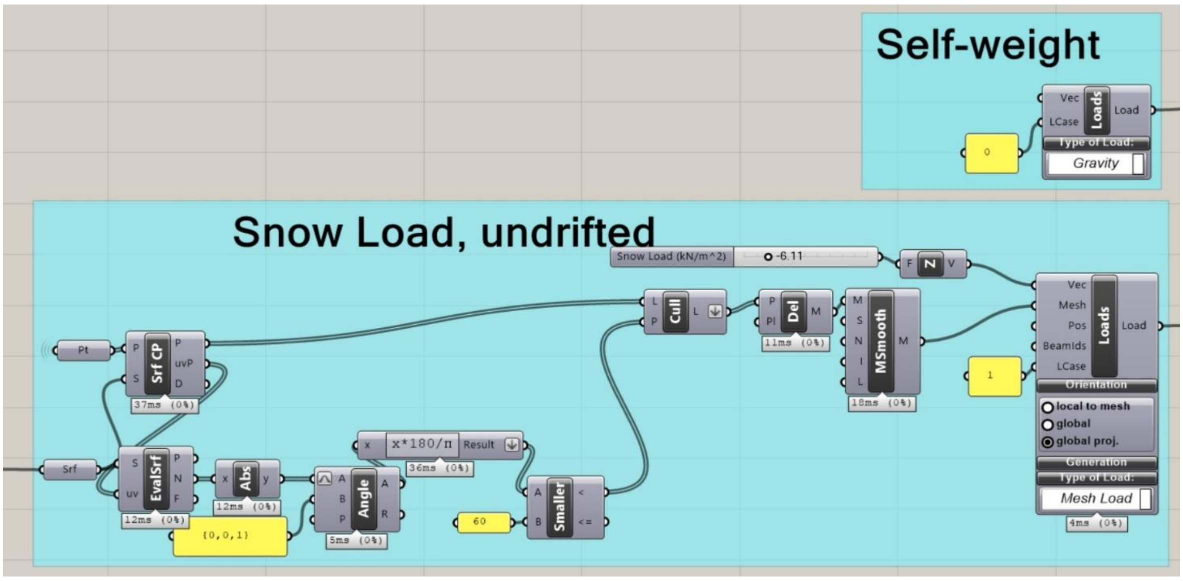
C#-code No.1 Concrete

Chose between Dome/Barrel Vault/Plate - Support conditions



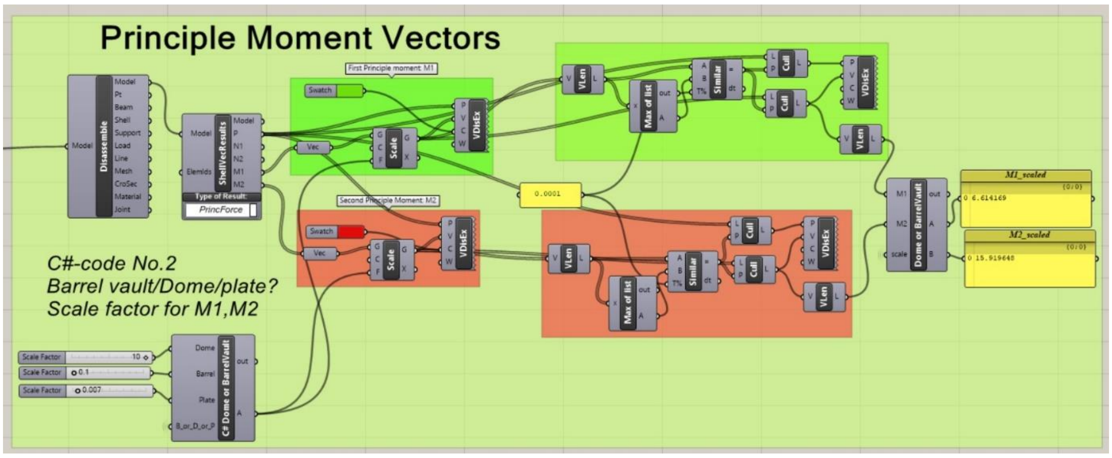
```
/**/
private void RunScript(List<Point3d> B_list, List<Point3d> D_list, List<Point3d> plate_list,
{
  //If Barrel vault is true - Add supports for Barrel Vault
  if (B_or_D_or_P[0])
  {
    A = B_list;
  }
  //If Dome is true - Add supports for Dome

  if (B_or_D_or_P[1])
  {
    A = D_list;
  }
  //If Plate is true - Add supports for Plate

  if (B_or_D_or_P[2])
  {
    A = plate_list;
  }


}
```

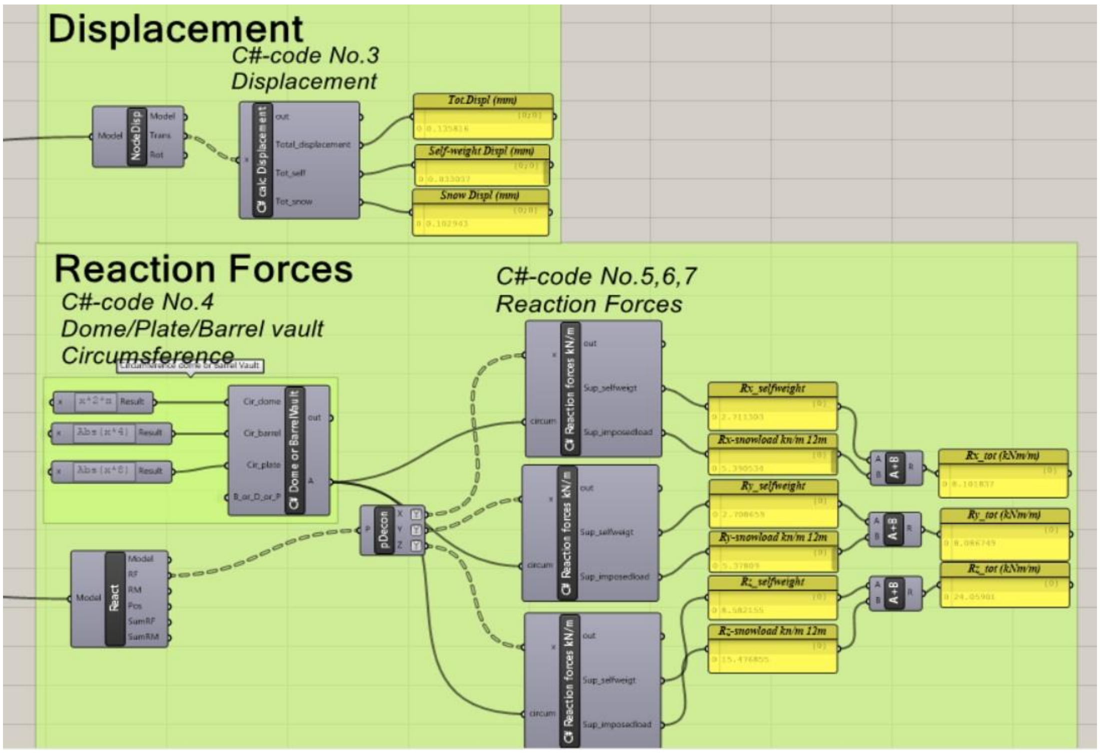
***Define loads on structure**



PART 4: Concrete model

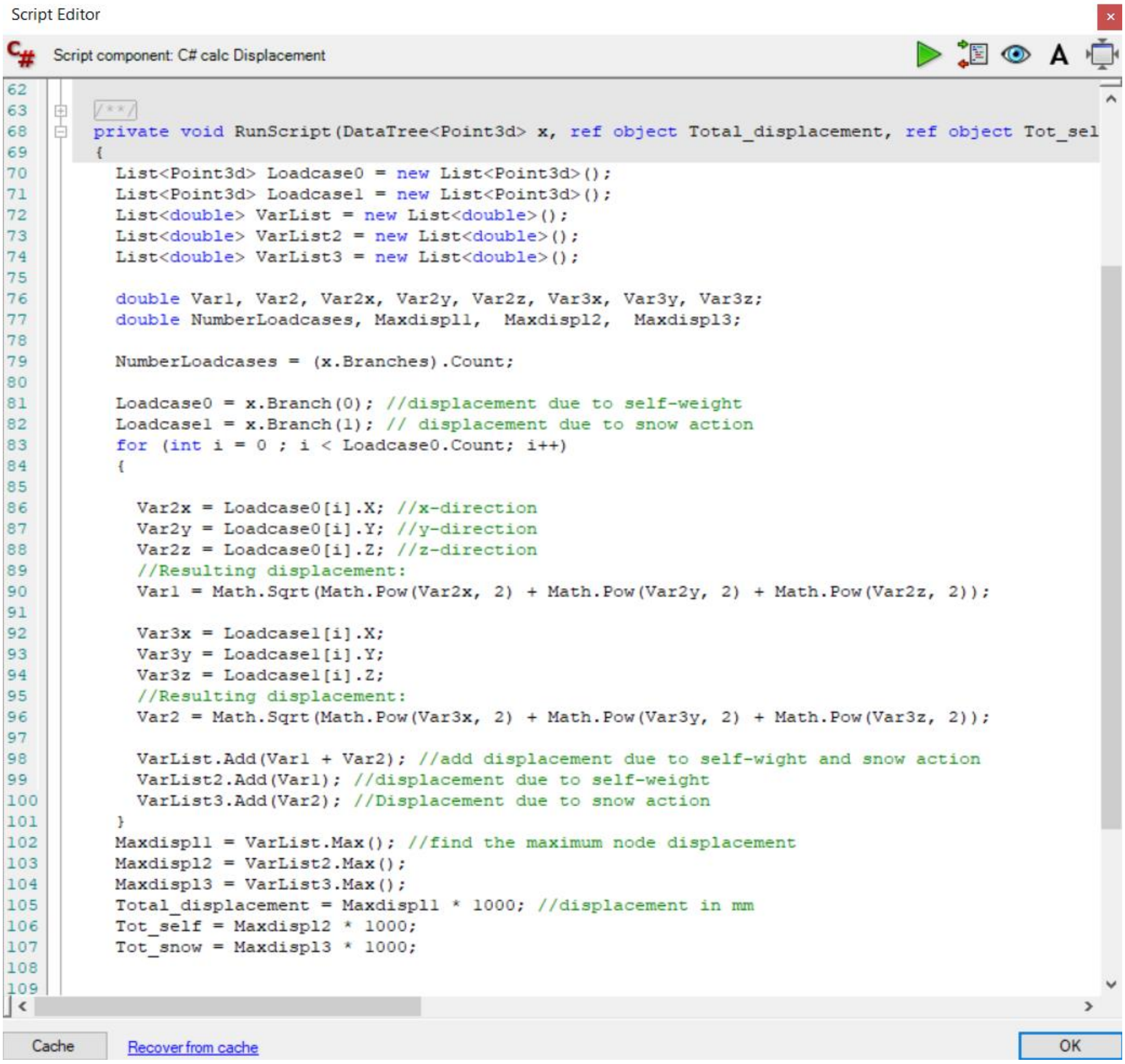
Results:
*** Principle Moment Vectors**
*** Displacement**
*** Reaction Forces**
*** Shell Forces**
*** Principle Stress lines**





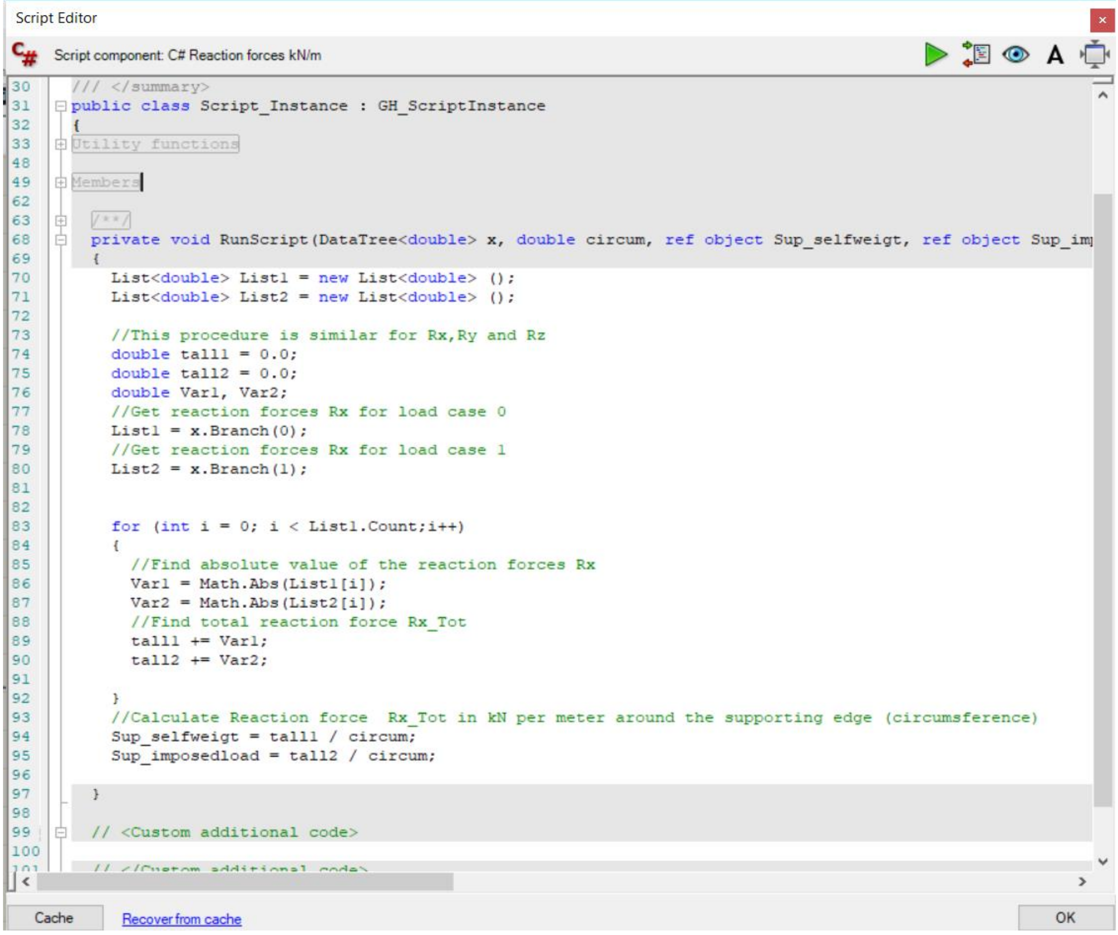
C# code No. 2 and No. 4 are similar to C# code No. 1.
The purpose of the codes are to choose a value based on if it is a barrel vault, dome or plate

C#-code No.3 Concrete
Find displacement due to self-weigth, snow and total displ.

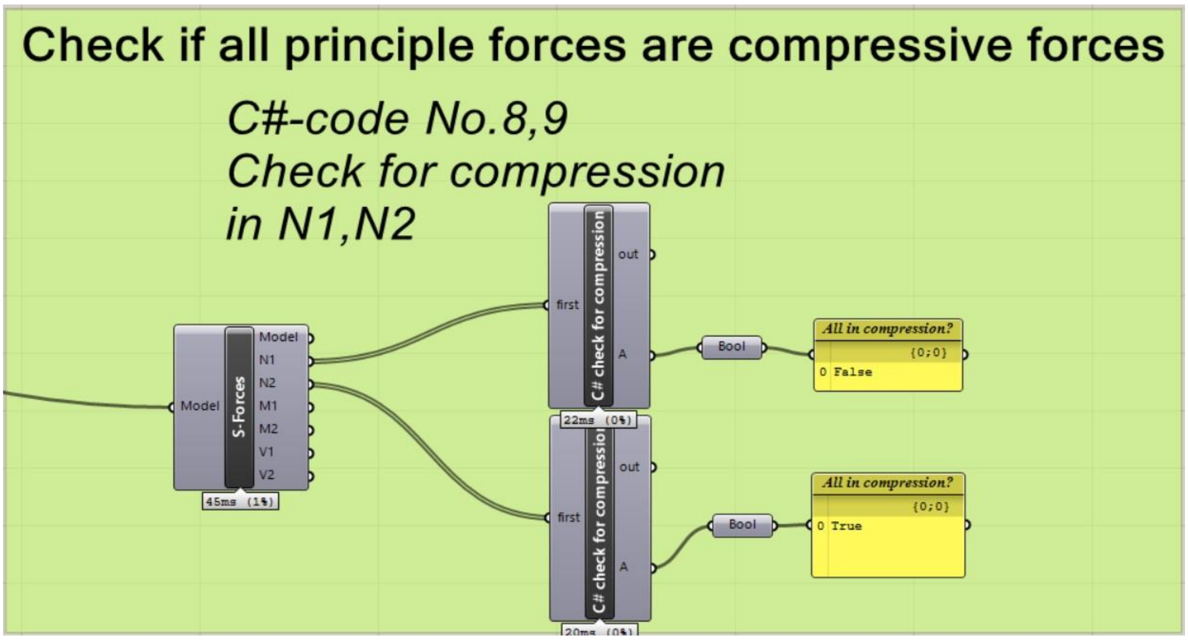


```
/**/
private void RunScript(DataTree<Point3d> x, ref object Total_displacement, ref object Tot_sel
{
  List<Point3d> Loadcase0 = new List<Point3d>();
  List<Point3d> Loadcase1 = new List<Point3d>();
  List<double> VarList = new List<double>();
  List<double> VarList2 = new List<double>();
  List<double> VarList3 = new List<double>();

  double Var1, Var2, Var2x, Var2y, Var2z, Var3x, Var3y, Var3z;
  double NumberLoadcases, Maxdispl1,  Maxdispl2,  Maxdispl3;

  NumberLoadcases = (x.Branches).Count;

  Loadcase0 = x.Branch(0); //displacement due to self-weight
  Loadcase1 = x.Branch(1); // displacement due to snow action
  for (int i = 0 ; i < Loadcase0.Count; i++)
  {

    Var2x = Loadcase0[i].X; //x-direction
    Var2y = Loadcase0[i].Y; //y-direction
    Var2z = Loadcase0[i].Z; //z-direction
    //Resulting displacement:
    Var1 = Math.Sqrt(Math.Pow(Var2x, 2) + Math.Pow(Var2y, 2) + Math.Pow(Var2z, 2));

    Var3x = Loadcase1[i].X;
    Var3y = Loadcase1[i].Y;
    Var3z = Loadcase1[i].Z;
    //Resulting displacement:
    Var2 = Math.Sqrt(Math.Pow(Var3x, 2) + Math.Pow(Var3y, 2) + Math.Pow(Var3z, 2));

    VarList.Add(Var1 + Var2); //add displacement due to self-wight and snow action
    VarList2.Add(Var1); //displacement due to self-weight
    VarList3.Add(Var2); //Displacement due to snow action
  }
  Maxdispl1 = VarList.Max(); //find the maximum node displacement
  Maxdispl2 = VarList2.Max();
  Maxdispl3 = VarList3.Max();
  Total_displacement = Maxdispl1 * 1000; //displacement in mm
  Tot_self = Maxdispl2 * 1000;
  Tot_snow = Maxdispl3 * 1000;

```

C#-code No.5,6,7 Concrete
Reaction forces Rx,Ry,Rz



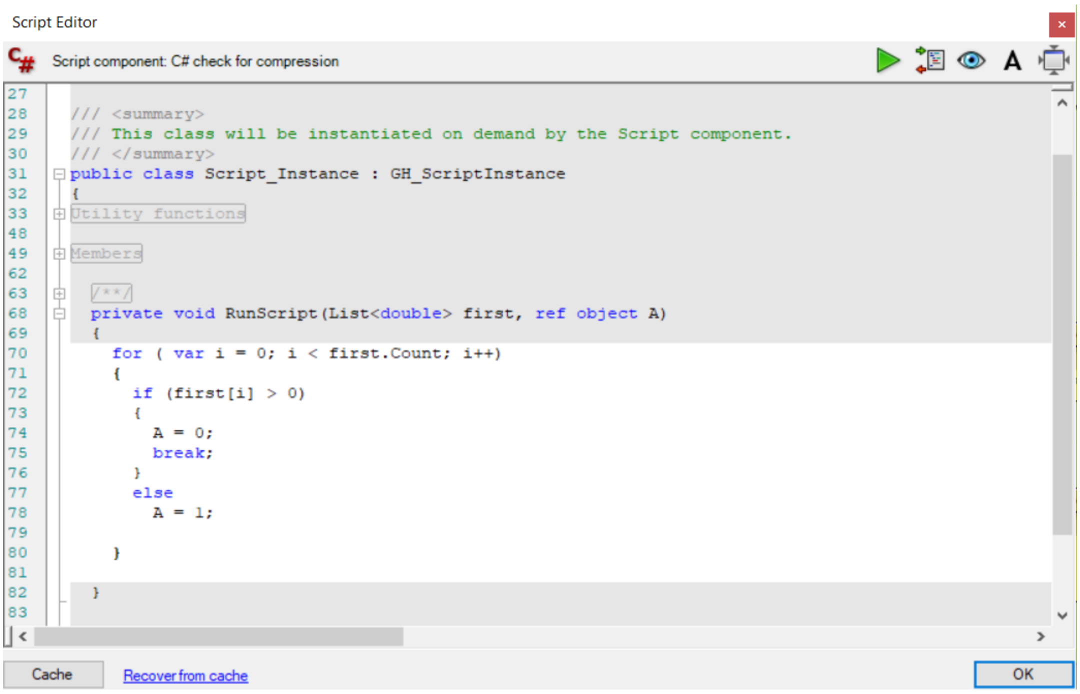
```
  /// </summary>
public class Script_Instance : GH_ScriptInstance
{
Utility functions

Members

  /**/
  private void RunScript(DataTree<double> x, double circum, ref object Sup_selfweigt, ref object Sup_im
  {
    List<double> List1 = new List<double> ();
    List<double> List2 = new List<double> ();

    //This procedure is similar for Rx,Ry and Rz
    double tall1 = 0.0;
    double tall2 = 0.0;
    double Var1, Var2;
    //Get reaction forces Rx for load case 0
    List1 = x.Branch(0);
    //Get reaction forces Rx for load case 1
    List2 = x.Branch(1);


    for (int i = 0; i < List1.Count;i++)
    {
      //Find absolute value of the reaction forces Rx
      Var1 = Math.Abs(List1[i]);
      Var2 = Math.Abs(List2[i]);
      //Find total reaction force Rx_Tot
      tall1 += Var1;
      tall2 += Var2;

    }
    //Calculate Reaction force  Rx_Tot in kN per meter around the supporting edge (circumsference)
    Sup_selfweigt = tall1 / circum;
    Sup_imposedload = tall2 / circum;

  }

  // <Custom additional code>

  // </Custom additional code>
```



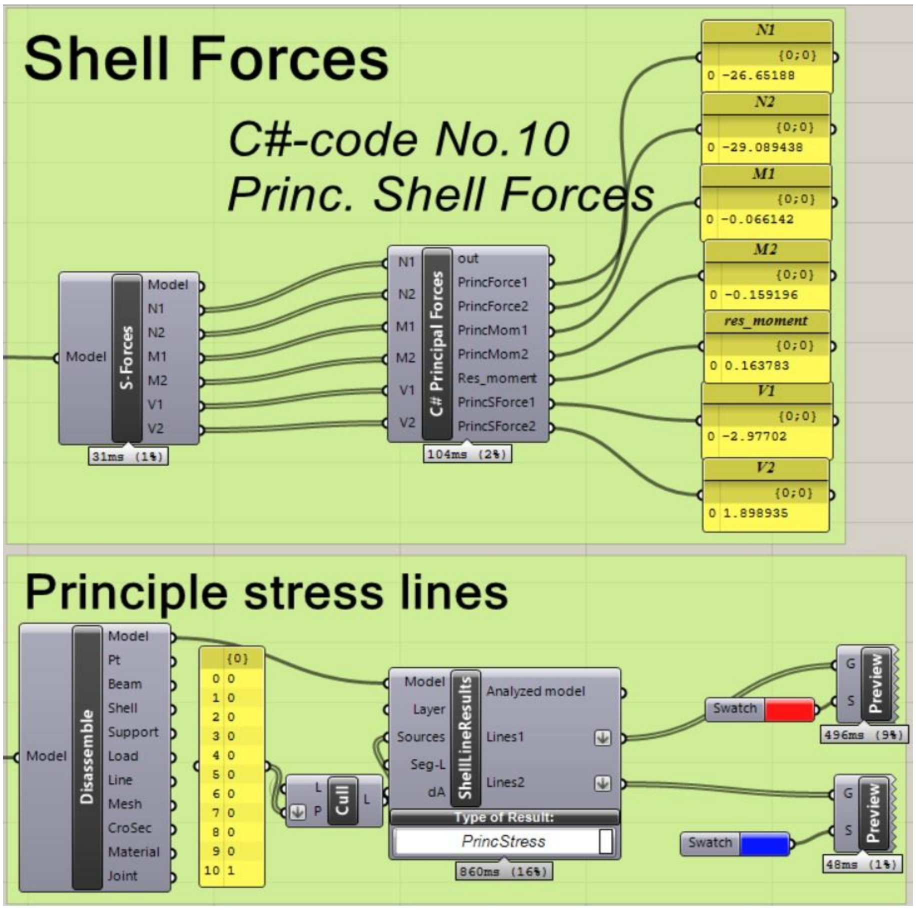
Check if all principle forces are compressive forces

C#-code No.8,9
Check for compression
in N1,N2

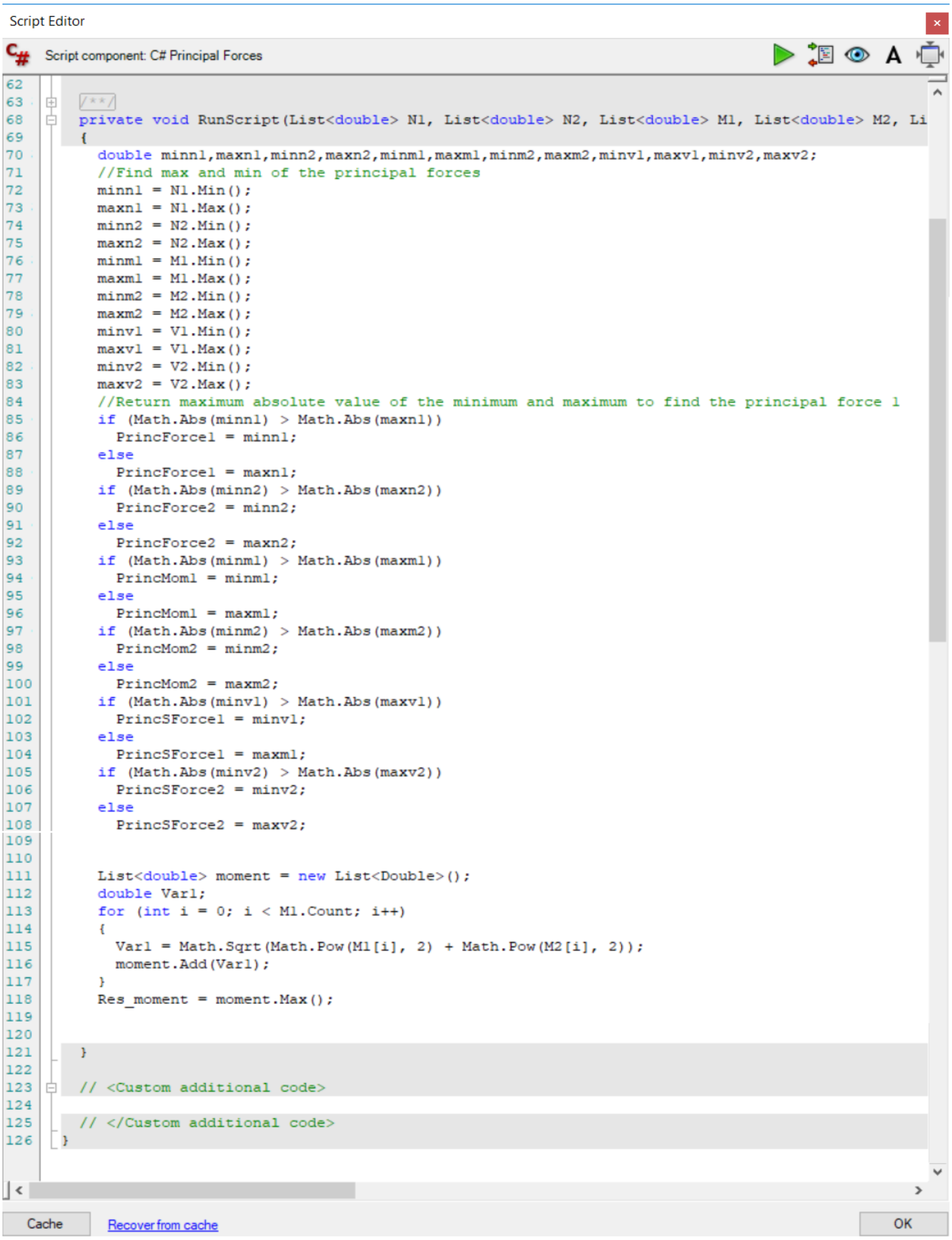
C#-code No.8,9 Concrete
Check if all forces are in compression



```
/// <summary>
/// This class will be instantiated on demand by the Script component.
/// </summary>
public class Script_Instance : GH_ScriptInstance
{
Utility functions

Members

  /**/
  private void RunScript(List<double> first, ref object A)
  {
    for ( var i = 0; i < first.Count; i++)
    {
      if (first[i] > 0)
      {
        A = 0;
        break;
      }
      else
        A = 1;

    }

  }
```



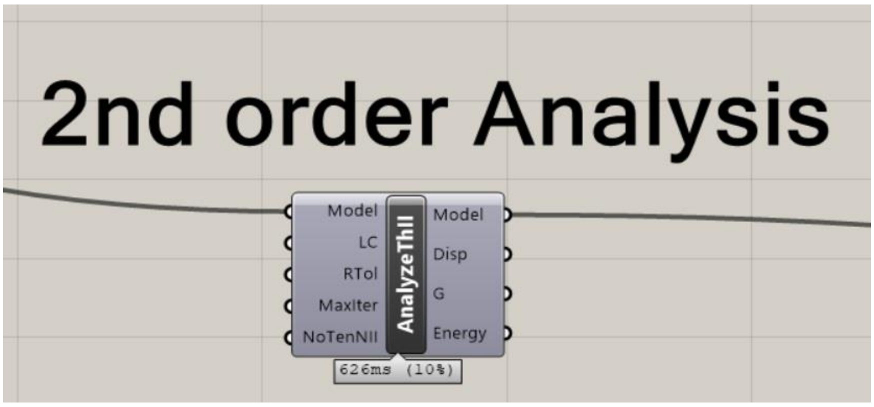
C#-code No.10 Concrete
Principal shell forces and resulting moment



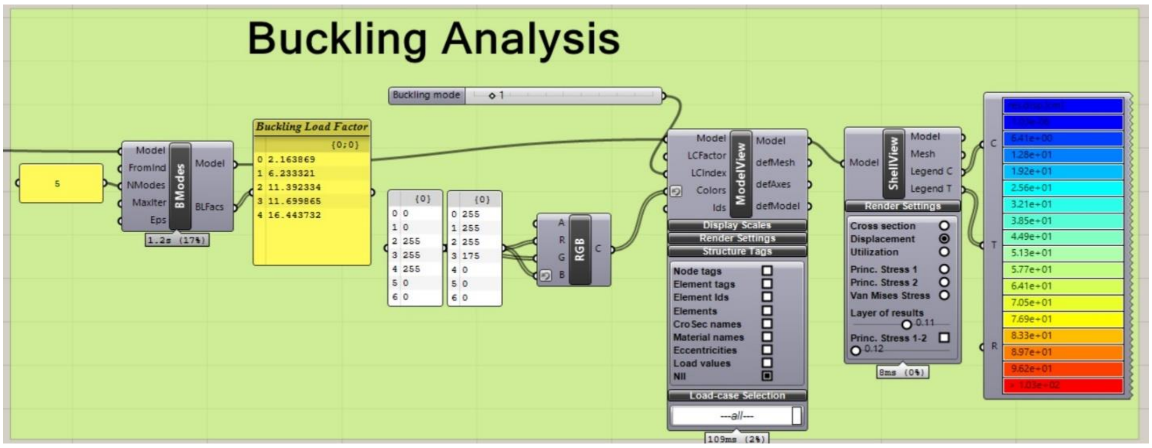
```
/**/
private void RunScript(List<double> N1, List<double> N2, List<double> M1, List<double> M2, Li
{
  double minn1,maxn1,minn2,maxn2,minm1,maxm1,minm2,maxm2,minv1,maxv1,minv2,maxv2;
  //Find max and min of the principal forces
  minn1 = N1.Min();
  maxn1 = N1.Max();
  minn2 = N2.Min();
  maxn2 = N2.Max();
  minm1 = M1.Min();
  maxm1 = M1.Max();
  minm2 = M2.Min();
  maxm2 = M2.Max();
  minv1 = V1.Min();
  maxv1 = V1.Max();
  minv2 = V2.Min();
  maxv2 = V2.Max();
  //Return maximum absolute value of the minimum and maximum to find the principal force 1
  if (Math.Abs(minn1) > Math.Abs(maxn1))
    PrincForce1 = minn1;
  else
    PrincForce1 = maxn1;
  if (Math.Abs(minn2) > Math.Abs(maxn2))
    PrincForce2 = minn2;
  else
    PrincForce2 = maxn2;
  if (Math.Abs(minm1) > Math.Abs(maxm1))
    PrincMom1 = minm1;
  else
    PrincMom1 = maxm1;
  if (Math.Abs(minm2) > Math.Abs(maxm2))
    PrincMom2 = minm2;
  else
    PrincMom2 = maxm2;
  if (Math.Abs(minv1) > Math.Abs(maxv1))
    PrincSForce1 = minv1;
  else
    PrincSForce1 = maxm1;
  if (Math.Abs(minv2) > Math.Abs(maxv2))
    PrincSForce2 = minv2;
  else
    PrincSForce2 = maxv2;


  List<double> moment = new List<Double>();
  double Var1;
  for (int i = 0; i < M1.Count; i++)
  {
    Var1 = Math.Sqrt(Math.Pow(M1[i], 2) + Math.Pow(M2[i], 2));
    moment.Add(Var1);
  }
  Res_moment = moment.Max();


}

// <Custom additional code>

// </Custom additional code>
}
```

PART 5: Concrete model

*2nd order analysis



*Calculation of Buckling Modes



Buckling Load Factor

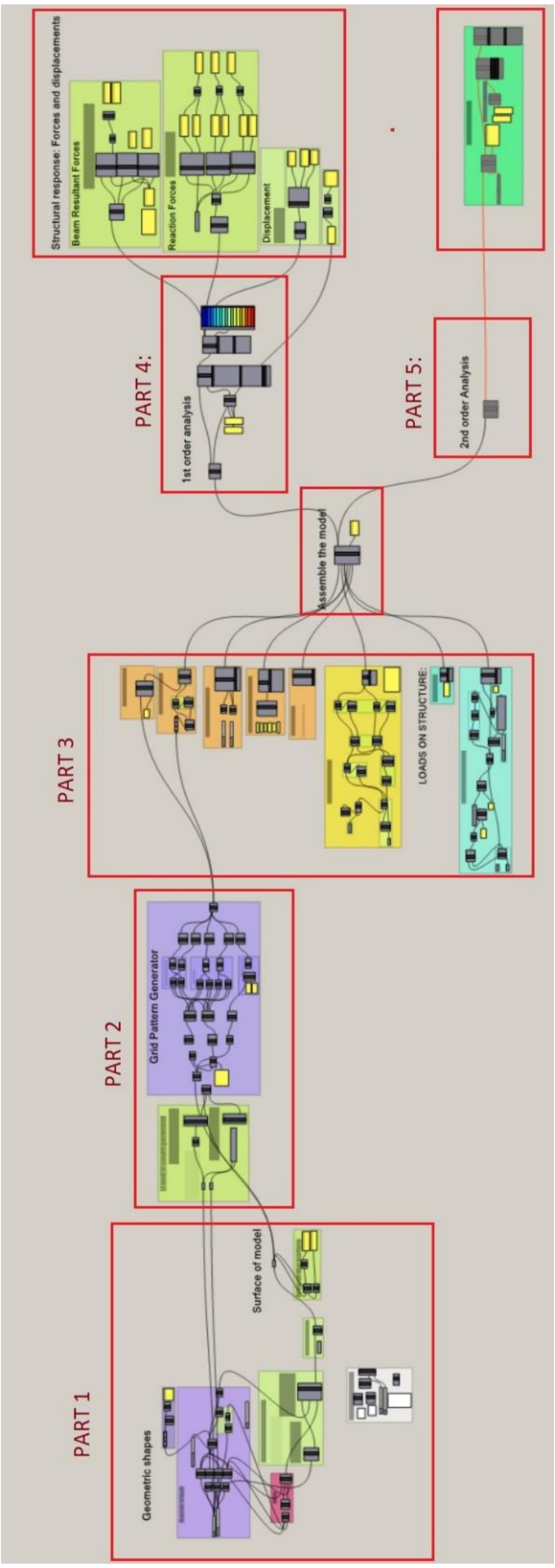
```
        {0;0}
0 2.163869
1 6.233321
2 11.392334
3 11.699865
4 16.443732
```

```
 {0}      {0}
0 0      0 255
1 0      1 255
2 255    2 255
3 255    3 175
4 255    4 0
5 0      5 0
6 0      6 0
```

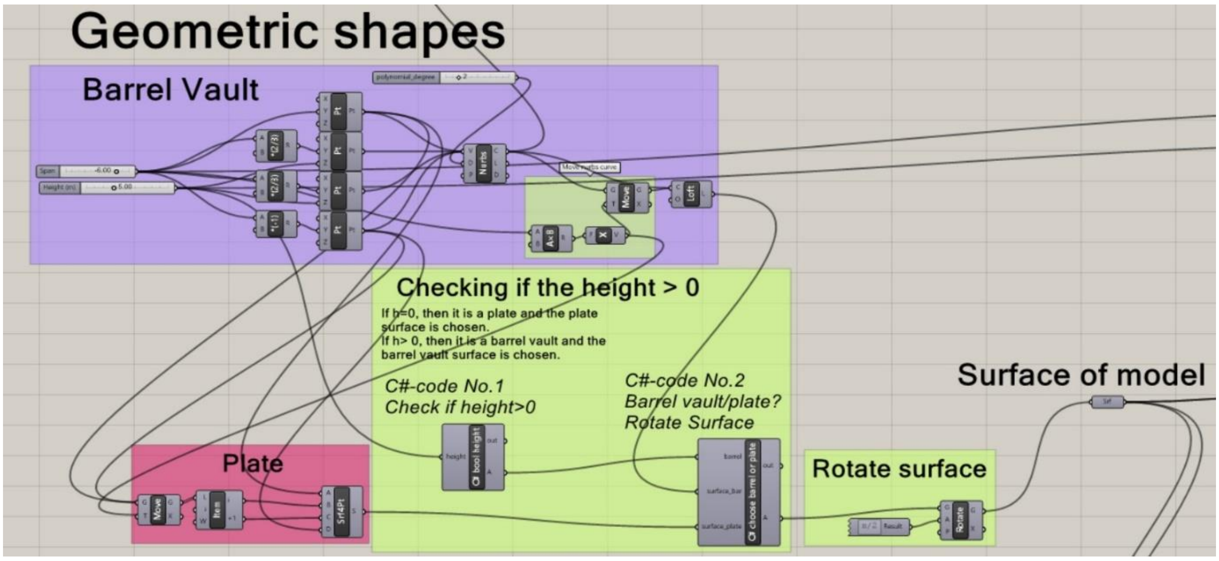
Node tags □
Element tags □
Element Ids □
Elements □
CroSec names □
Material names □
Eccentricities □
Load values □
NII ☒

Cross section ○
Displacement ●
Utilization ○
Princ. Stress 1 ○
Princ. Stress 2 ○
Van Mises Stress ○
Layer of results 0.11
Princ. Stress 1-2 □ 0.12

2. Grid shell model Barrel Vault/Plate (BV/P)

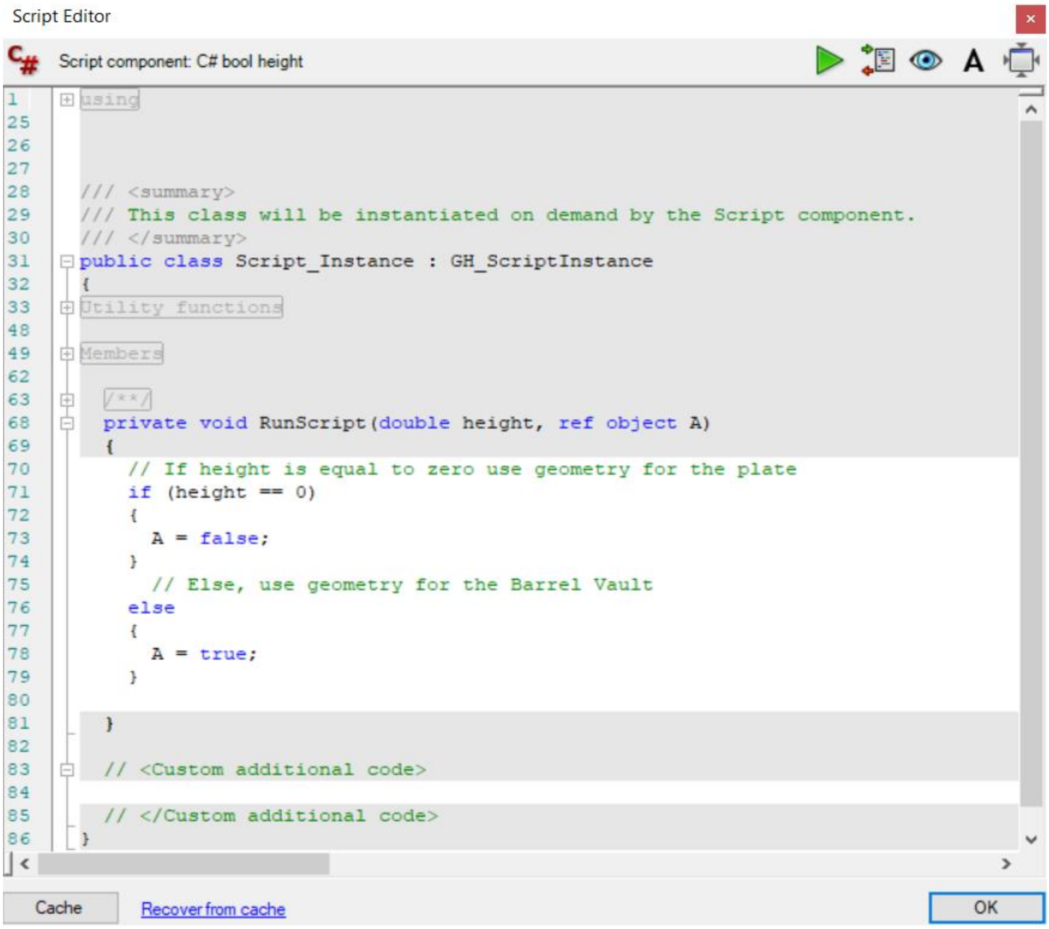


PART 1: Grid shell barrel vault/plate



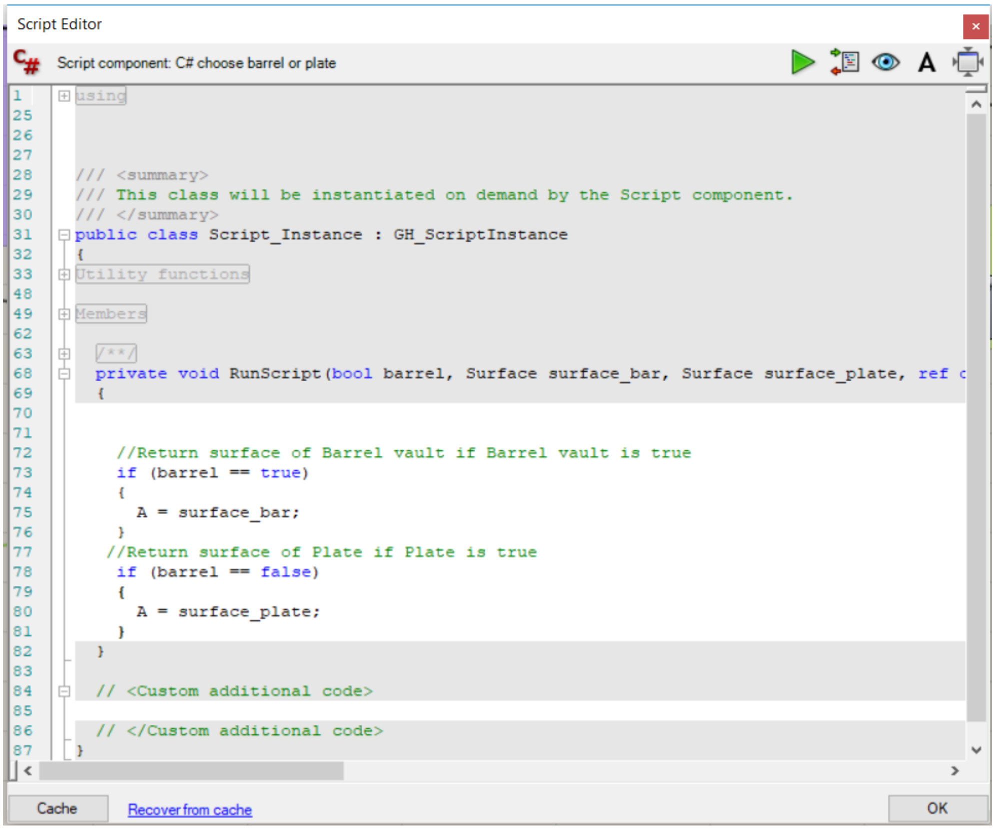
If h=0, then it is a plate and the plate surface is chosen.
If h> 0, then it is a barrel vault and the barrel vault surface is chosen.

C#-code No.1 Grid shell BV/P
Check if height equals 0 or not



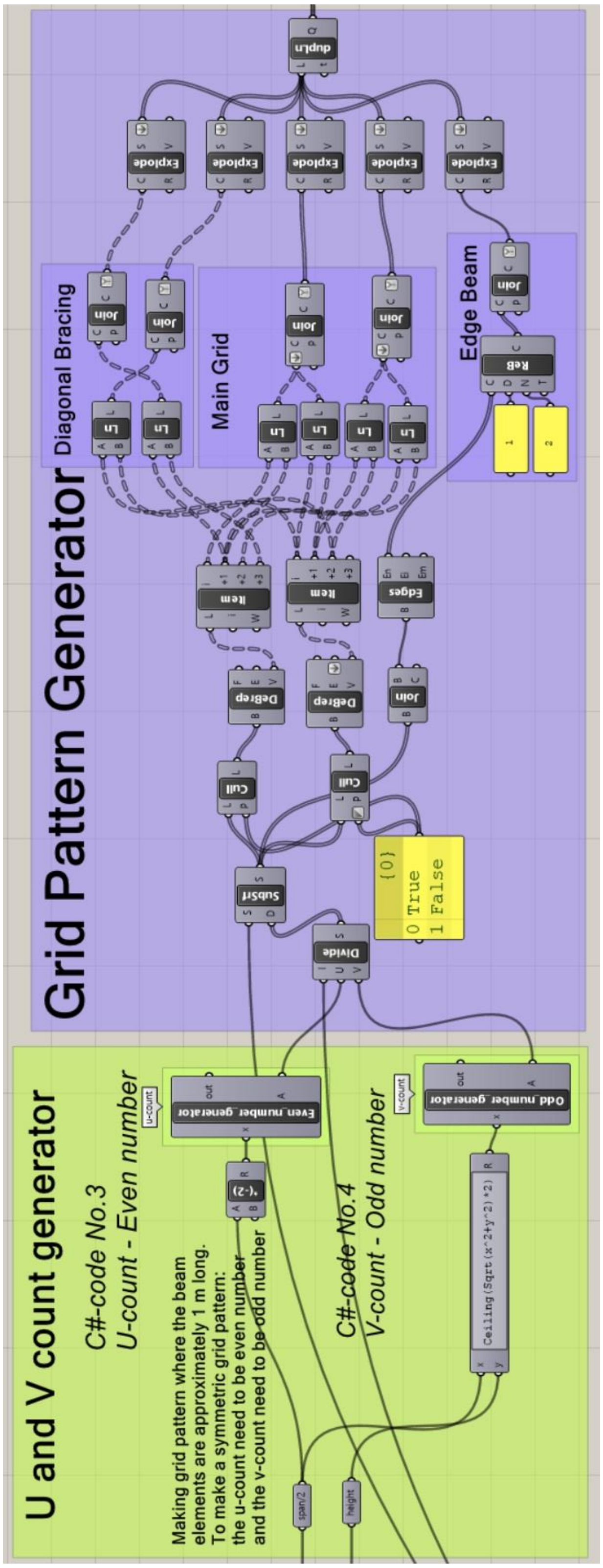
```
using


/// <summary>
/// This class will be instantiated on demand by the Script component.
/// </summary>
public class Script_Instance : GH_ScriptInstance
{
Utility functions

Members


  /**/
  private void RunScript(double height, ref object A)
  {
    // If height is equal to zero use geometry for the plate
    if (height == 0)
    {
      A = false;
    }
      // Else, use geometry for the Barrel Vault
    else
    {
      A = true;
    }

  }

  // <Custom additional code>

  // </Custom additional code>
}
```

C#-code No.2 Grid shell BV/P
Choose either barrel vault or plate surface



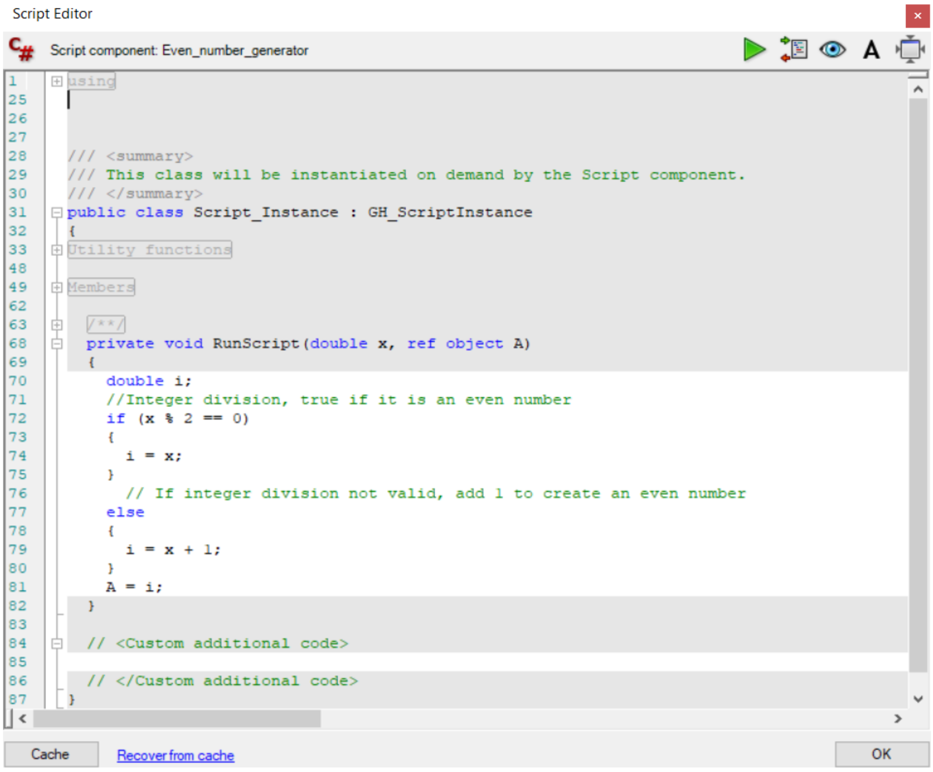
```
using



/// <summary>
/// This class will be instantiated on demand by the Script component.
/// </summary>
public class Script_Instance : GH_ScriptInstance
{
Utility functions

Members

  /**/
  private void RunScript(bool barrel, Surface surface_bar, Surface surface_plate, ref c
  {


    //Return surface of Barrel vault if Barrel vault is true
    if (barrel == true)
    {
      A = surface_bar;
    }
   //Return surface of Plate if Plate is true
    if (barrel == false)
    {
      A = surface_plate;
    }
  }

  // <Custom additional code>

  // </Custom additional code>
}
```

PART 2: Grid shell barrel vault/plate



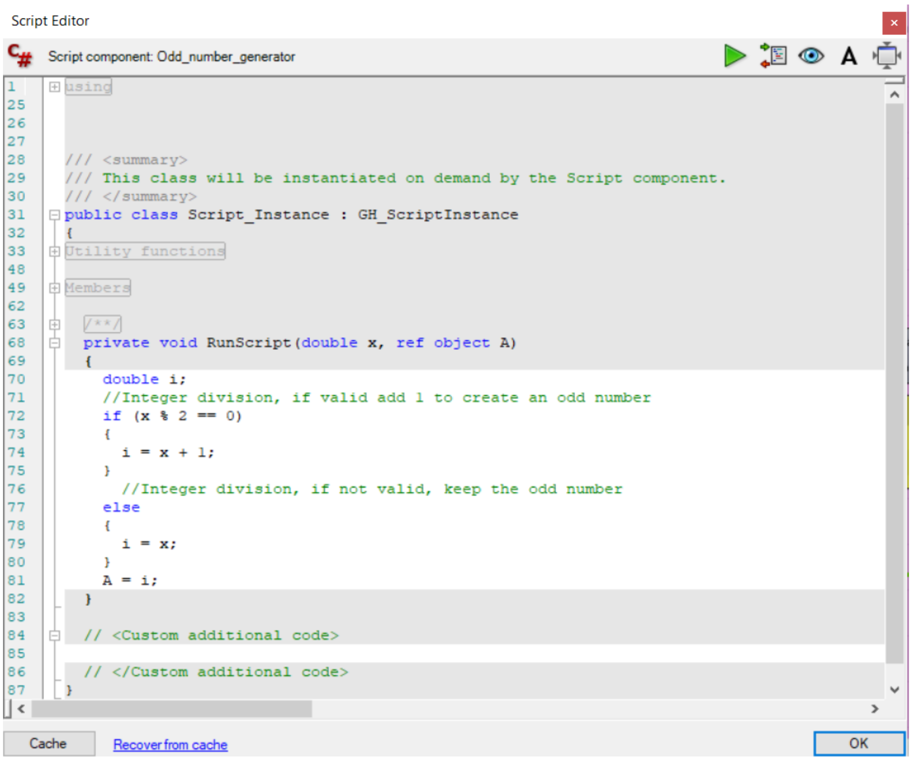
Making grid pattern where the beam elements are approximately 1 m long.
To make a symmetric grid pattern:
the u-count need to be even number
and the v-count need to be odd number

C#-code No.3 Grid shell BV/P
U-Count: generate even numbers



```
using


/// <summary>
/// This class will be instantiated on demand by the Script component.
/// </summary>
public class Script_Instance : GH_ScriptInstance
{
Utility functions

Members

  /**/
  private void RunScript(double x, ref object A)
  {
    double i;
    //Integer division, true if it is an even number
    if (x % 2 == 0)
    {
      i = x;
    }
      // If integer division not valid, add 1 to create an even number
    else
    {
      i = x + 1;
    }
    A = i;
  }

  // <Custom additional code>

  // </Custom additional code>
}
```

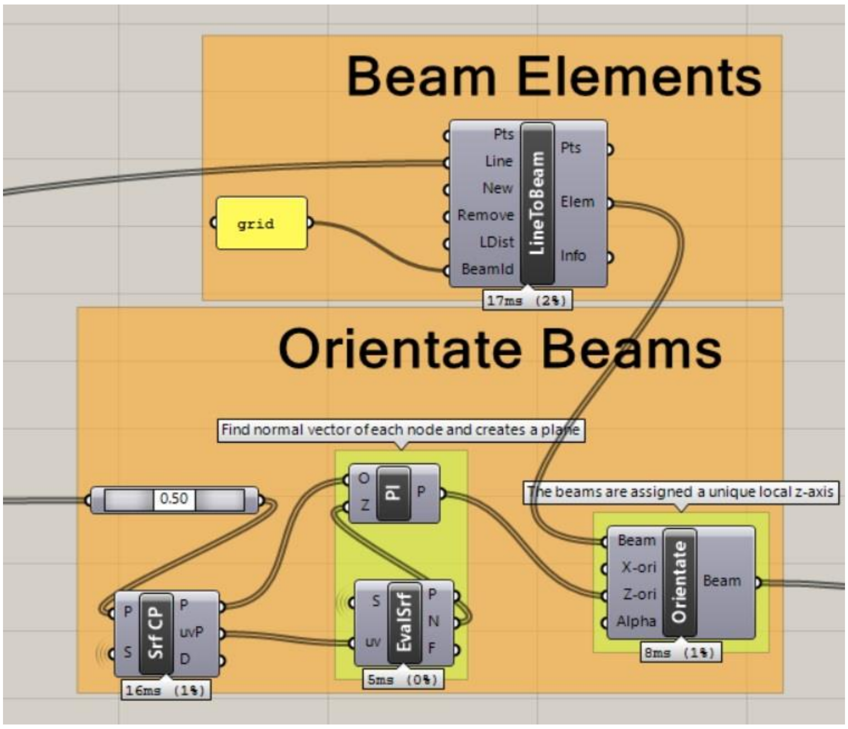
C#-code No.4 Grid shell BV/P
V-Count: generate odd numbers



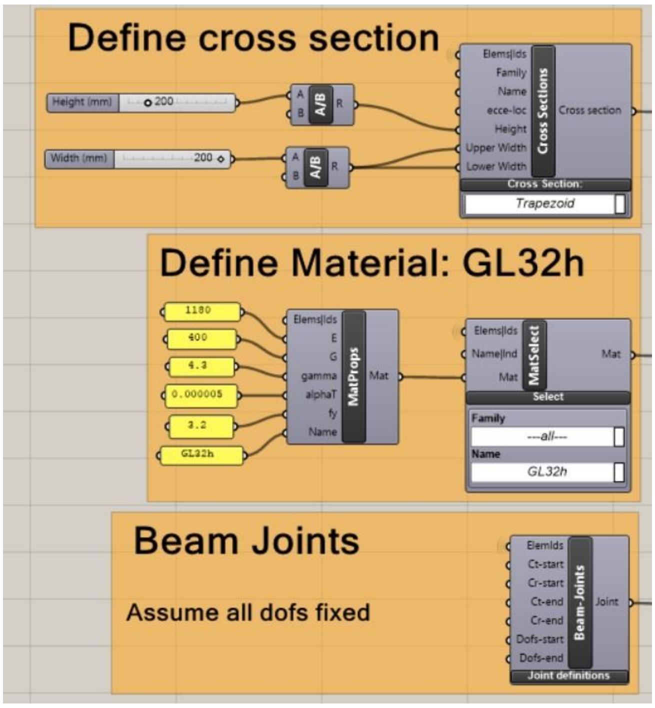
```
using


/// <summary>
/// This class will be instantiated on demand by the Script component.
/// </summary>
public class Script_Instance : GH_ScriptInstance
{
Utility functions

Members

  /**/
  private void RunScript(double x, ref object A)
  {
    double i;
    //Integer division, if valid add 1 to create an odd number
    if (x % 2 == 0)
    {
      i = x + 1;
    }
      //Integer division, if not valid, keep the odd number
    else
    {
      i = x;
    }
    A = i;
  }

  // <Custom additional code>

  // </Custom additional code>
}
```

PART 3: Grid shell barrel vault/plate

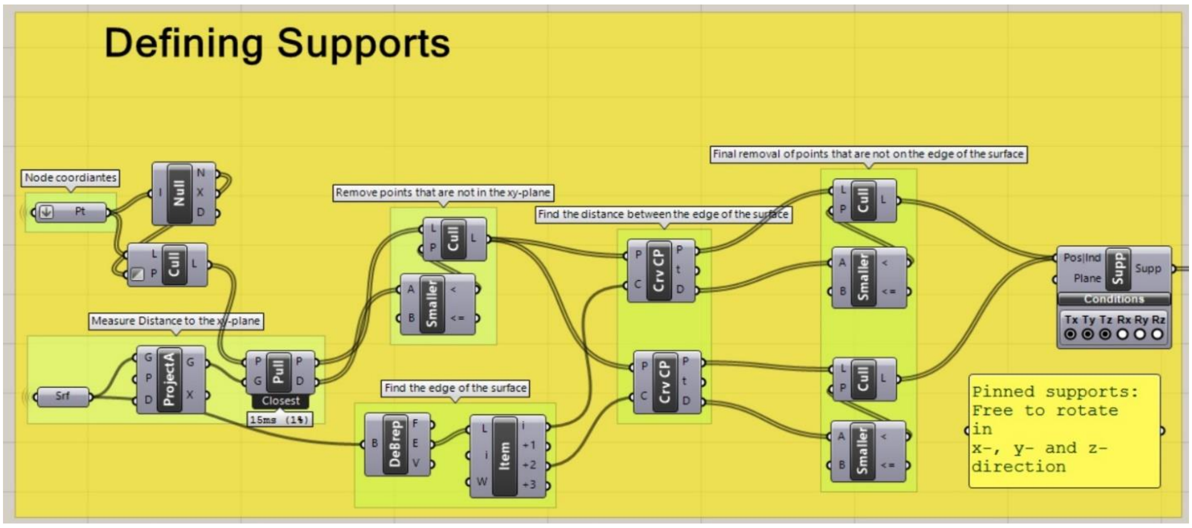
***Create beam elements**



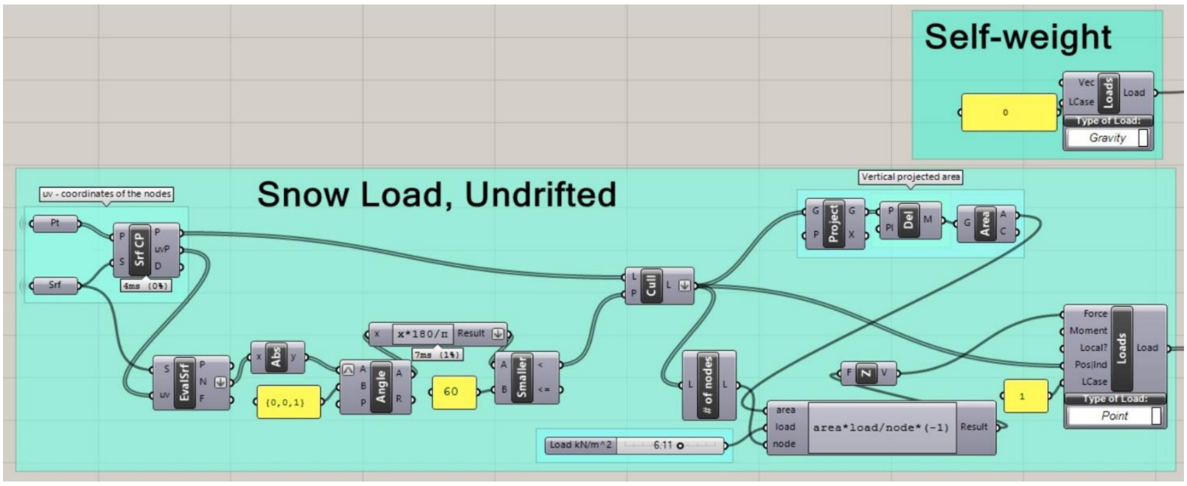
*** Assign material, cross section and joint properties**

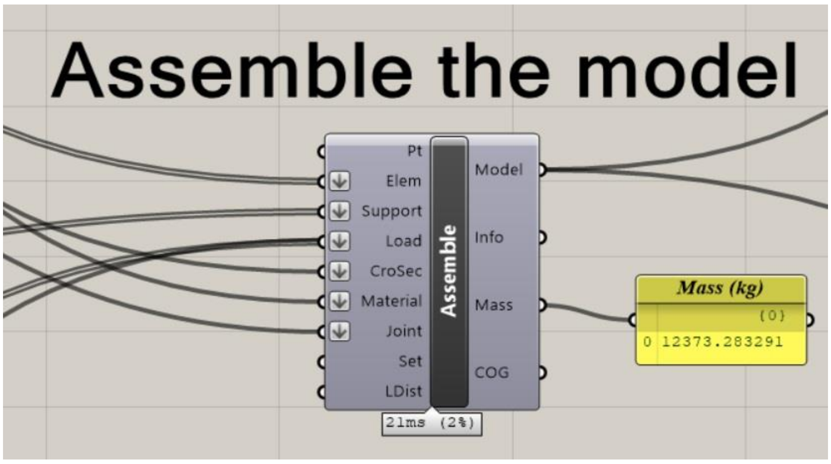


***Define supports at relevant points**


Defining Supports
Node coordiantes
Pt
Null
Cull
Measure Distance to the xy-plane
Srf
ProjectA
Pull
Closest
16ms (1%)
Remove points that are not in the xy-plane
Smaller
Find the edge of the surface
DeBrep
Item
Find the distance between the edge of the surface
Crv CP
Final removal of points that are not on the edge of the surface
Supp
Pos|Ind
Plane
Conditions
Tx Ty Tz Rx Ry Rz
Pinned supports:
Free to rotate
in
x-, y- and z-
direction

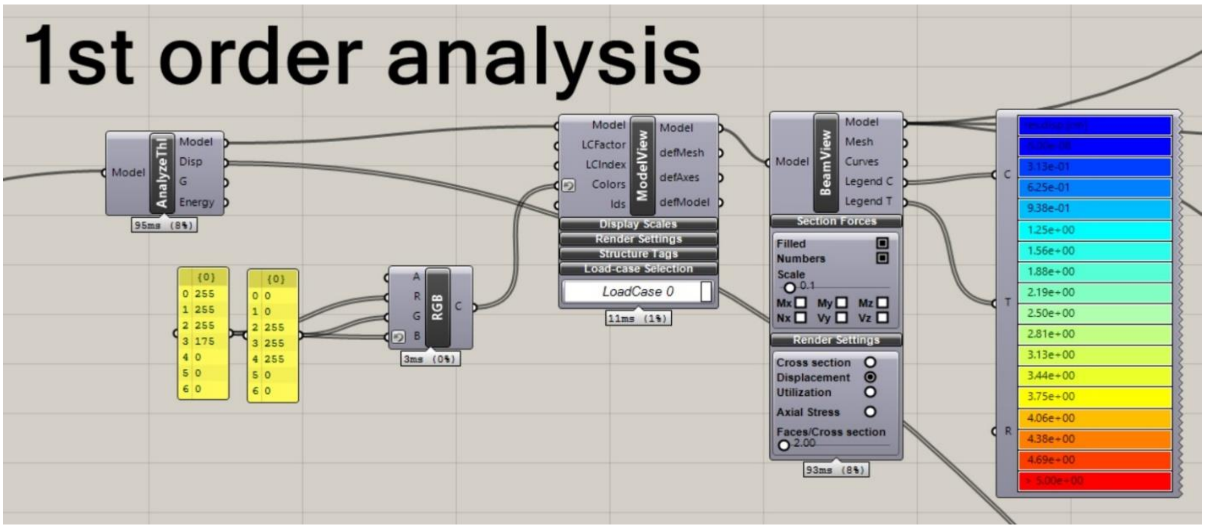
***Define Loads on structure**



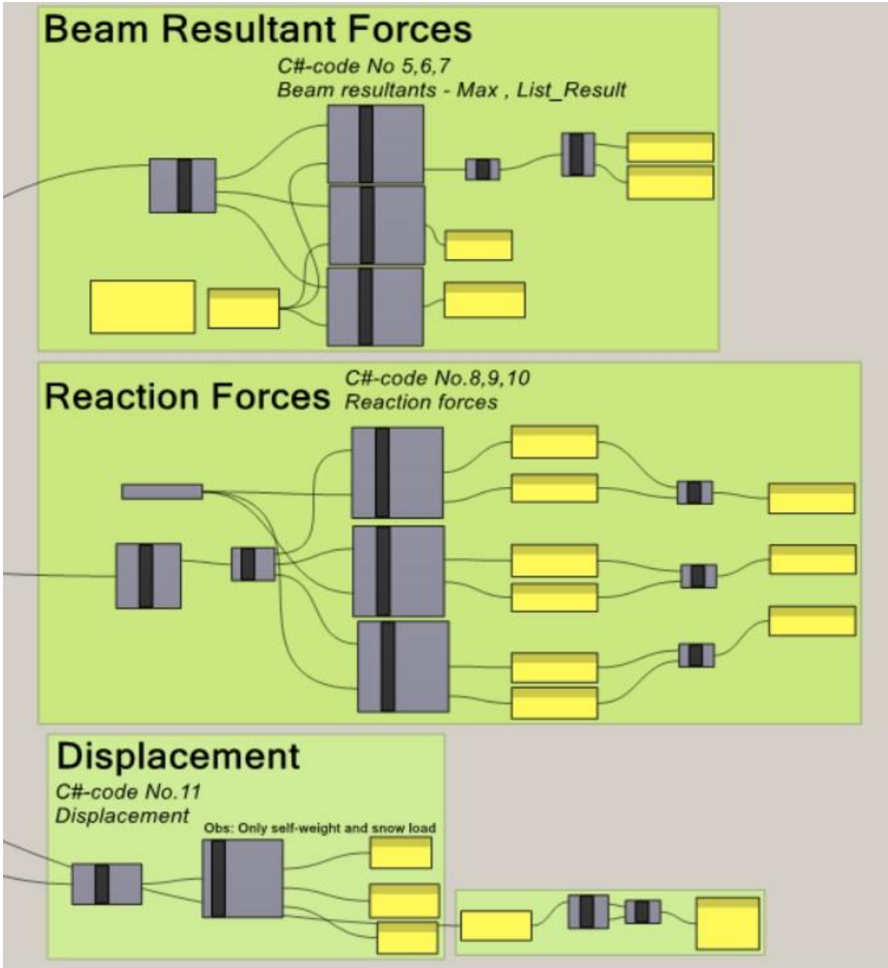


PART 4: Grid shell barrel vault/plate

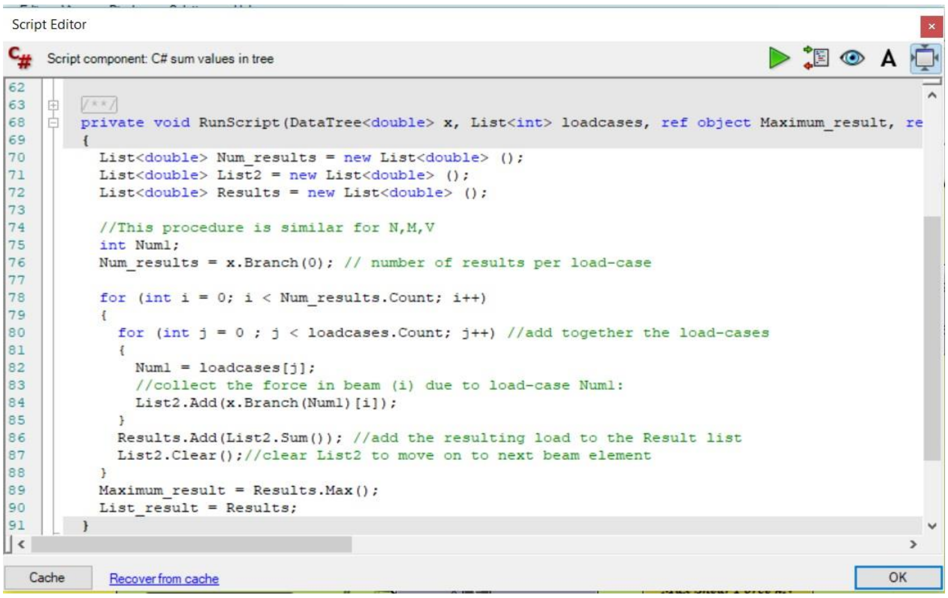
***1st order analysis**



Results:



C#-code No.5,6,7 Grid shell BV/P
Sum together resulting forces from different load cases



```
/**/
private void RunScript(DataTree<double> x, List<int> loadcases, ref object Maximum_result, re
{
  List<double> Num_results = new List<double> ();
  List<double> List2 = new List<double> ();
  List<double> Results = new List<double> ();

  //This procedure is similar for N,M,V
  int Numl;
  Num_results = x.Branch(0); // number of results per load-case

  for (int i = 0; i < Num_results.Count; i++)
  {
    for (int j = 0 ; j < loadcases.Count; j++) //add together the load-cases
    {
      Numl = loadcases[j];
      //collect the force in beam (i) due to load-case Numl:
      List2.Add(x.Branch(Numl)[i]);
    }
    Results.Add(List2.Sum()); //add the resulting load to the Result list
    List2.Clear();//clear List2 to move on to next beam element
  }
  Maximum_result = Results.Max();
  List_result = Results;
}
```

C#-code No.8,9,10 Grid shell BV/P
Reaction forces (Rx,Ry,Rz)
This code is equal to C# code No. 5,6,7 in the concrete model

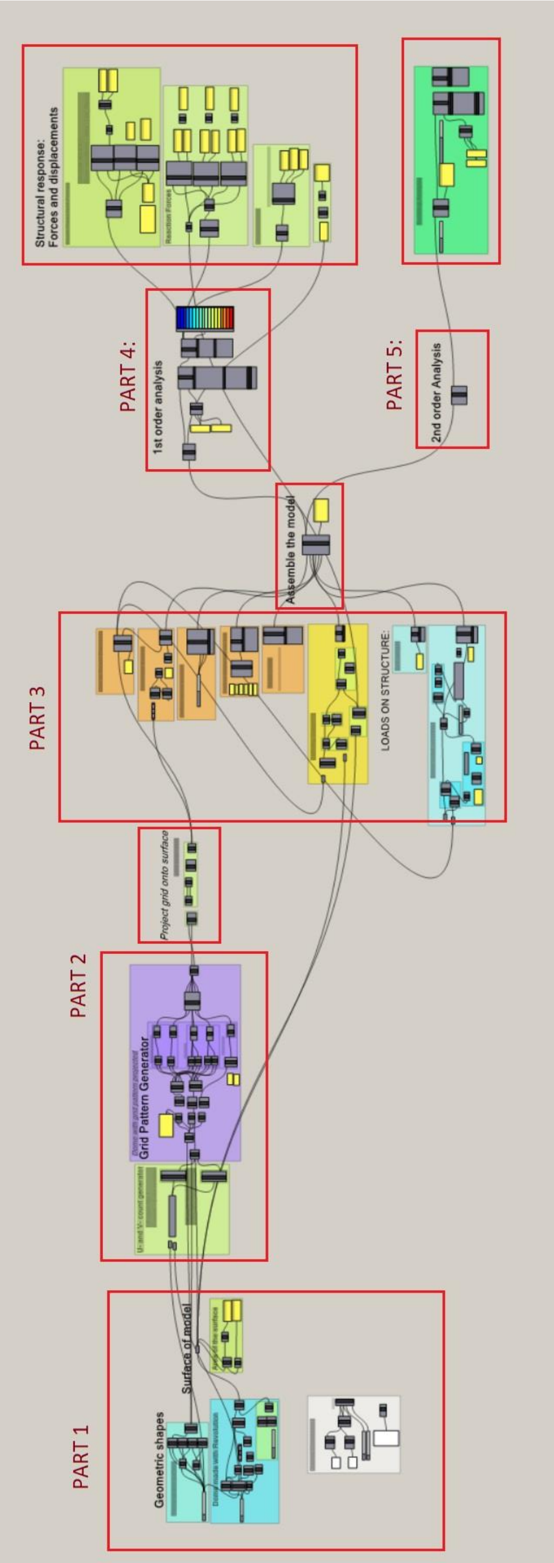
C#-code No.11 Grid shell BV/P
Find displacement due to self-weigth, snow and total displ.
This code is equal to C#-code No.3 Concrete

PART 5: Grid shell barrel vault/plate

*2nd order analysis
* Calculation of buckling modes

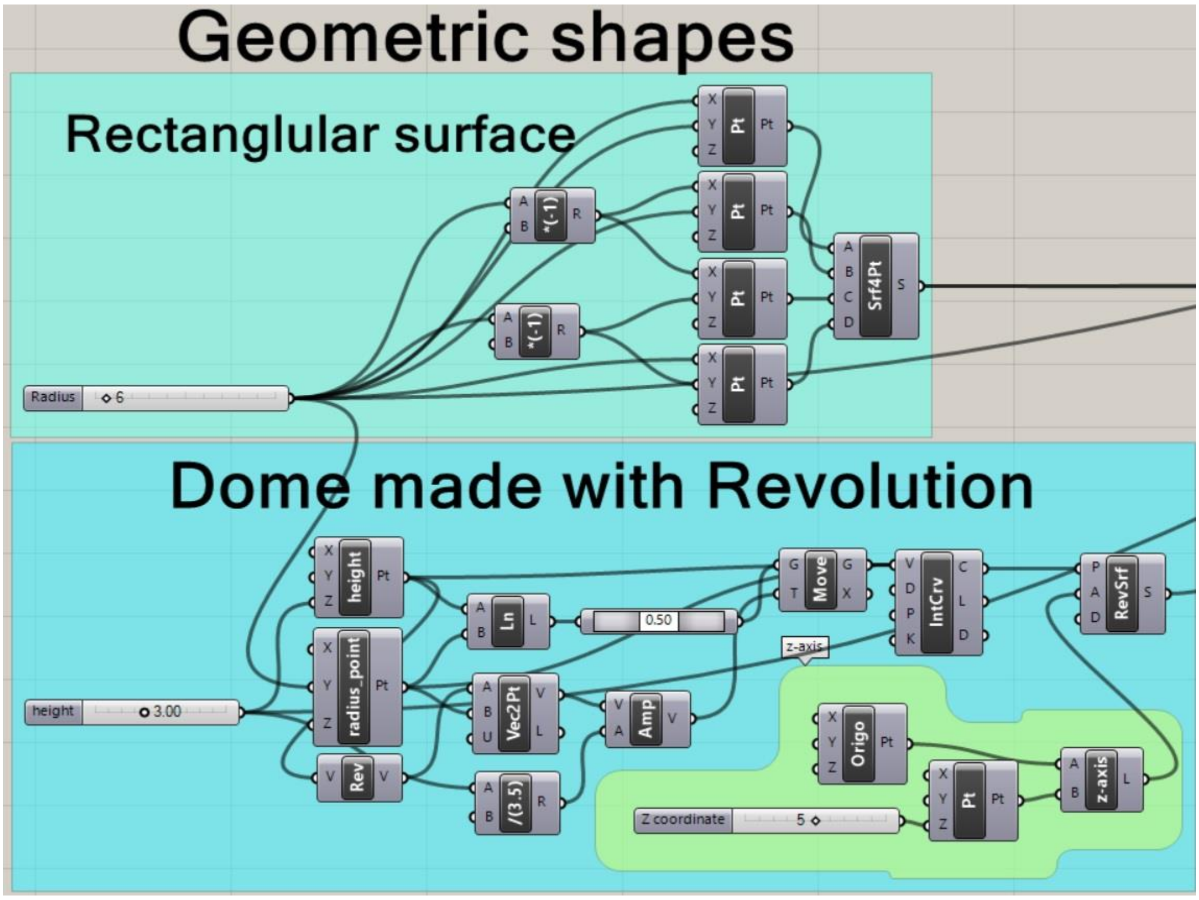
Same procedure as for concrete shell

3. Grid shell model Dome (D)



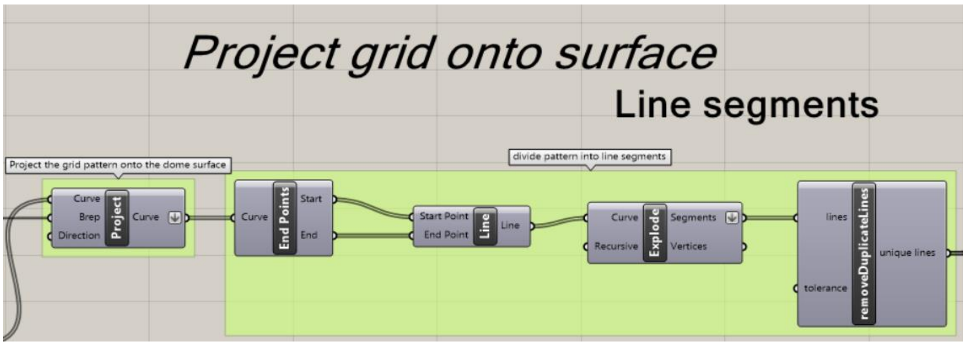
For the Grid shell dome PART 3, 4 and 5, will be similar to grid shell barrel vault and plate Hence, these part are not presented. The only difference is the creation of the geometric shape and the projection of the grid pattern onto the surface

PART 1: Grid shell Dome

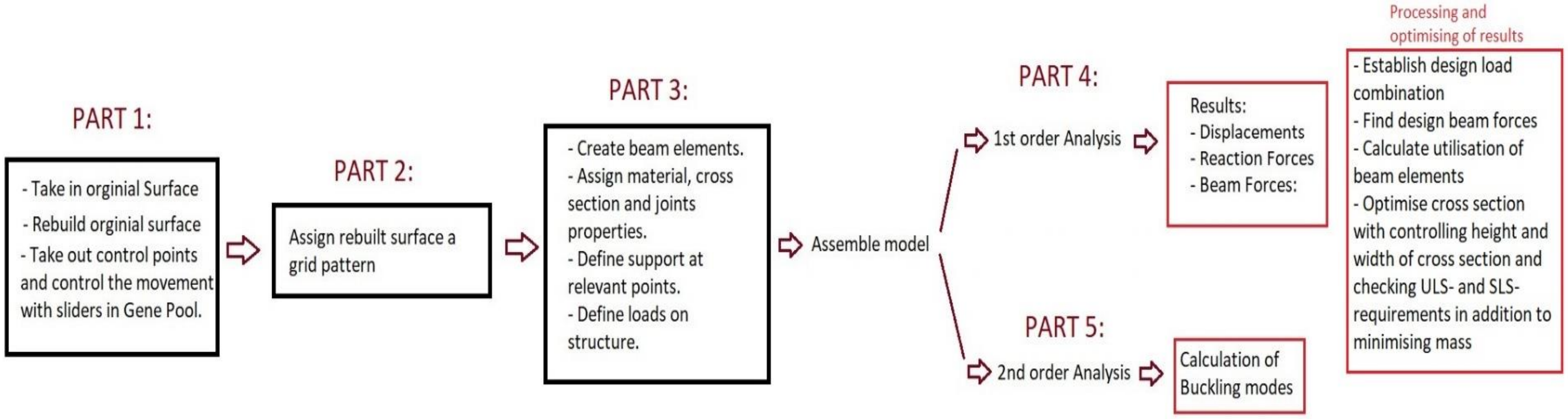


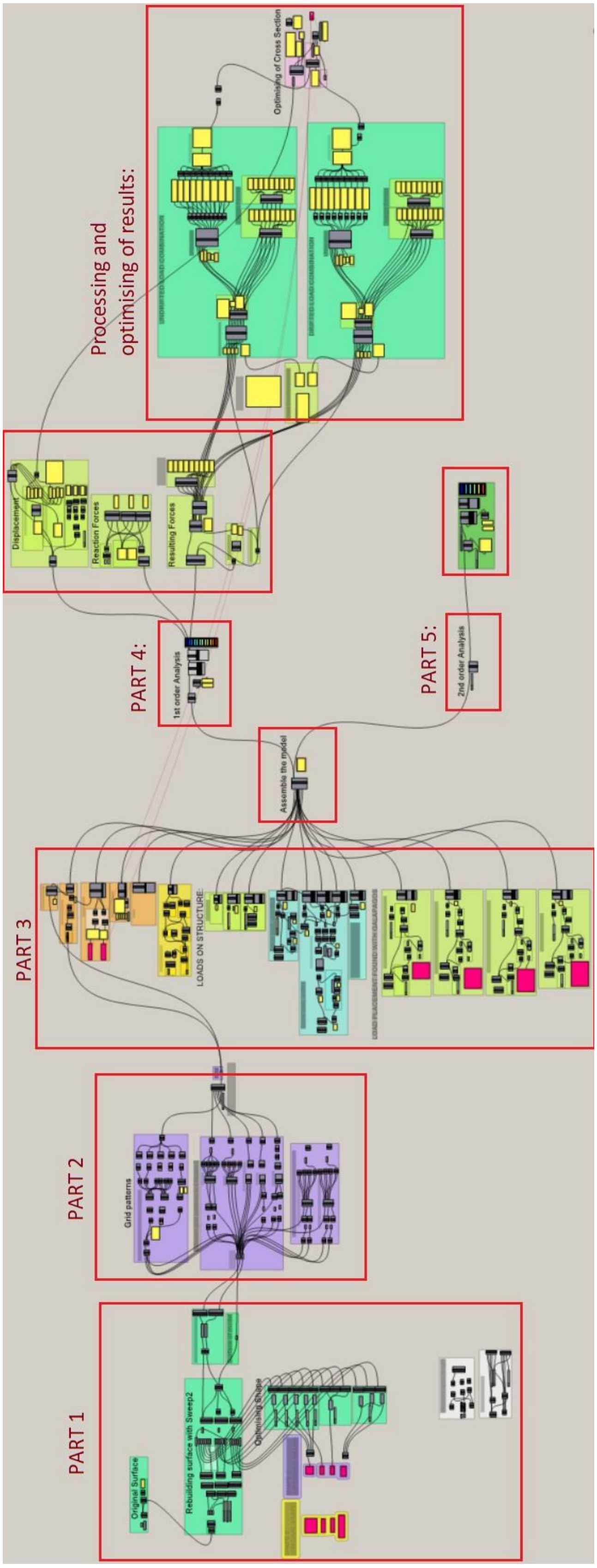
PART 2: Grid shell Dome

The grid pattern is created similarly to the barrel vault
The grid pattern is projected onto the dome-surface


Project grid onto surface
Line segments
Project the grid pattern onto the dome surface
divide pattern into line segments
Project
End Points
Line
Explode
removeDuplicateLines
unique lines
tolerance

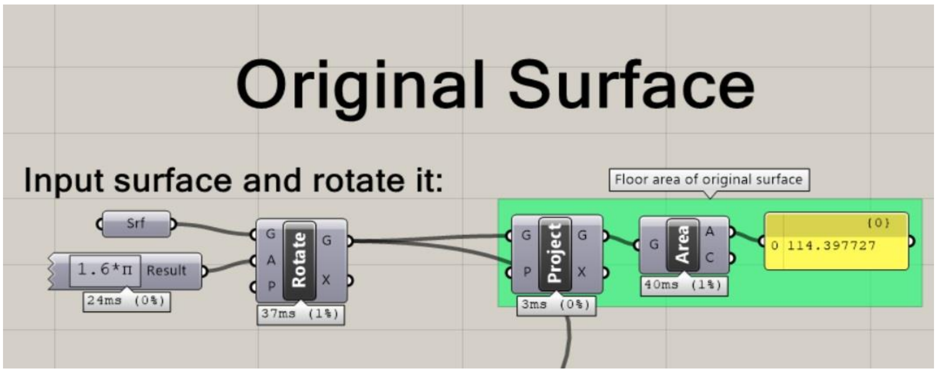
C | Grasshopper code for Case Study


PART 1:
- Take in orginial Surface
- Rebuild orginial surface
- Take out control points and control the movement with sliders in Gene Pool.
PART 2:
Assign rebuilt surface a grid pattern
PART 3:
- Create beam elements.
- Assign material, cross section and joints properties.
- Define support at relevant points.
- Define loads on structure.
Assemble model
PART 4:
1st order Analysis
Results:
- Displacements
- Reaction Forces
- Beam Forces:
PART 5:
2nd order Analysis
Calculation of Buckling modes
Processing and optimising of results
- Establish design load combination
- Find design beam forces
- Calculate utilisation of beam elements
- Optimise cross section with controlling height and width of cross section and checking ULS- and SLS-requirements in addition to minimising mass


PART 1
Original Surface
Rebuilding surface with Sweep2
Optimising Shape
PART 2
Grid patterns
PART 3
LOADS ON STRUCTURE
LOAD PLACEMENT FOUND WITH GALAPAGOS
Assemble the model
PART 4:
1st order Analysis
PART 5:
2nd order Analysis
Displacement
Reaction Forces
Resulting Forces
Processing and optimising of results:
Optimising of Cross Section

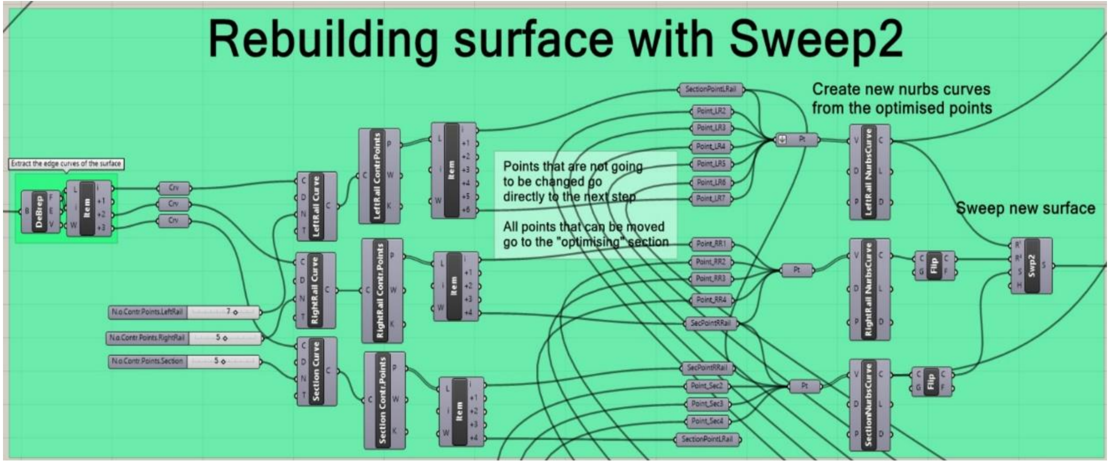
PART 1

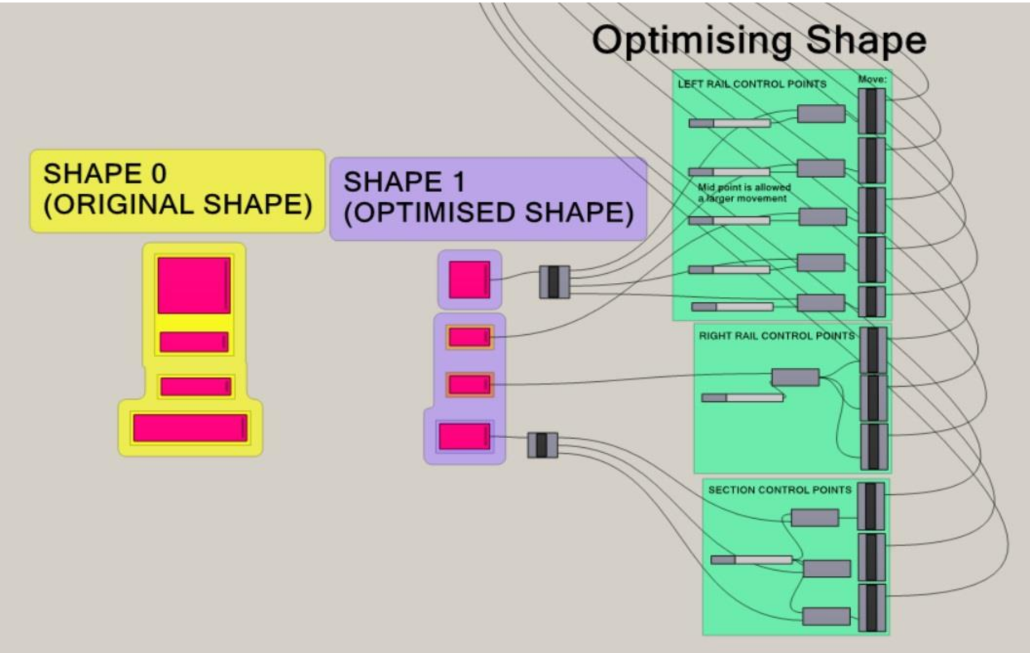
*** Take in the original surface**



*** Rebuild original surface**

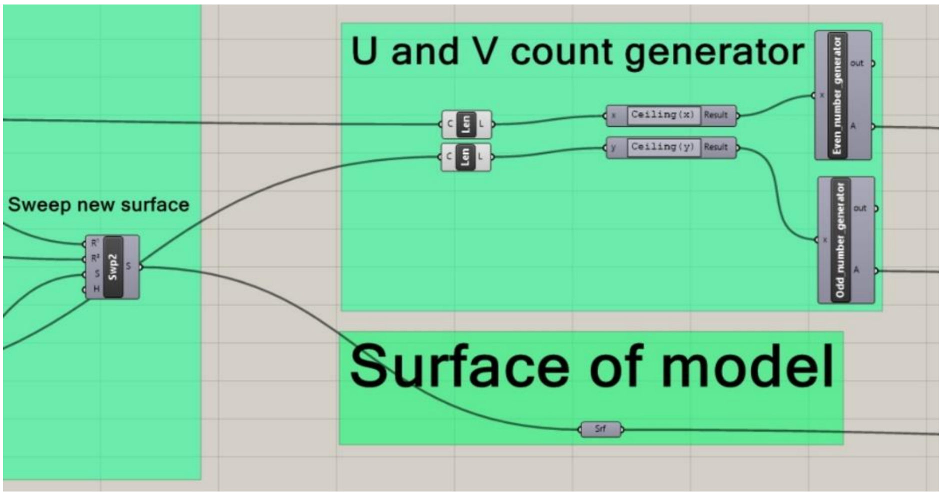
*** Take out control points and control the movement with sliders in Gene Pool**

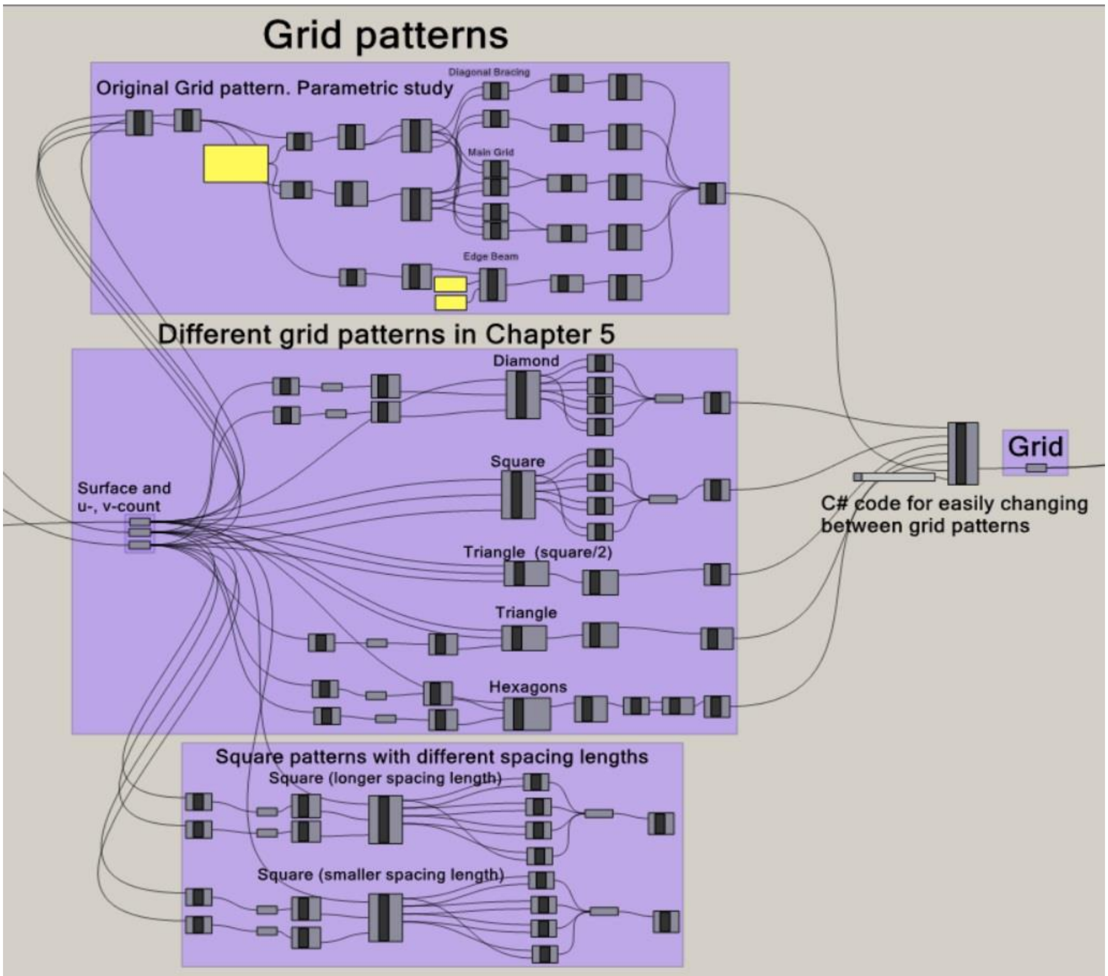




PART 2

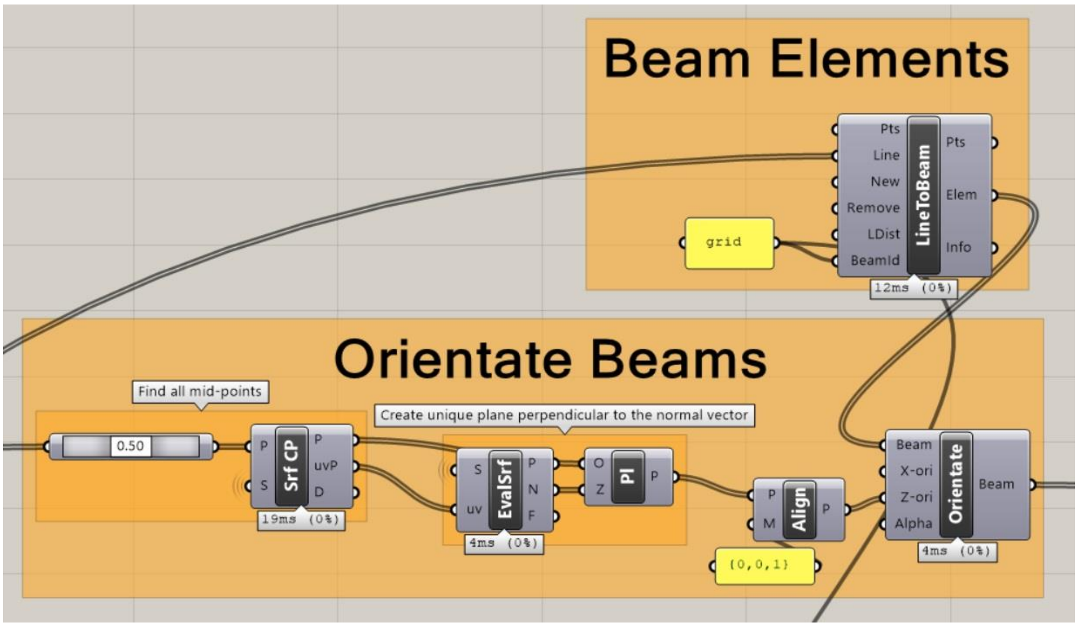
***Assign rebuild surface a grid pattern**



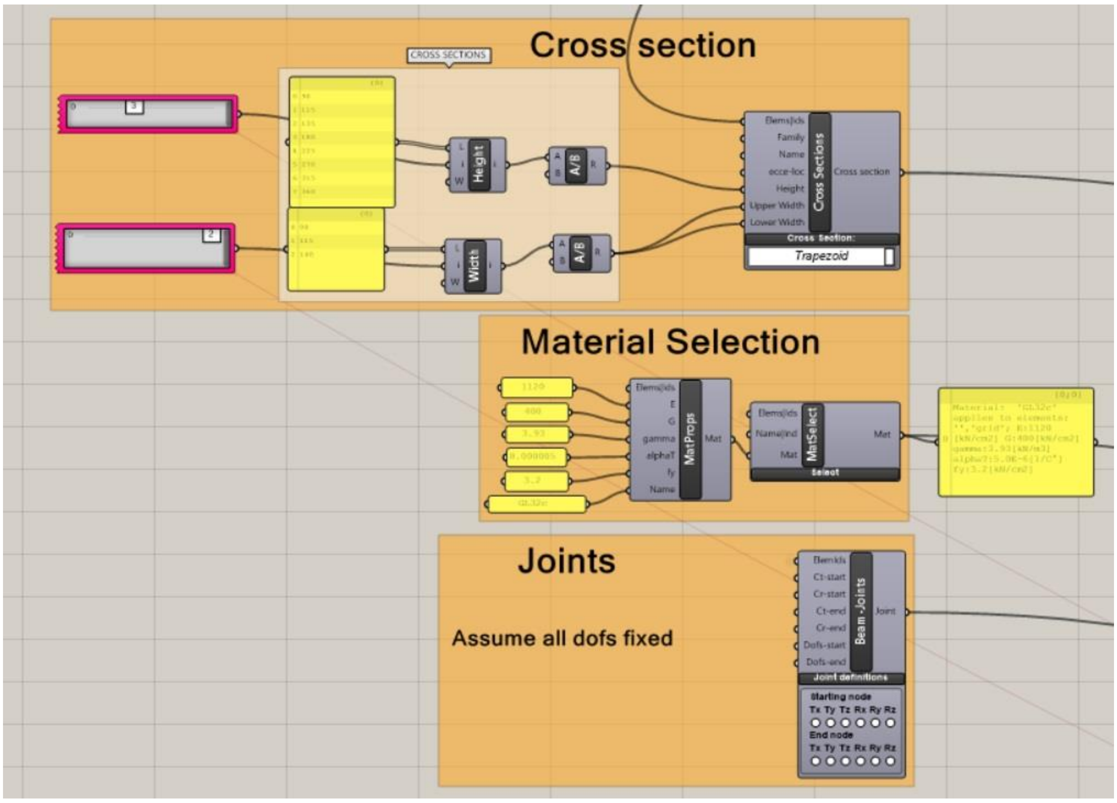

Grid patterns
Original Grid pattern. Parametric study
Diagonal Bracing
Main Grid
Edge Beam
Different grid patterns in Chapter 5
Diamond
Square
Surface and
u-, v-count
Triangle (square/2)
Triangle
Hexagons
Square patterns with different spacing lengths
Square (longer spacing length)
Square (smaller spacing length)
Grid
C# code for easily changing
between grid patterns

PART 3

*Create beam elements

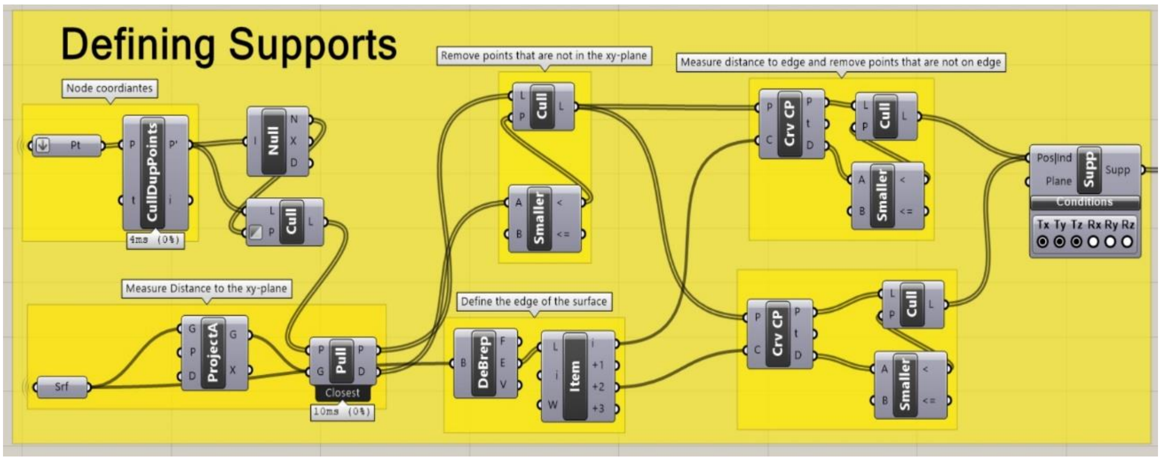


*Assign material, cross section and joint properties

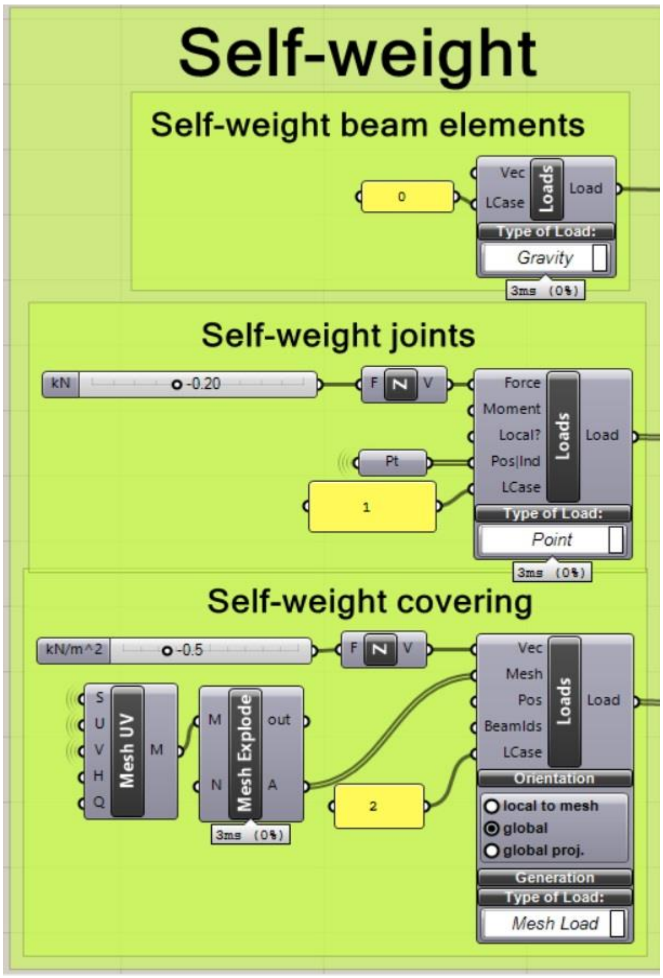


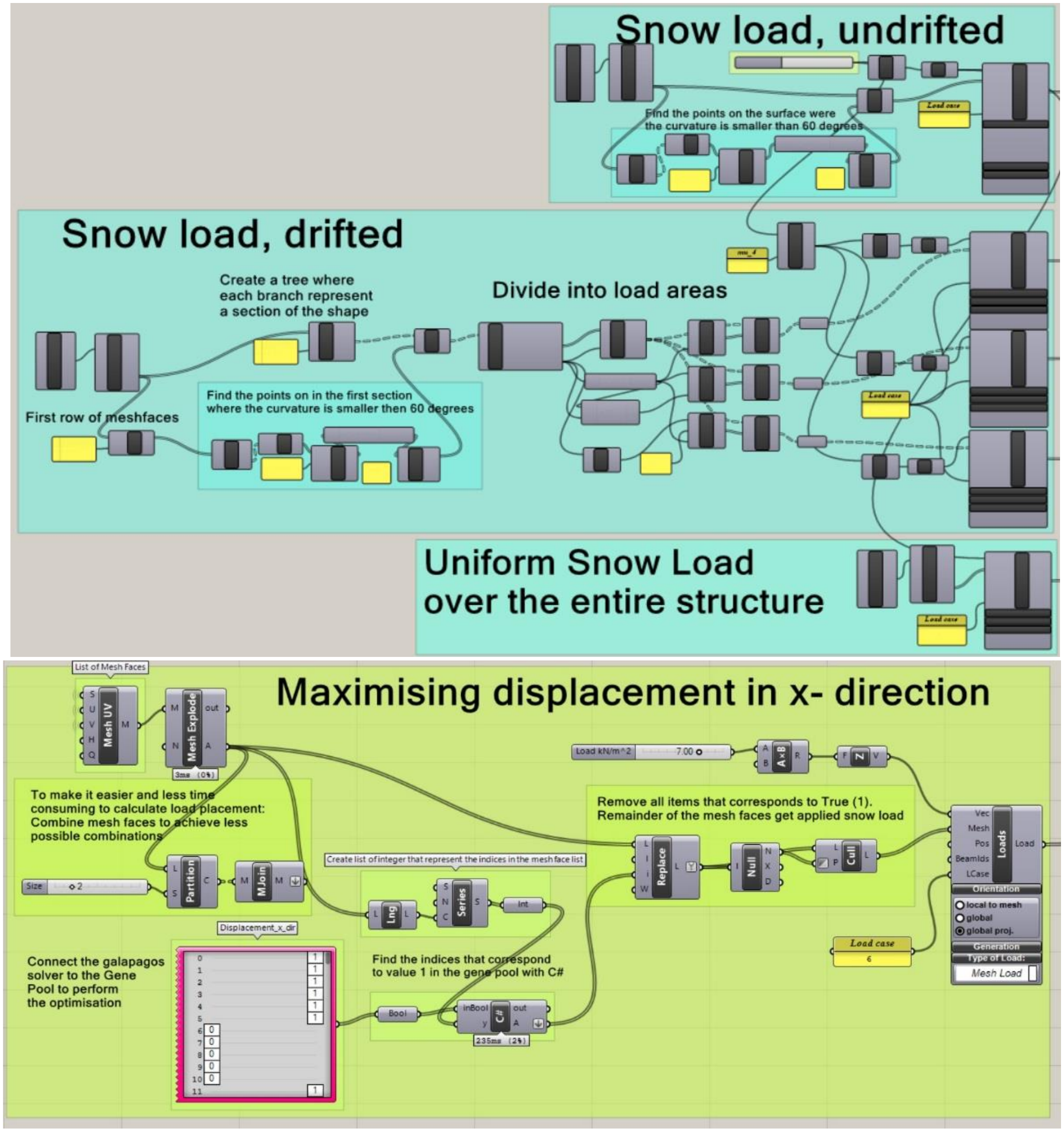
```
Material: 'GL32c'
applies to elements:
'','grid'; E:1120
[kN/cm2] G:400[kN/cm2]
gamma:3.93[kN/m3]
alphaT:5.0E-6[1/C°]
fy:3.2[kN/cm2]
```

*Define supports at relevant points

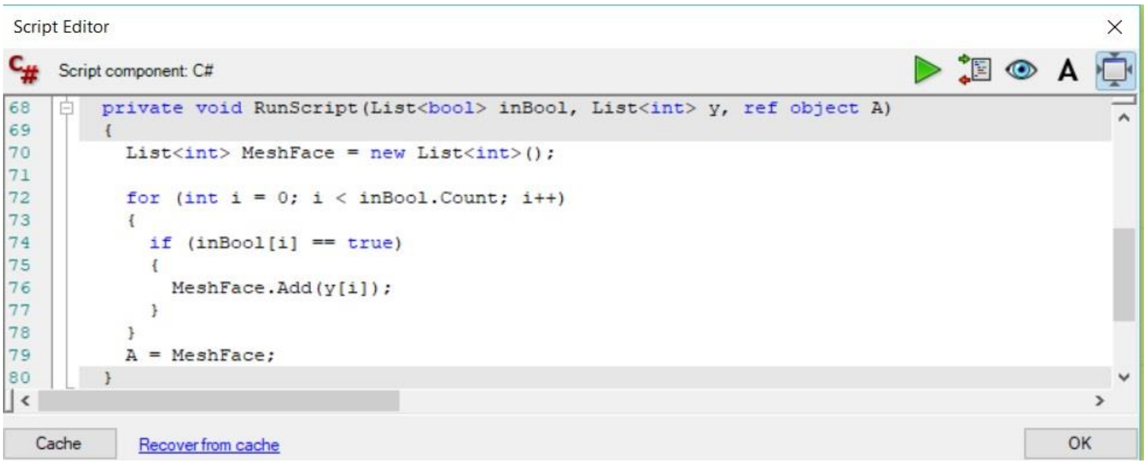


*Define loads on structure

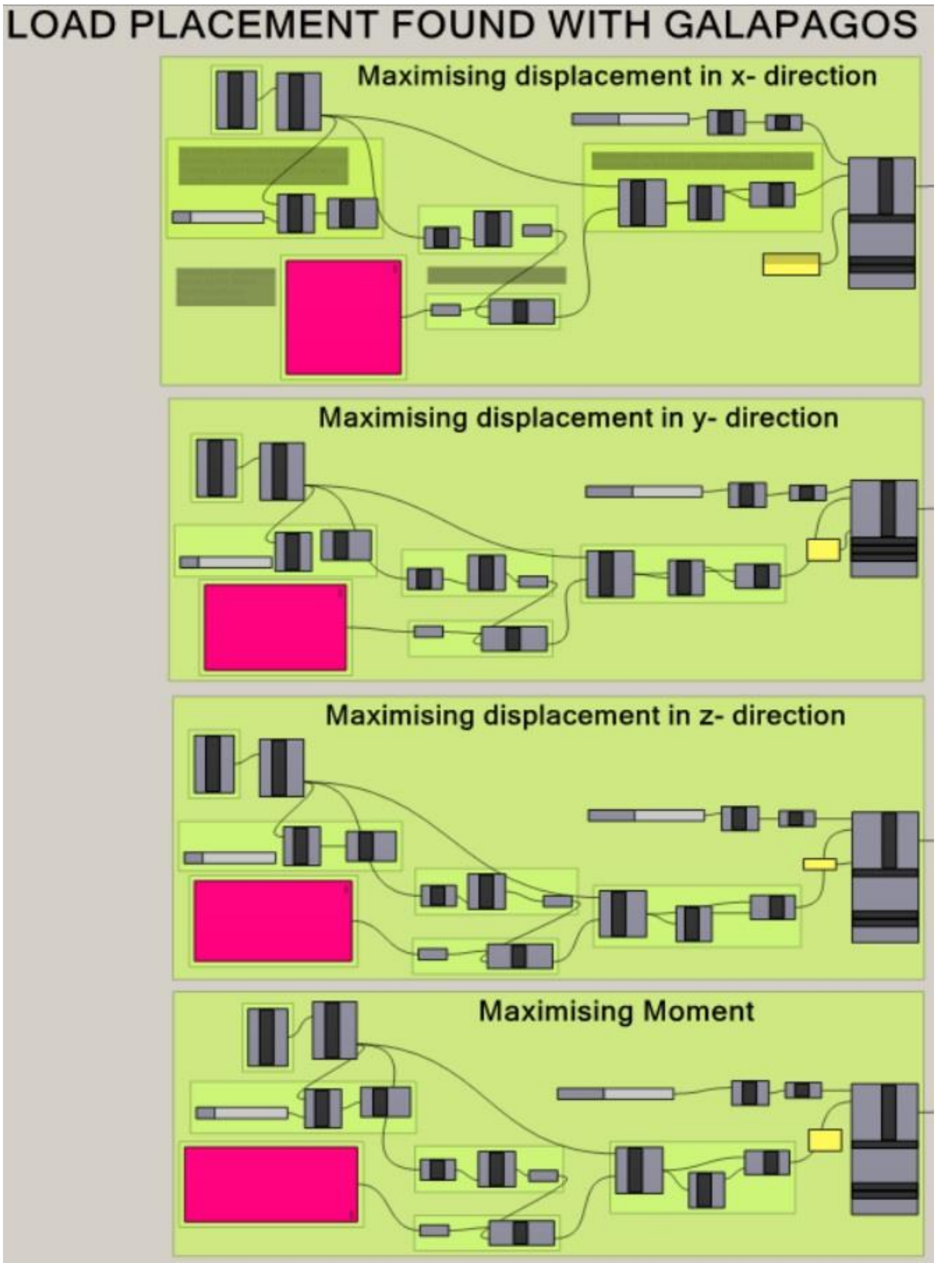



Snow load, undrifted
Find the points on the surface were the curvature is smaller than 60 degrees
Load case
Snow load, drifted
Create a tree where each branch represent a section of the shape
Divide into load areas
First row of meshfaces
Find the points on in the first section where the curvature is smaller then 60 degrees
Load case
Uniform Snow Load over the entire structure
Load case
Maximising displacement in x- direction
List of Mesh Faces
Mesh UV
Mesh Explode
Load kN/m^2
7.00
A×B
Z
To make it easier and less time consuming to calculate load placement: Combine mesh faces to achieve less possible combinations
Size
2
Partition
MJoin
Create list of integer that represent the indices in the mesh face list
Lng
Series
Int
Remove all items that corresponds to True (1). Remainder of the mesh faces get applied snow load
Replace
Null
Cull
Loads
Vec
Mesh
Pos
BeamIds
LCase
Load
Orientation
local to mesh
global
global proj.
Generation
Type of Load:
Mesh Load
Load case
6
Displacement_x_dir
Connect the galapagos solver to the Gene Pool to perform the optimisation
Find the indices that correspond to value 1 in the gene pool with C#
Bool
inBool
y
out
A
C#

C# code for retrieving the mesh faces which corresponds to True-value

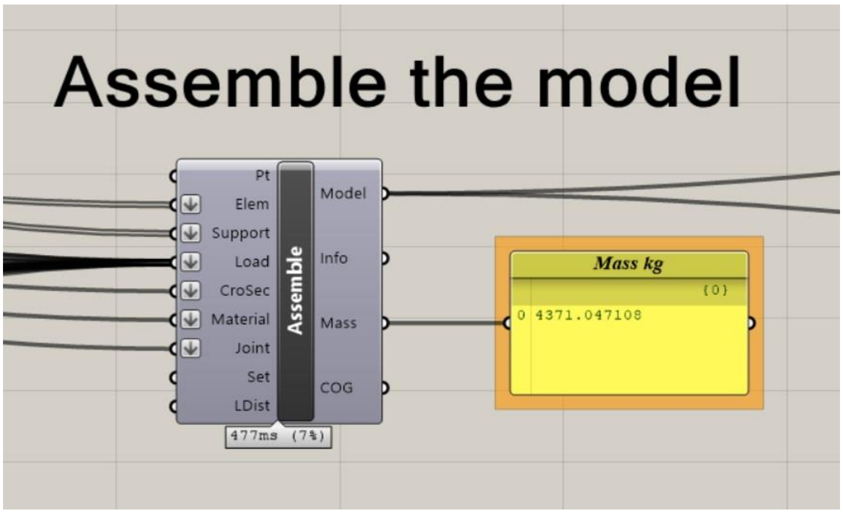


```
private void RunScript(List<bool> inBool, List<int> y, ref object A)
{
  List<int> MeshFace = new List<int>();

  for (int i = 0; i < inBool.Count; i++)
  {
    if (inBool[i] == true)
    {
      MeshFace.Add(y[i]);
    }
  }
  A = MeshFace;
}
```



LOAD PLACEMENT FOUND WITH GALAPAGOS
Maximising displacement in x- direction
Maximising displacement in y- direction
Maximising displacement in z- direction
Maximising Moment

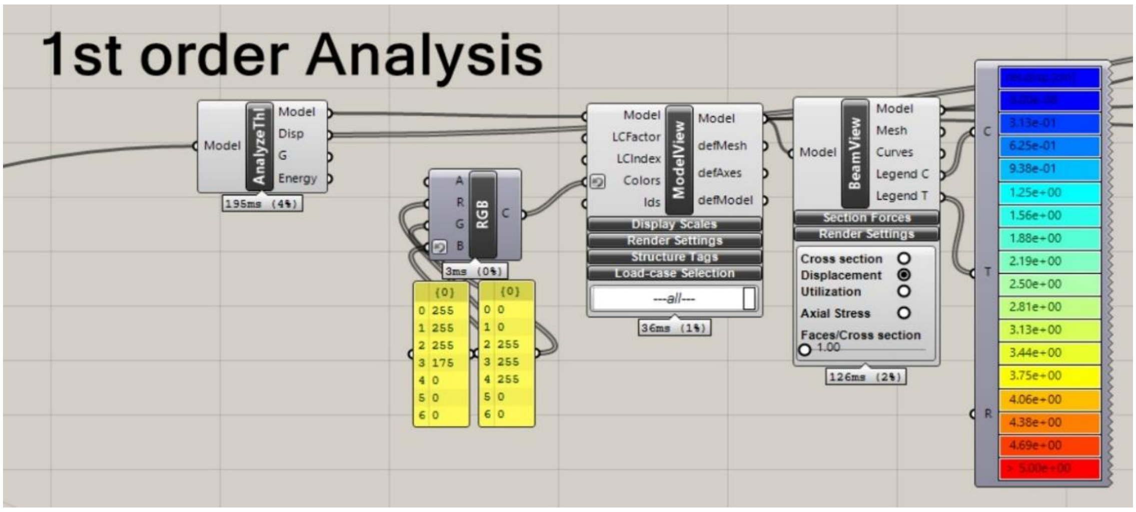
***Assemble model:**



```
0 4371.047108
```

PART 4:

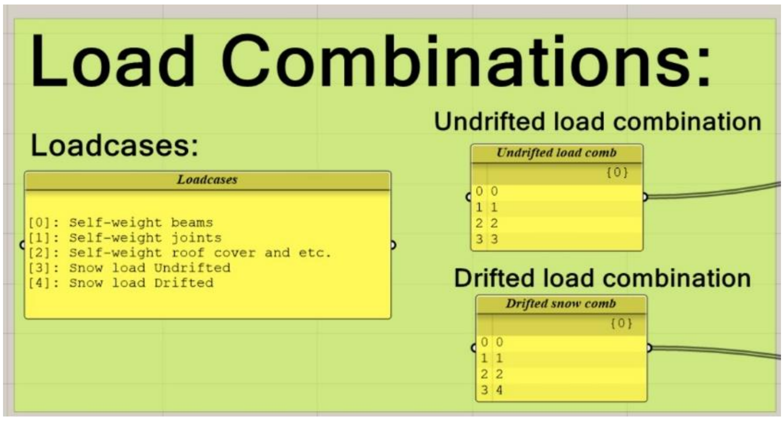
***1st order analysis**



{0}	
0	255
1	255
2	255
3	175
4	0
5	0
6	0

{0}	
0	0
1	0
2	255
3	255
4	255
5	0
6	0

Cross section ○
Displacement ●
Utilization ○
Axial Stress ○
Faces/Cross section ○ 1.00



Loadcases

```
[0]: Self-weight beams
[1]: Self-weight joints
[2]: Self-weight roof cover and etc.
[3]: Snow load Undrifted
[4]: Snow load Drifted
```

Undrifted load comb

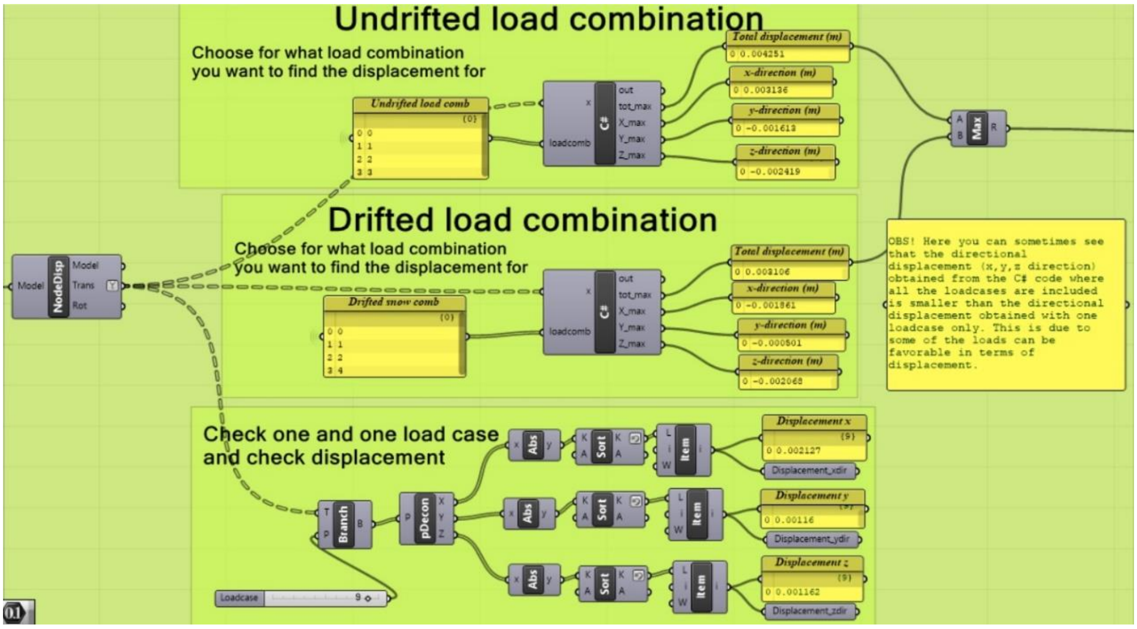
{0}	
0	0
1	1
2	2
3	3

Drifted snow comb

{0}	
0	0
1	1
2	2
3	4

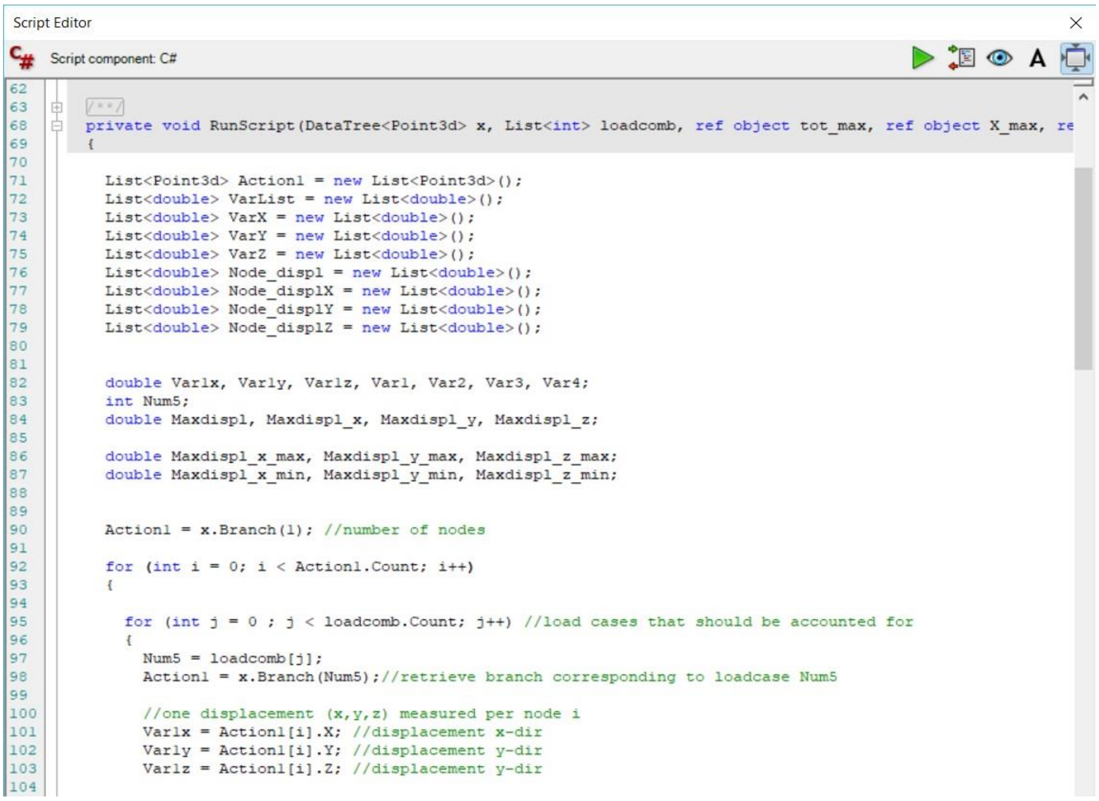
Results:

*Displacements

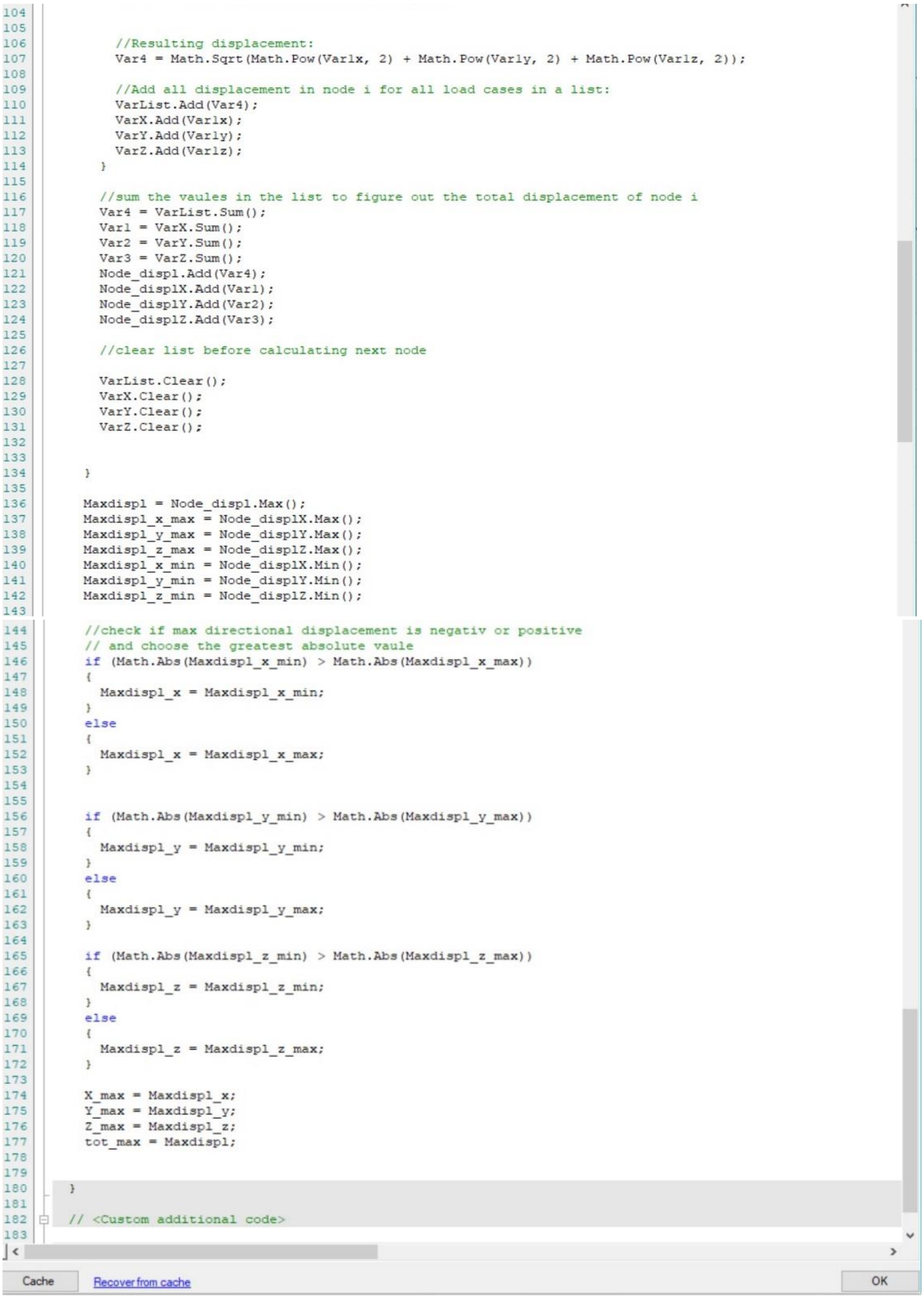


OBS! Here you can sometimes see that the directional displacement (x,y,z direction) obtained from the C# code where all the loadcases are included is smaller than the directional displacement obtained with one loadcase only. This is due to some of the loads can be favorable in terms of displacement.

C # code for adding together displacements for different loadcombinations

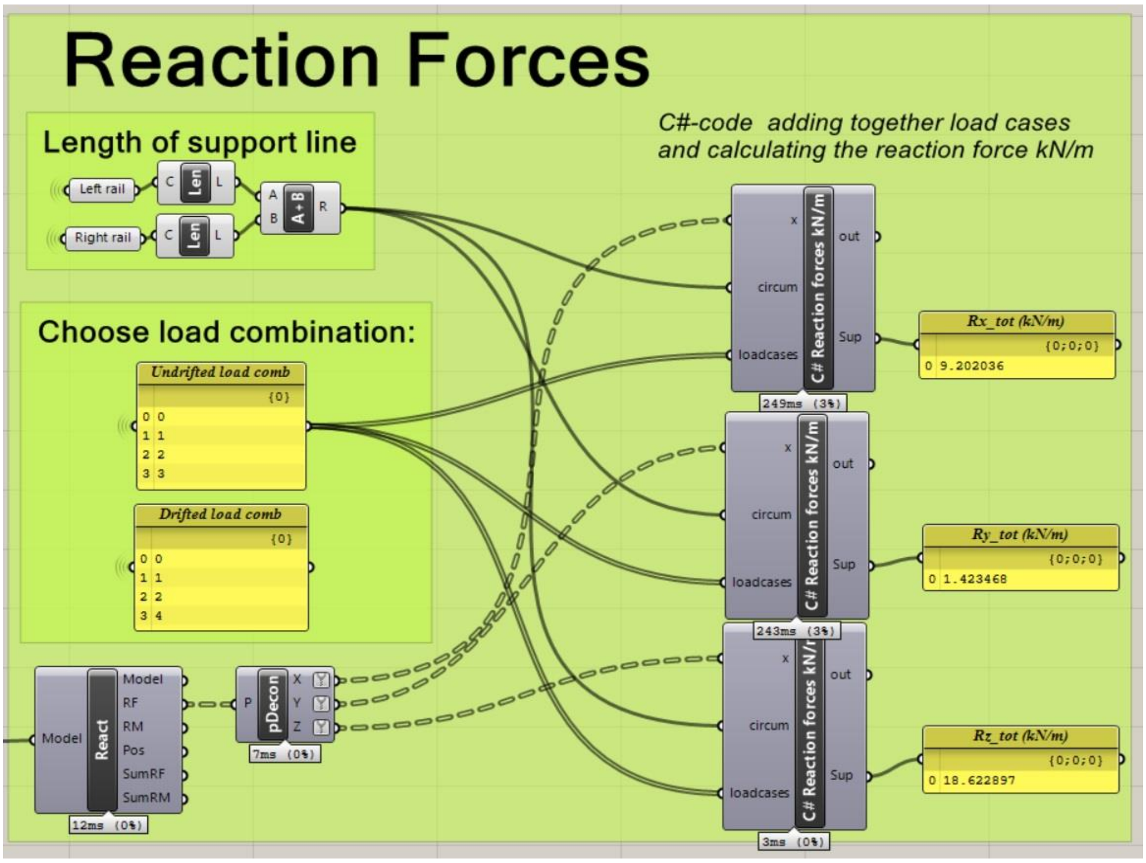


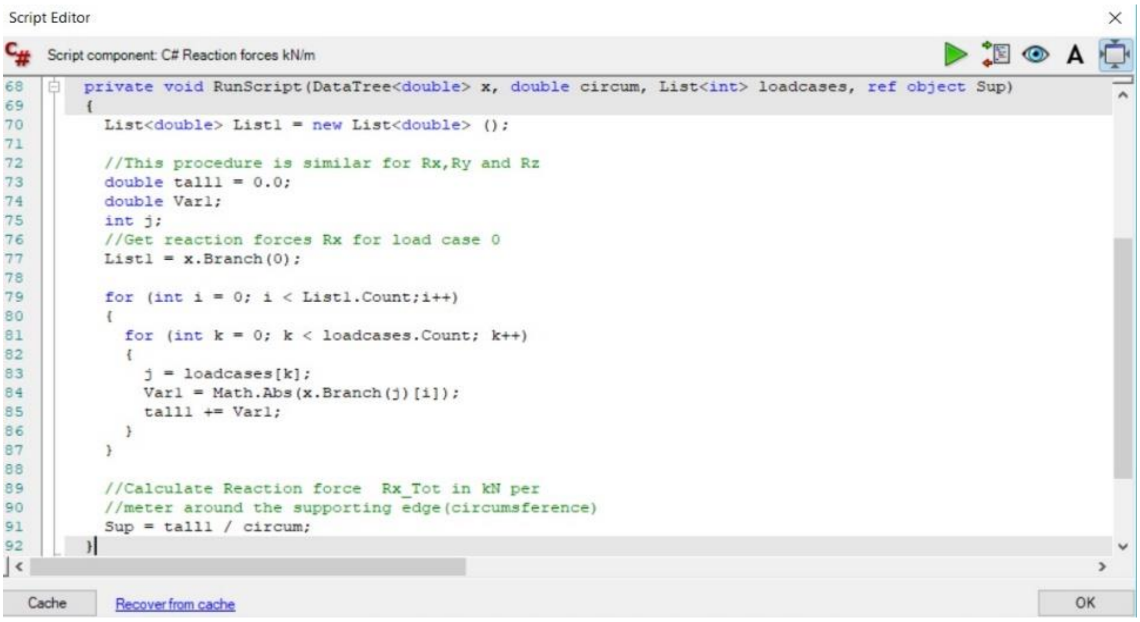
```
/**/
private void RunScript(DataTree<Point3d> x, List<int> loadcomb, ref object tot_max, ref object X_max, re
{

  List<Point3d> Action1 = new List<Point3d>();
  List<double> VarList = new List<double>();
  List<double> VarX = new List<double>();
  List<double> VarY = new List<double>();
  List<double> VarZ = new List<double>();
  List<double> Node_displ = new List<double>();
  List<double> Node_displX = new List<double>();
  List<double> Node_displY = new List<double>();
  List<double> Node_displZ = new List<double>();


  double Var1x, Var1y, Var1z, Var1, Var2, Var3, Var4;
  int Num5;
  double Maxdispl, Maxdispl_x, Maxdispl_y, Maxdispl_z;

  double Maxdispl_x_max, Maxdispl_y_max, Maxdispl_z_max;
  double Maxdispl_x_min, Maxdispl_y_min, Maxdispl_z_min;


  Action1 = x.Branch(1); //number of nodes

  for (int i = 0; i < Action1.Count; i++)
  {

    for (int j = 0 ; j < loadcomb.Count; j++) //load cases that should be accounted for
    {
      Num5 = loadcomb[j];
      Action1 = x.Branch(Num5);//retrieve branch corresponding to loadcase Num5

      //one displacement (x,y,z) measured per node i
      Var1x = Action1[i].X; //displacement x-dir
      Var1y = Action1[i].Y; //displacement y-dir
      Var1z = Action1[i].Z; //displacement y-dir
```



```
        //Resulting displacement:
        Var4 = Math.Sqrt(Math.Pow(Var1x, 2) + Math.Pow(Var1y, 2) + Math.Pow(Var1z, 2));

        //Add all displacement in node i for all load cases in a list:
        VarList.Add(Var4);
        VarX.Add(Var1x);
        VarY.Add(Var1y);
        VarZ.Add(Var1z);
      }

      //sum the vaules in the list to figure out the total displacement of node i
      Var4 = VarList.Sum();
      Var1 = VarX.Sum();
      Var2 = VarY.Sum();
      Var3 = VarZ.Sum();
      Node_displ.Add(Var4);
      Node_displX.Add(Var1);
      Node_displY.Add(Var2);
      Node_displZ.Add(Var3);

      //clear list before calculating next node

      VarList.Clear();
      VarX.Clear();
      VarY.Clear();
      VarZ.Clear();


    }

    Maxdispl = Node_displ.Max();
    Maxdispl_x_max = Node_displX.Max();
    Maxdispl_y_max = Node_displY.Max();
    Maxdispl_z_max = Node_displZ.Max();
    Maxdispl_x_min = Node_displX.Min();
    Maxdispl_y_min = Node_displY.Min();
    Maxdispl_z_min = Node_displZ.Min();

    //check if max directional displacement is negativ or positive
    // and choose the greatest absolute vaule
    if (Math.Abs(Maxdispl_x_min) > Math.Abs(Maxdispl_x_max))
    {
      Maxdispl_x = Maxdispl_x_min;
    }
    else
    {
      Maxdispl_x = Maxdispl_x_max;
    }


    if (Math.Abs(Maxdispl_y_min) > Math.Abs(Maxdispl_y_max))
    {
      Maxdispl_y = Maxdispl_y_min;
    }
    else
    {
      Maxdispl_y = Maxdispl_y_max;
    }

    if (Math.Abs(Maxdispl_z_min) > Math.Abs(Maxdispl_z_max))
    {
      Maxdispl_z = Maxdispl_z_min;
    }
    else
    {
      Maxdispl_z = Maxdispl_z_max;
    }

    X_max = Maxdispl_x;
    Y_max = Maxdispl_y;
    Z_max = Maxdispl_z;
    tot_max = Maxdispl;


  }

  // <Custom additional code>
```

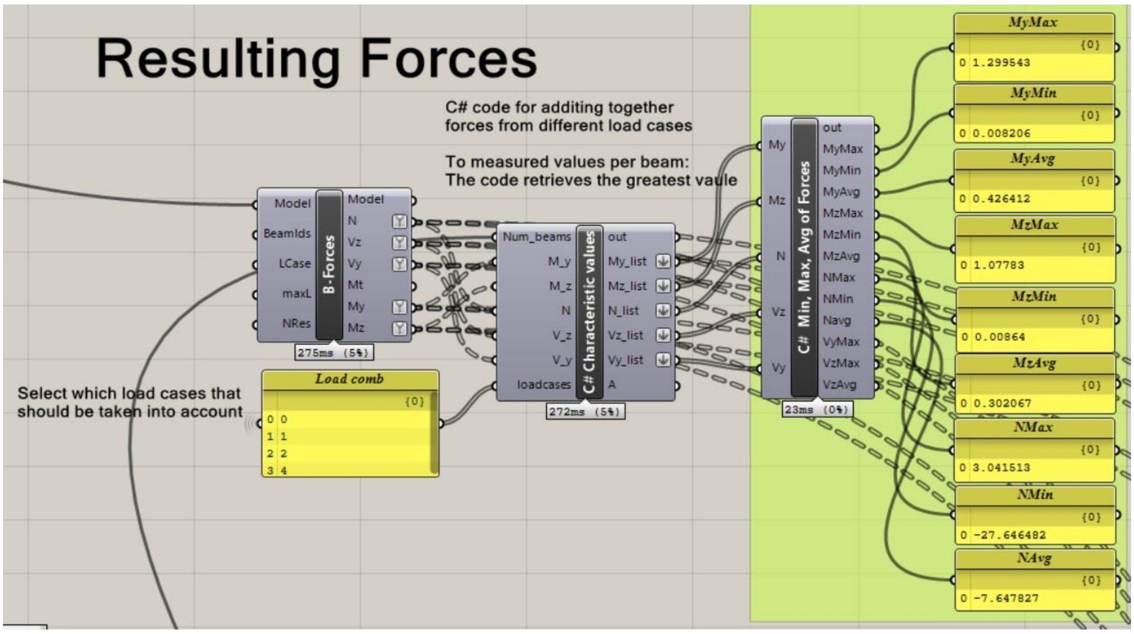
*Reaction Forces





```
private void RunScript(DataTree<double> x, double circum, List<int> loadcases, ref object Sup)
{
  List<double> List1 = new List<double> ();

  //This procedure is similar for Rx,Ry and Rz
  double tall1 = 0.0;
  double Var1;
  int j;
  //Get reaction forces Rx for load case 0
  List1 = x.Branch(0);

  for (int i = 0; i < List1.Count;i++)
  {
    for (int k = 0; k < loadcases.Count; k++)
    {
      j = loadcases[k];
      Var1 = Math.Abs(x.Branch(j)[i]);
      tall1 += Var1;
    }
  }

  //Calculate Reaction force  Rx_Tot in kN per
  //meter around the supporting edge(circumsference)
  Sup = tall1 / circum;
}
```

*Beam Forces



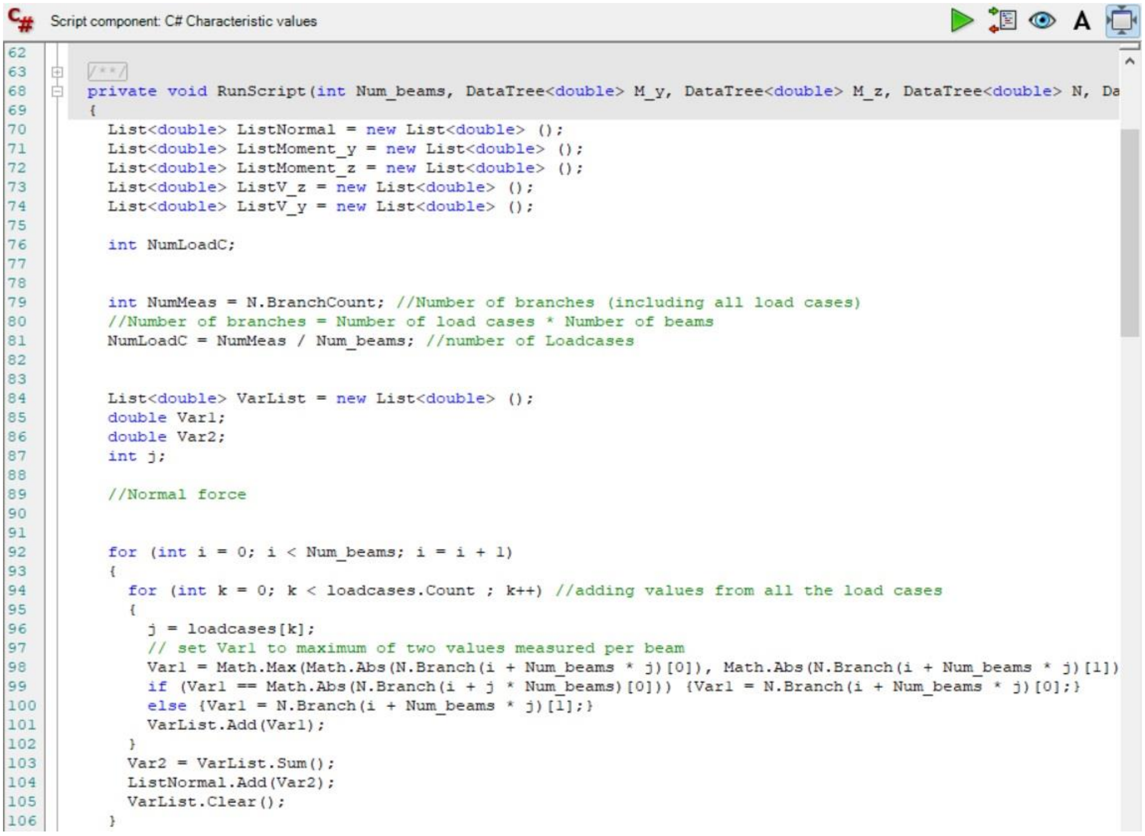
Resulting Forces

C# code for additing together forces from different load cases

To measured values per beam: The code retrieves the greatest vaule

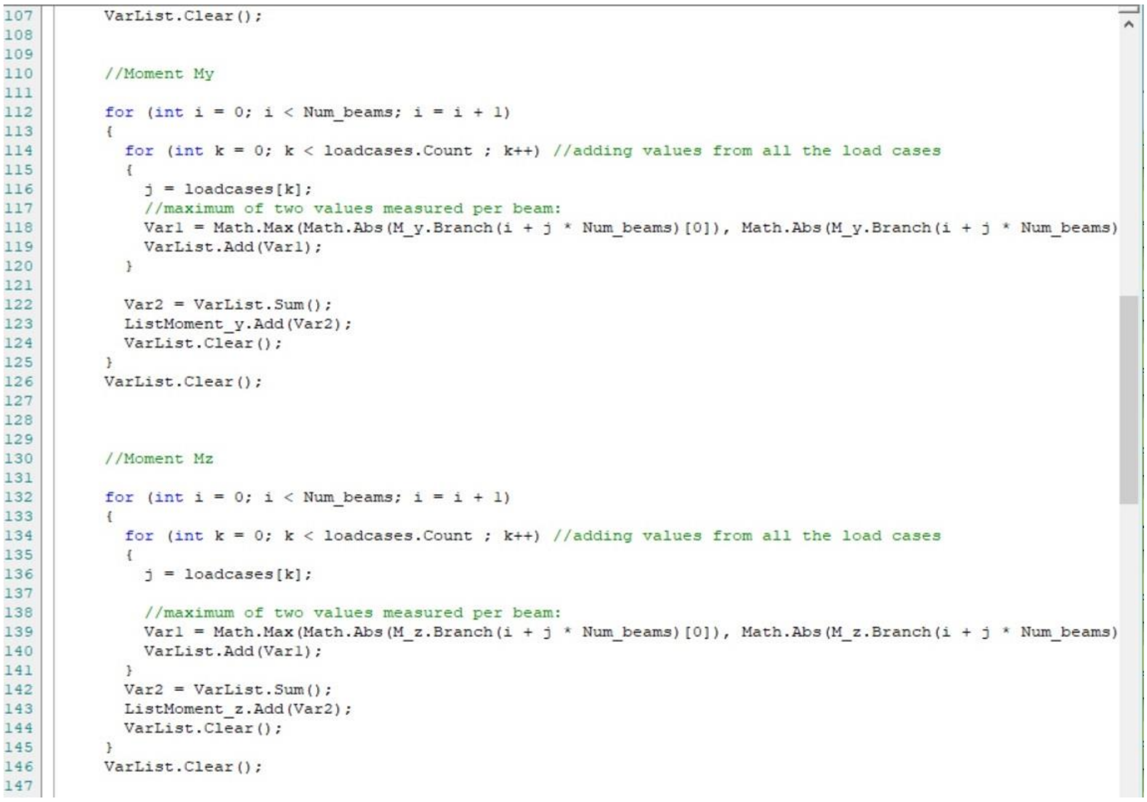
Select which load cases that should be taken into account

C# Characteristic values

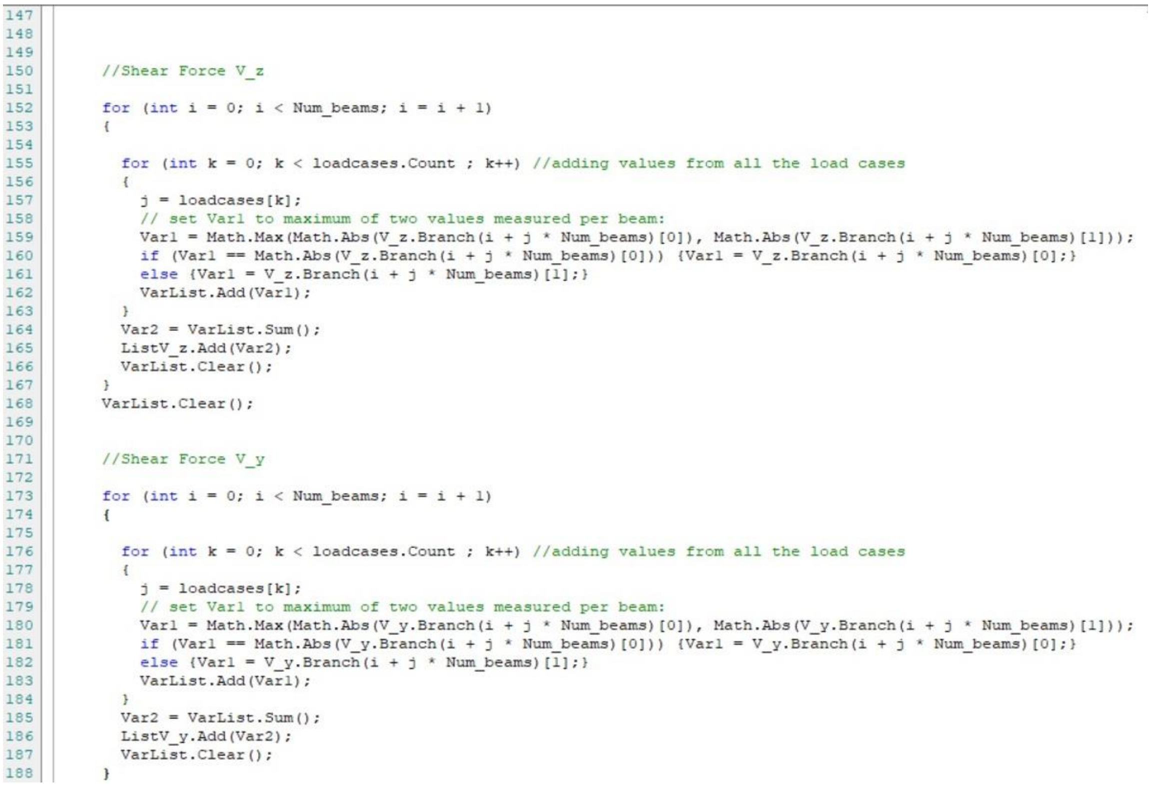


Script component: C# Characteristic values

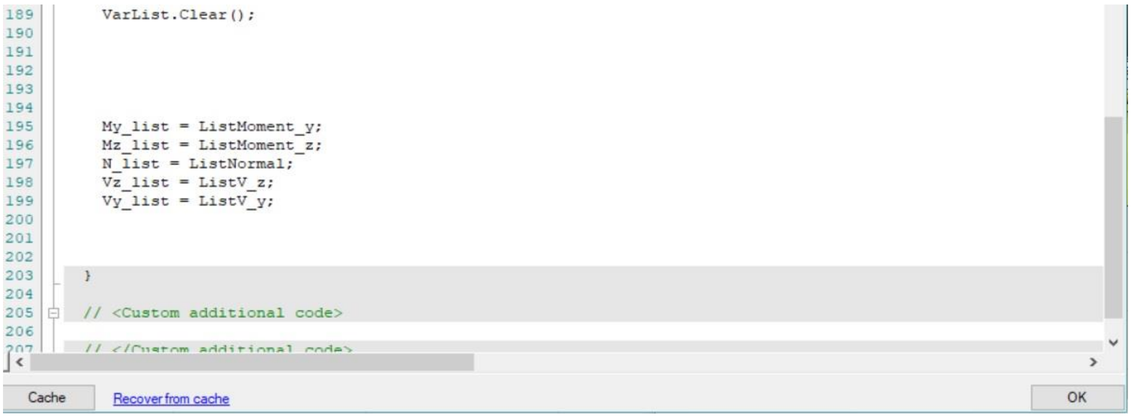
```
/**/
private void RunScript(int Num_beams, DataTree<double> M_y, DataTree<double> M_z, DataTree<double> N, Da
{
  List<double> ListNormal = new List<double> ();
  List<double> ListMoment_y = new List<double> ();
  List<double> ListMoment_z = new List<double> ();
  List<double> ListV_z = new List<double> ();
  List<double> ListV_y = new List<double> ();

  int NumLoadC;


  int NumMeas = N.BranchCount; //Number of branches (including all load cases)
  //Number of branches = Number of load cases * Number of beams
  NumLoadC = NumMeas / Num_beams; //number of Loadcases


  List<double> VarList = new List<double> ();
  double Var1;
  double Var2;
  int j;

  //Normal force


  for (int i = 0; i < Num_beams; i = i + 1)
  {
    for (int k = 0; k < loadcases.Count ; k++) //adding values from all the load cases
    {
      j = loadcases[k];
      // set Var1 to maximum of two values measured per beam
      Var1 = Math.Max(Math.Abs(N.Branch(i + Num_beams * j)[0]), Math.Abs(N.Branch(i + Num_beams * j)[1])
      if (Var1 == Math.Abs(N.Branch(i + j * Num_beams)[0])) {Var1 = N.Branch(i + Num_beams * j)[0];}
      else {Var1 = N.Branch(i + Num_beams * j)[1];}
      VarList.Add(Var1);
    }
    Var2 = VarList.Sum();
    ListNormal.Add(Var2);
    VarList.Clear();
  }
```



```
    VarList.Clear();


    //Moment My

    for (int i = 0; i < Num_beams; i = i + 1)
    {
      for (int k = 0; k < loadcases.Count ; k++) //adding values from all the load cases
      {
        j = loadcases[k];
        //maximum of two values measured per beam:
        Var1 = Math.Max(Math.Abs(M_y.Branch(i + j * Num_beams)[0]), Math.Abs(M_y.Branch(i + j * Num_beams)
        VarList.Add(Var1);
      }

      Var2 = VarList.Sum();
      ListMoment_y.Add(Var2);
      VarList.Clear();
    }
    VarList.Clear();



    //Moment Mz

    for (int i = 0; i < Num_beams; i = i + 1)
    {
      for (int k = 0; k < loadcases.Count ; k++) //adding values from all the load cases
      {
        j = loadcases[k];

        //maximum of two values measured per beam:
        Var1 = Math.Max(Math.Abs(M_z.Branch(i + j * Num_beams)[0]), Math.Abs(M_z.Branch(i + j * Num_beams)
        VarList.Add(Var1);
      }
      Var2 = VarList.Sum();
      ListMoment_z.Add(Var2);
      VarList.Clear();
    }
    VarList.Clear();

```

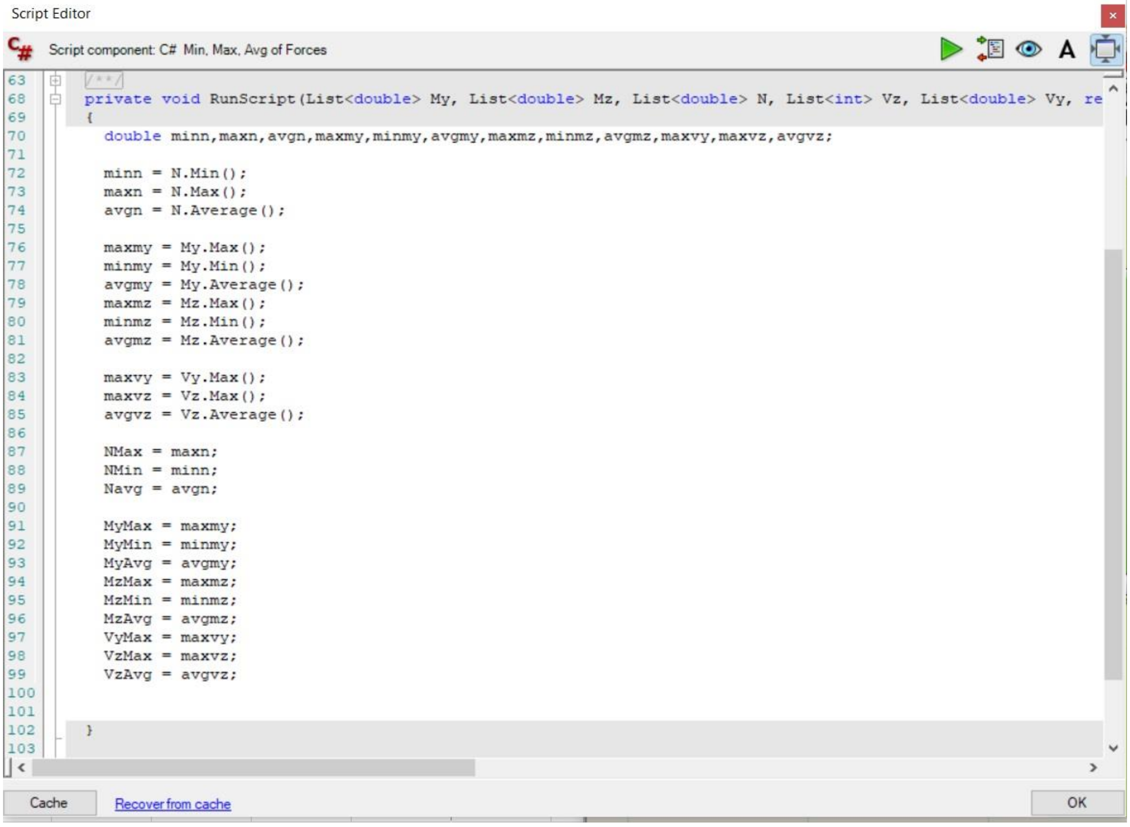


```



    //Shear Force V_z

    for (int i = 0; i < Num_beams; i = i + 1)
    {

      for (int k = 0; k < loadcases.Count ; k++) //adding values from all the load cases
      {
        j = loadcases[k];
        // set Var1 to maximum of two values measured per beam:
        Var1 = Math.Max(Math.Abs(V_z.Branch(i + j * Num_beams)[0]), Math.Abs(V_z.Branch(i + j * Num_beams)[1]));
        if (Var1 == Math.Abs(V_z.Branch(i + j * Num_beams)[0])) {Var1 = V_z.Branch(i + j * Num_beams)[0];}
        else {Var1 = V_z.Branch(i + j * Num_beams)[1];}
        VarList.Add(Var1);
      }
      Var2 = VarList.Sum();
      ListV_z.Add(Var2);
      VarList.Clear();
    }
    VarList.Clear();


    //Shear Force V_y

    for (int i = 0; i < Num_beams; i = i + 1)
    {

      for (int k = 0; k < loadcases.Count ; k++) //adding values from all the load cases
      {
        j = loadcases[k];
        // set Var1 to maximum of two values measured per beam:
        Var1 = Math.Max(Math.Abs(V_y.Branch(i + j * Num_beams)[0]), Math.Abs(V_y.Branch(i + j * Num_beams)[1]));
        if (Var1 == Math.Abs(V_y.Branch(i + j * Num_beams)[0])) {Var1 = V_y.Branch(i + j * Num_beams)[0];}
        else {Var1 = V_y.Branch(i + j * Num_beams)[1];}
        VarList.Add(Var1);
      }
      Var2 = VarList.Sum();
      ListV_y.Add(Var2);
      VarList.Clear();
    }
```



```
    VarList.Clear();




    My_list = ListMoment_y;
    Mz_list = ListMoment_z;
    N_list = ListNormal;
    Vz_list = ListV_z;
    Vy_list = ListV_y;



  }

  // <Custom additional code>

  // </Custom additional code>
```

C # Min, Max, Avg of Forces



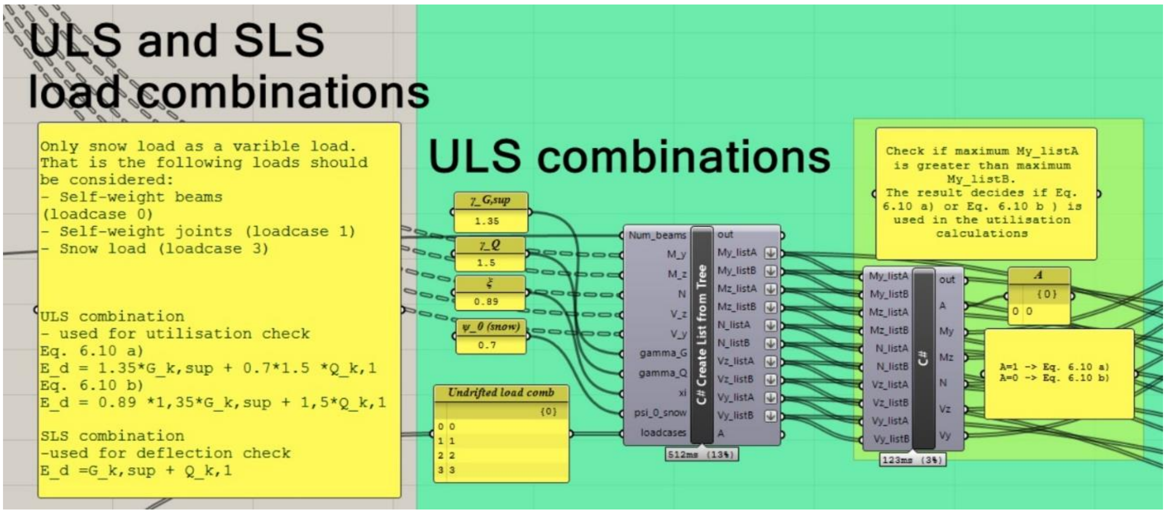
Script Editor

Script component: C# Min, Max, Avg of Forces

```
/**/
private void RunScript(List<double> My, List<double> Mz, List<double> N, List<int> Vz, List<double> Vy, re
{
  double minn,maxn,avgn,maxmy,minmy,avgmy,maxmz,minmz,avgmz,maxvy,maxvz,avgvz;

  minn = N.Min();
  maxn = N.Max();
  avgn = N.Average();

  maxmy = My.Max();
  minmy = My.Min();
  avgmy = My.Average();
  maxmz = Mz.Max();
  minmz = Mz.Min();
  avgmz = Mz.Average();

  maxvy = Vy.Max();
  maxvz = Vz.Max();
  avgvz = Vz.Average();

  NMax = maxn;
  NMin = minn;
  Navg = avgn;

  MyMax = maxmy;
  MyMin = minmy;
  MyAvg = avgmy;
  MzMax = maxmz;
  MzMin = minmz;
  MzAvg = avgmz;
  VyMax = maxvy;
  VzMax = maxvz;
  VzAvg = avgvz;


}
```

Processing and optimising of results

*Design load combinations



Only snow load as a varible load.
That is the following loads should be considered:
- Self-weight beams (loadcase 0)
- Self-weight joints (loadcase 1)
- Snow load (loadcase 3)

ULS combination
- used for utilisation check
Eq. 6.10 a)
E_d = 1.35*G_k,sup + 0.7*1.5 *Q_k,1
Eq. 6.10 b)
E_d = 0.89 *1,35*G_k,sup + 1,5*Q_k,1

SLS combination
-used for deflection check
E_d =G_k,sup + Q_k,1

γ_G,sup 1.35

γ_Q 1.5

ξ 0.89

ψ_0 (snow) 0.7

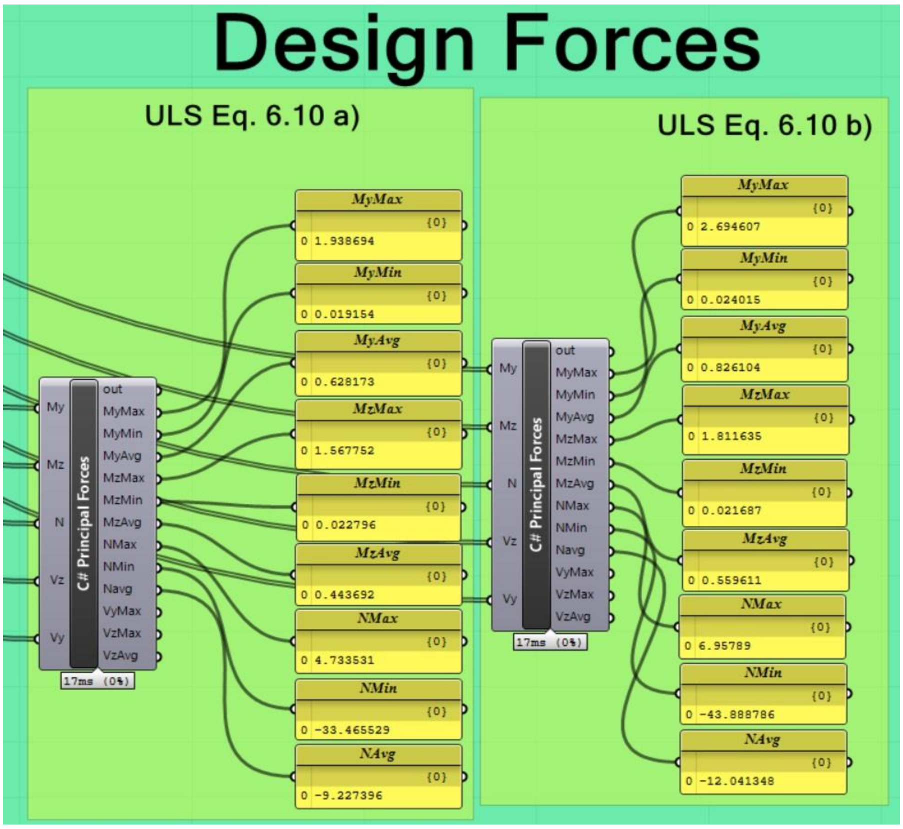
Undrifted load comb {0}
0 0
1 1
2 2
3 3

Check if maximum My_listA is greater than maximum My_listB.
The result decides if Eq. 6.10 a) or Eq. 6.10 b) is used in the utilisation calculations

A {0}
0 0

A=1 -> Eq. 6.10 a)
A=0 -> Eq. 6.10 b)

*Design beam forces



ULS Eq. 6.10 a)

MyMax {0}
0 1.938694

MyMin {0}
0 0.019154

MyAvg {0}
0 0.628173

MzMax {0}
0 1.567752

MzMin {0}
0 0.022796

MzAvg {0}
0 0.443692

NMax {0}
0 4.733531

NMin {0}
0 -33.465529

NAvg {0}
0 -9.227396

ULS Eq. 6.10 b)

MyMax {0}
0 2.694607

MyMin {0}
0 0.024015

MyAvg {0}
0 0.826104

MzMax {0}
0 1.811635

MzMin {0}
0 0.021687

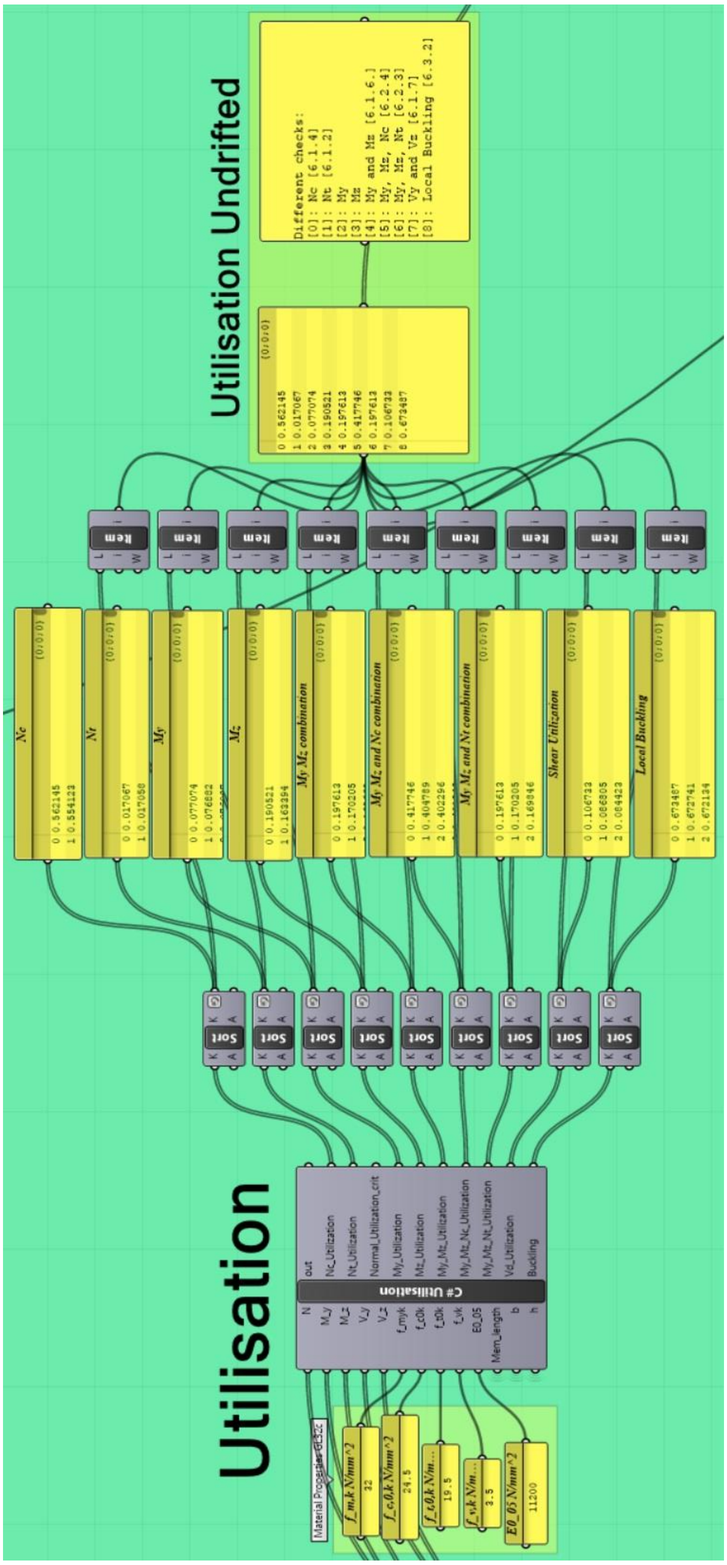
MzAvg {0}
0 0.559611

NMax {0}
0 6.95789

NMin {0}
0 -43.888786

NAvg {0}
0 -12.041348

*Utilisation of timber beam elements according to Eurocode 5



```
Nc
{0;0;0}
0 0.562145
1 0.554123
```

```
Nt
{0;0;0}
0 0.017067
1 0.017058
```

```
My
{0;0;0}
0 0.077074
1 0.076882
```

```
Mz
{0;0;0}
0 0.190521
1 0.163394
```

```
My Mz combination
{0;0;0}
0 0.197613
1 0.170205
```

```
My Mz and Nc combination
{0;0;0}
0 0.417746
1 0.404789
2 0.402296
```

```
My Mz and Nt combination
{0;0;0}
0 0.197613
1 0.170205
2 0.169846
```

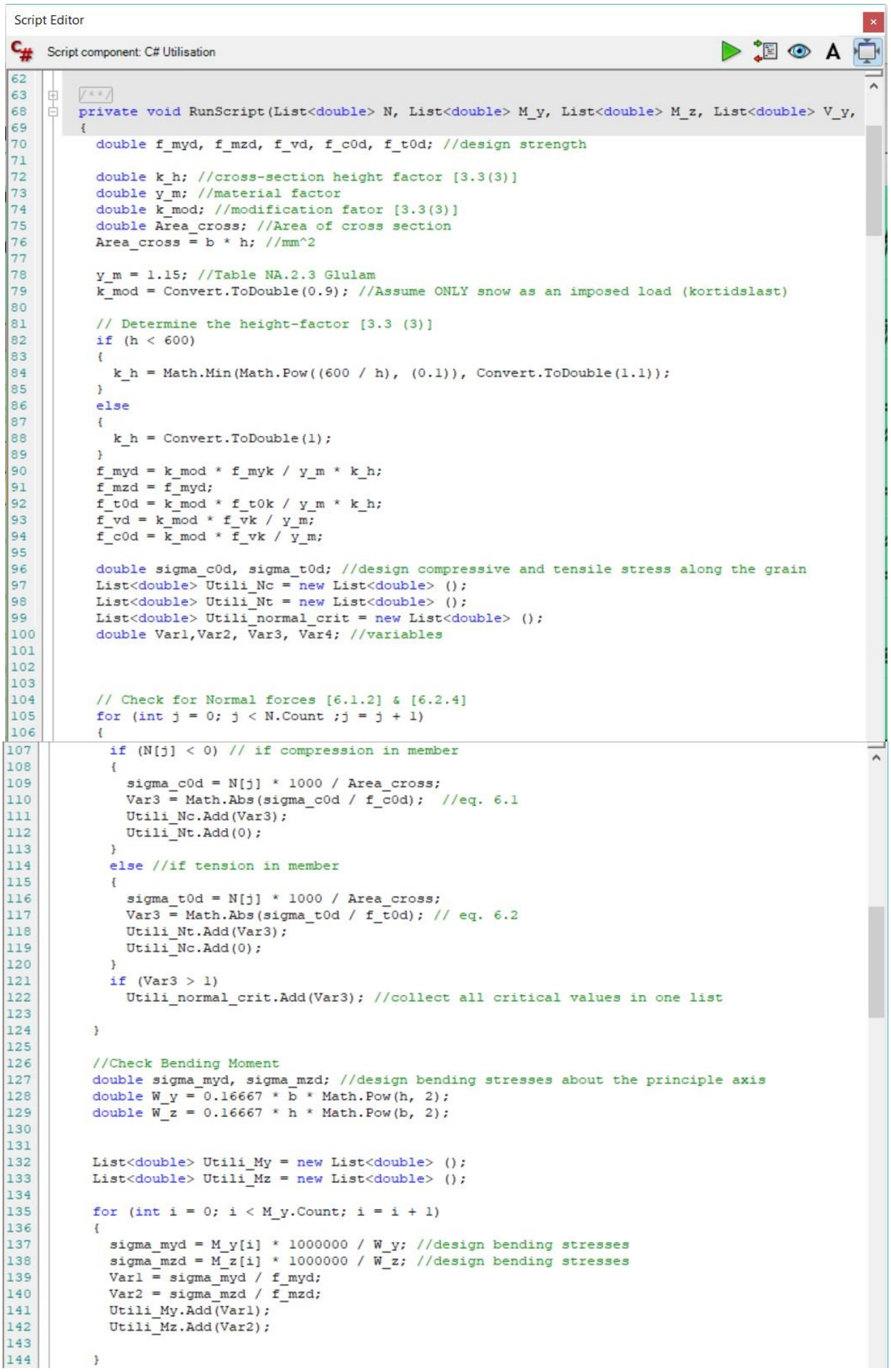
```
Shear Utilization
{0;0;0}
0 0.106733
1 0.086805
2 0.084423
```

```
Local Buckling
{0;0;0}
0 0.673487
1 0.672741
2 0.672134
```

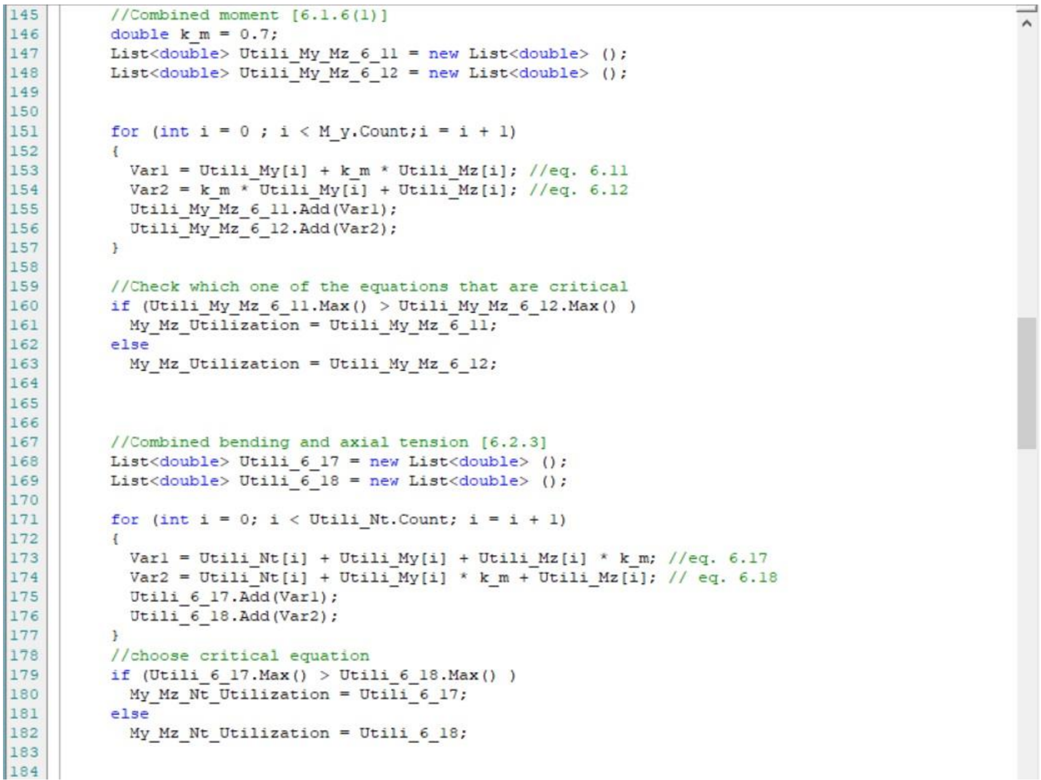
```
{0;0;0}
0 0.562145
1 0.017067
2 0.077074
3 0.190521
4 0.197613
5 0.417746
6 0.197613
7 0.106733
8 0.673487
```

```
Different checks:
[0]: Nc [6.1.4]
[1]: Nt [6.1.2]
[2]: My
[3]: Mz
[4]: My and Mz [6.1.6.]
[5]: My, Mz, Nc [6.2.4]
[6]: My, Mz, Nt [6.2.3]
[7]: Vy and Vz [6.1.7]
[8]: Local Buckling [6.3.2]
```

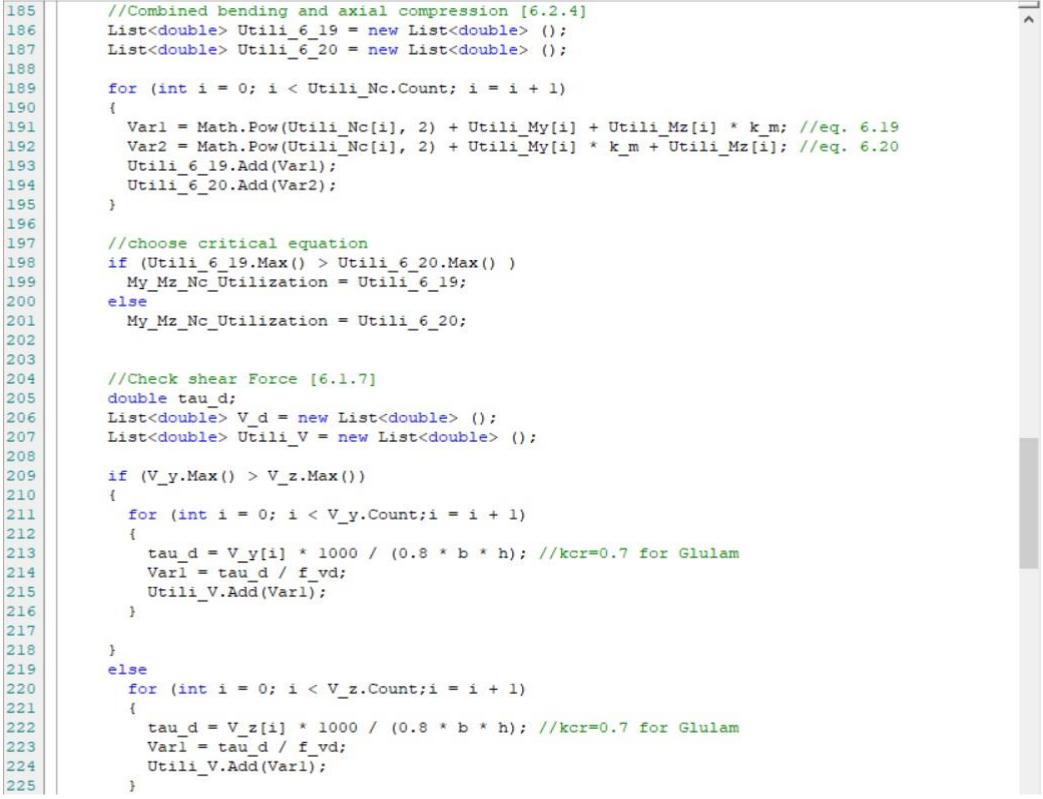
C# Utilisation code



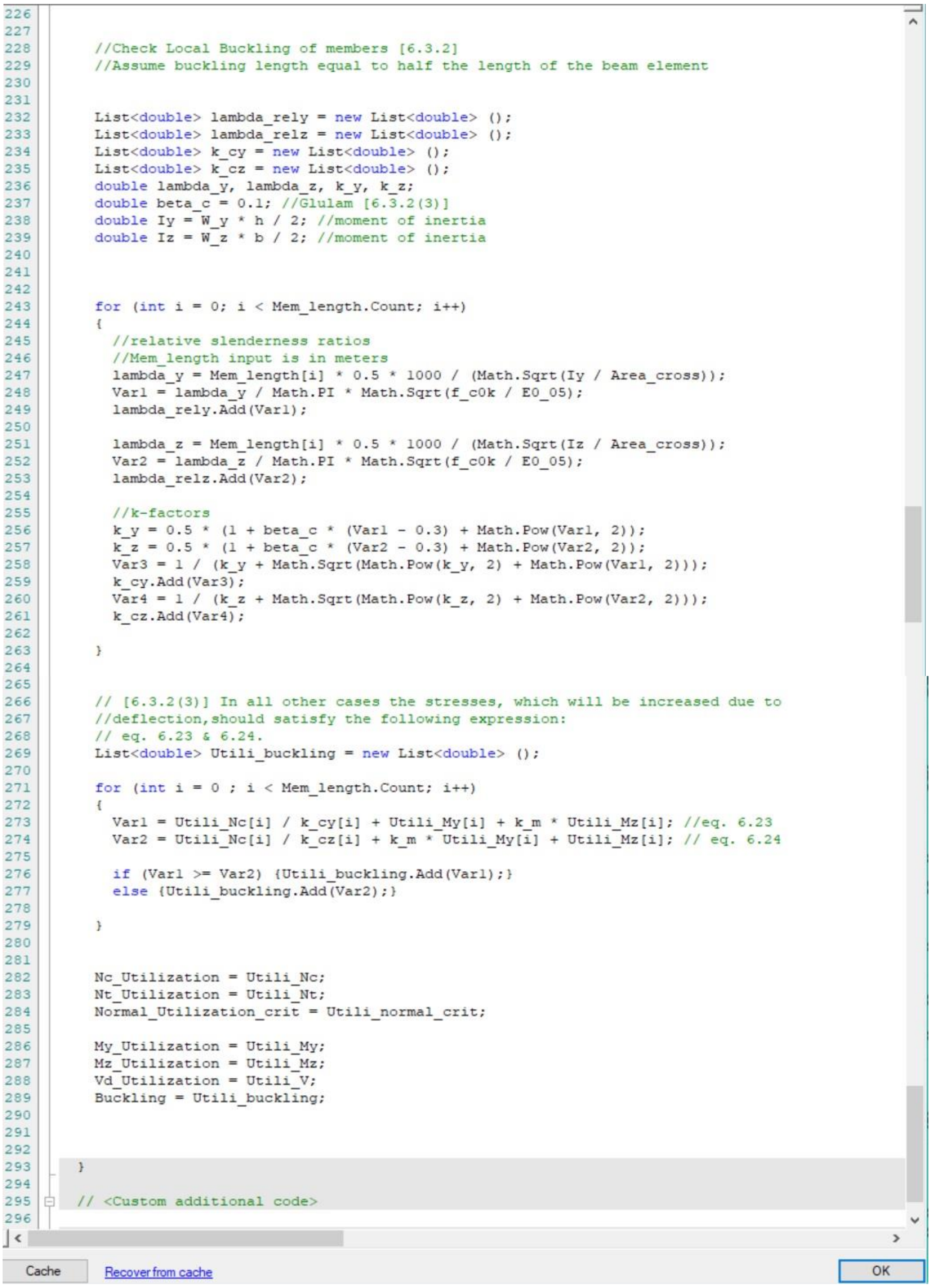
```
/**/
private void RunScript(List<double> N, List<double> M_y, List<double> M_z, List<double> V_y,
{
  double f_myd, f_mzd, f_vd, f_c0d, f_t0d; //design strength

  double k_h; //cross-section height factor [3.3(3)]
  double y_m; //material factor
  double k_mod; //modification fator [3.3(3)]
  double Area_cross; //Area of cross section
  Area_cross = b * h; //mm^2

  y_m = 1.15; //Table NA.2.3 Glulam
  k_mod = Convert.ToDouble(0.9); //Assume ONLY snow as an imposed load (kortidslast)

  // Determine the height-factor [3.3 (3)]
  if (h < 600)
  {
    k_h = Math.Min(Math.Pow((600 / h), (0.1)), Convert.ToDouble(1.1));
  }
  else
  {
    k_h = Convert.ToDouble(1);
  }
  f_myd = k_mod * f_myk / y_m * k_h;
  f_mzd = f_myd;
  f_t0d = k_mod * f_t0k / y_m * k_h;
  f_vd = k_mod * f_vk / y_m;
  f_c0d = k_mod * f_vk / y_m;

  double sigma_c0d, sigma_t0d; //design compressive and tensile stress along the grain
  List<double> Utili_Nc = new List<double> ();
  List<double> Utili_Nt = new List<double> ();
  List<double> Utili_normal_crit = new List<double> ();
  double Var1,Var2, Var3, Var4; //variables



  // Check for Normal forces [6.1.2] & [6.2.4]
  for (int j = 0; j < N.Count ;j = j + 1)
  {
    if (N[j] < 0) // if compression in member
    {
      sigma_c0d = N[j] * 1000 / Area_cross;
      Var3 = Math.Abs(sigma_c0d / f_c0d);  //eq. 6.1
      Utili_Nc.Add(Var3);
      Utili_Nt.Add(0);
    }
    else //if tension in member
    {
      sigma_t0d = N[j] * 1000 / Area_cross;
      Var3 = Math.Abs(sigma_t0d / f_t0d); // eq. 6.2
      Utili_Nt.Add(Var3);
      Utili_Nc.Add(0);
    }
    if (Var3 > 1)
      Utili_normal_crit.Add(Var3); //collect all critical values in one list

  }

  //Check Bending Moment
  double sigma_myd, sigma_mzd; //design bending stresses about the principle axis
  double W_y = 0.16667 * b * Math.Pow(h, 2);
  double W_z = 0.16667 * h * Math.Pow(b, 2);


  List<double> Utili_My = new List<double> ();
  List<double> Utili_Mz = new List<double> ();

  for (int i = 0; i < M_y.Count; i = i + 1)
  {
    sigma_myd = M_y[i] * 1000000 / W_y; //design bending stresses
    sigma_mzd = M_z[i] * 1000000 / W_z; //design bending stresses
    Var1 = sigma_myd / f_myd;
    Var2 = sigma_mzd / f_mzd;
    Utili_My.Add(Var1);
    Utili_Mz.Add(Var2);

  }
```



```
//Combined moment [6.1.6(1)]
double k_m = 0.7;
List<double> Utili_My_Mz_6_11 = new List<double> ();
List<double> Utili_My_Mz_6_12 = new List<double> ();


for (int i = 0 ; i < M_y.Count;i = i + 1)
{
  Var1 = Utili_My[i] + k_m * Utili_Mz[i]; //eq. 6.11
  Var2 = k_m * Utili_My[i] + Utili_Mz[i]; //eq. 6.12
  Utili_My_Mz_6_11.Add(Var1);
  Utili_My_Mz_6_12.Add(Var2);
}

//Check which one of the equations that are critical
if (Utili_My_Mz_6_11.Max() > Utili_My_Mz_6_12.Max() )
  My_Mz_Utilization = Utili_My_Mz_6_11;
else
  My_Mz_Utilization = Utili_My_Mz_6_12;



//Combined bending and axial tension [6.2.3]
List<double> Utili_6_17 = new List<double> ();
List<double> Utili_6_18 = new List<double> ();

for (int i = 0; i < Utili_Nt.Count; i = i + 1)
{
  Var1 = Utili_Nt[i] + Utili_My[i] + Utili_Mz[i] * k_m; //eq. 6.17
  Var2 = Utili_Nt[i] + Utili_My[i] * k_m + Utili_Mz[i]; // eq. 6.18
  Utili_6_17.Add(Var1);
  Utili_6_18.Add(Var2);
}
//choose critical equation
if (Utili_6_17.Max() > Utili_6_18.Max() )
  My_Mz_Nt_Utilization = Utili_6_17;
else
  My_Mz_Nt_Utilization = Utili_6_18;

```

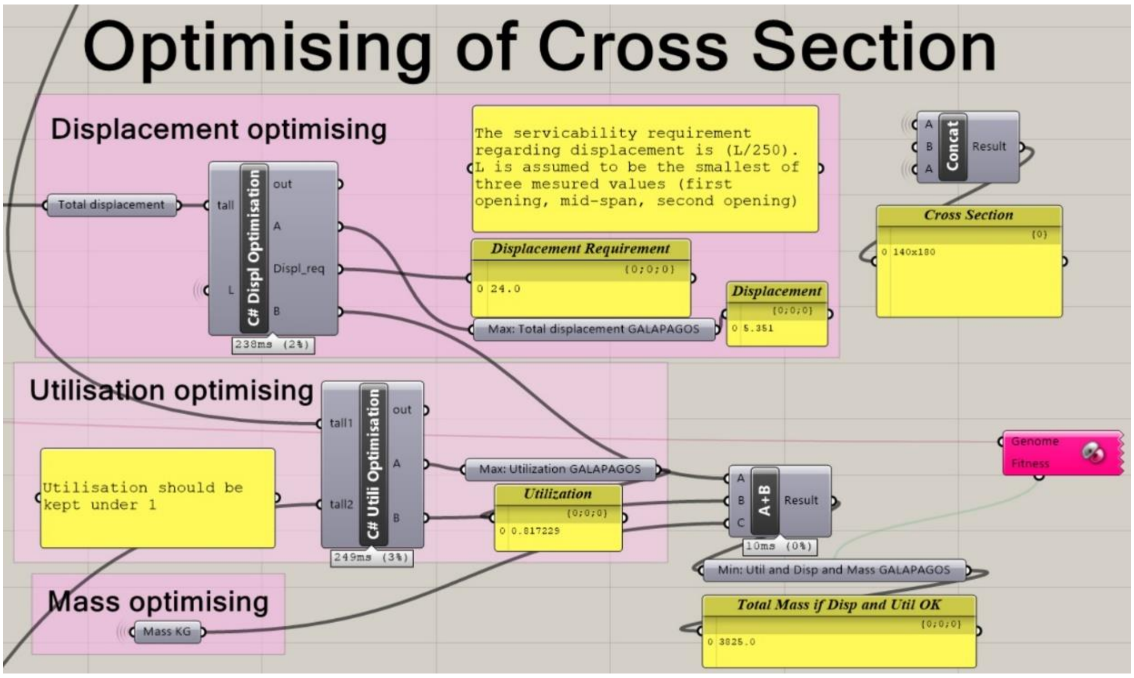


```
//Combined bending and axial compression [6.2.4]
List<double> Utili_6_19 = new List<double> ();
List<double> Utili_6_20 = new List<double> ();

for (int i = 0; i < Utili_Nc.Count; i = i + 1)
{
  Var1 = Math.Pow(Utili_Nc[i], 2) + Utili_My[i] + Utili_Mz[i] * k_m; //eq. 6.19
  Var2 = Math.Pow(Utili_Nc[i], 2) + Utili_My[i] * k_m + Utili_Mz[i]; //eq. 6.20
  Utili_6_19.Add(Var1);
  Utili_6_20.Add(Var2);
}

//choose critical equation
if (Utili_6_19.Max() > Utili_6_20.Max() )
  My_Mz_Nc_Utilization = Utili_6_19;
else
  My_Mz_Nc_Utilization = Utili_6_20;


//Check shear Force [6.1.7]
double tau_d;
List<double> V_d = new List<double> ();
List<double> Utili_V = new List<double> ();

if (V_y.Max() > V_z.Max())
{
  for (int i = 0; i < V_y.Count;i = i + 1)
  {
    tau_d = V_y[i] * 1000 / (0.8 * b * h); //kcr=0.7 for Glulam
    Var1 = tau_d / f_vd;
    Utili_V.Add(Var1);
  }

}
else
  for (int i = 0; i < V_z.Count;i = i + 1)
  {
    tau_d = V_z[i] * 1000 / (0.8 * b * h); //kcr=0.7 for Glulam
    Var1 = tau_d / f_vd;
    Utili_V.Add(Var1);
  }
```



```


    //Check Local Buckling of members [6.3.2]
    //Assume buckling length equal to half the length of the beam element


    List<double> lambda_rely = new List<double> ();
    List<double> lambda_relz = new List<double> ();
    List<double> k_cy = new List<double> ();
    List<double> k_cz = new List<double> ();
    double lambda_y, lambda_z, k_y, k_z;
    double beta_c = 0.1; //Glulam [6.3.2(3)]
    double Iy = W_y * h / 2; //moment of inertia
    double Iz = W_z * b / 2; //moment of inertia



    for (int i = 0; i < Mem_length.Count; i++)
    {
      //relative slenderness ratios
      //Mem_length input is in meters
      lambda_y = Mem_length[i] * 0.5 * 1000 / (Math.Sqrt(Iy / Area_cross));
      Var1 = lambda_y / Math.PI * Math.Sqrt(f_c0k / E0_05);
      lambda_rely.Add(Var1);

      lambda_z = Mem_length[i] * 0.5 * 1000 / (Math.Sqrt(Iz / Area_cross));
      Var2 = lambda_z / Math.PI * Math.Sqrt(f_c0k / E0_05);
      lambda_relz.Add(Var2);

      //k-factors
      k_y = 0.5 * (1 + beta_c * (Var1 - 0.3) + Math.Pow(Var1, 2));
      k_z = 0.5 * (1 + beta_c * (Var2 - 0.3) + Math.Pow(Var2, 2));
      Var3 = 1 / (k_y + Math.Sqrt(Math.Pow(k_y, 2) + Math.Pow(Var1, 2)));
      k_cy.Add(Var3);
      Var4 = 1 / (k_z + Math.Sqrt(Math.Pow(k_z, 2) + Math.Pow(Var2, 2)));
      k_cz.Add(Var4);

    }


    // [6.3.2(3)] In all other cases the stresses, which will be increased due to
    //deflection,should satisfy the following expression:
    // eq. 6.23 & 6.24.
    List<double> Utili_buckling = new List<double> ();

    for (int i = 0 ; i < Mem_length.Count; i++)
    {
      Var1 = Utili_Nc[i] / k_cy[i] + Utili_My[i] + k_m * Utili_Mz[i]; //eq. 6.23
      Var2 = Utili_Nc[i] / k_cz[i] + k_m * Utili_My[i] + Utili_Mz[i]; // eq. 6.24

      if (Var1 >= Var2) {Utili_buckling.Add(Var1);}
      else {Utili_buckling.Add(Var2);}

    }


    Nc_Utilization = Utili_Nc;
    Nt_Utilization = Utili_Nt;
    Normal_Utilization_crit = Utili_normal_crit;

    My_Utilization = Utili_My;
    Mz_Utilization = Utili_Mz;
    Vd_Utilization = Utili_V;
    Buckling = Utili_buckling;



  }

  // <Custom additional code>

```

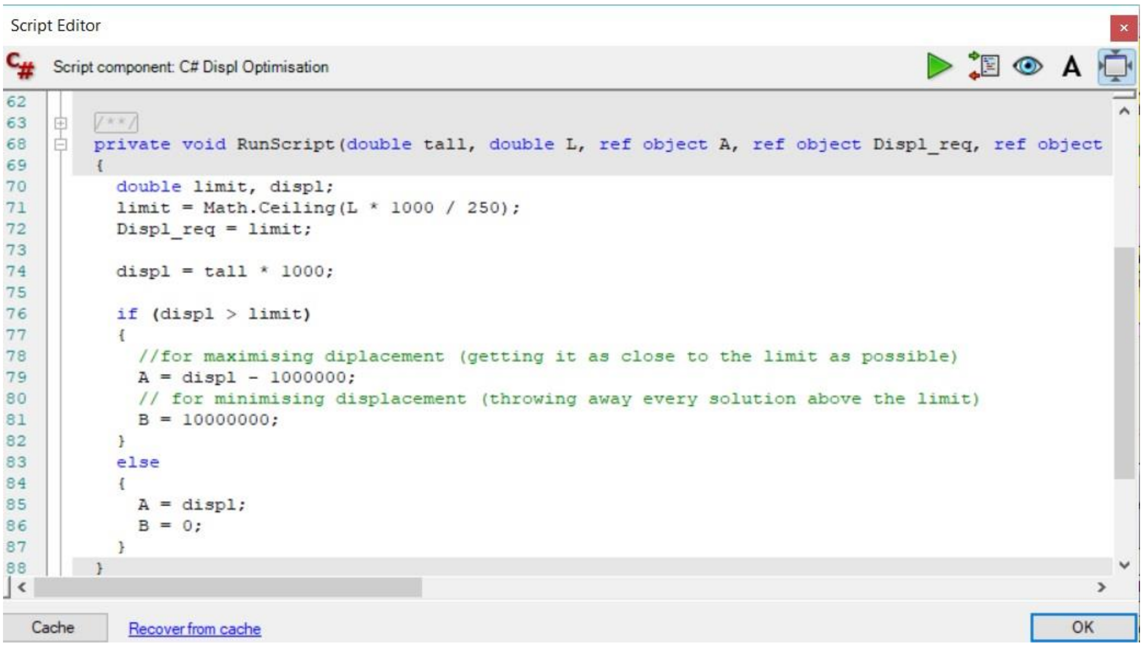
*Optimising of cross section



The servicability requirement regarding displacement is (L/250). L is assumed to be the smallest of three mesured values (first opening, mid-span, second opening)

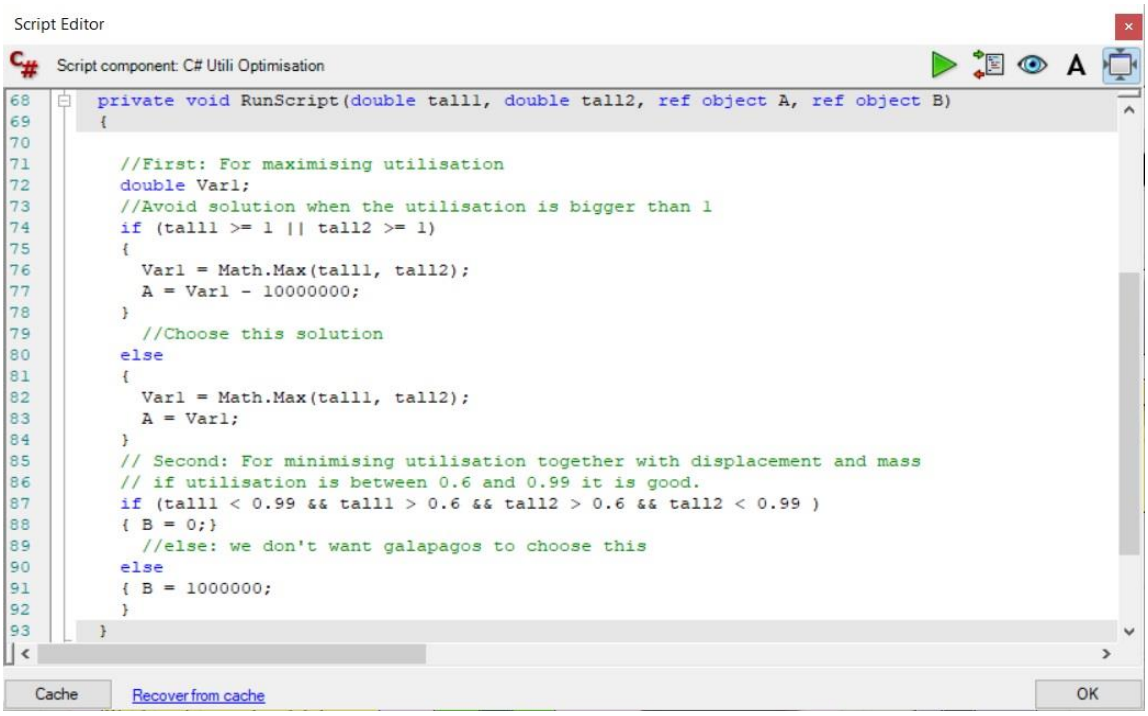
Utilisation should be kept under 1

C# Displacement Optimisation



```
/**/
private void RunScript(double tall, double L, ref object A, ref object Displ_req, ref object
{
  double limit, displ;
  limit = Math.Ceiling(L * 1000 / 250);
  Displ_req = limit;

  displ = tall * 1000;

  if (displ > limit)
  {
    //for maximising diplacement (getting it as close to the limit as possible)
    A = displ - 1000000;
    // for minimising displacement (throwing away every solution above the limit)
    B = 10000000;
  }
  else
  {
    A = displ;
    B = 0;
  }
}
```

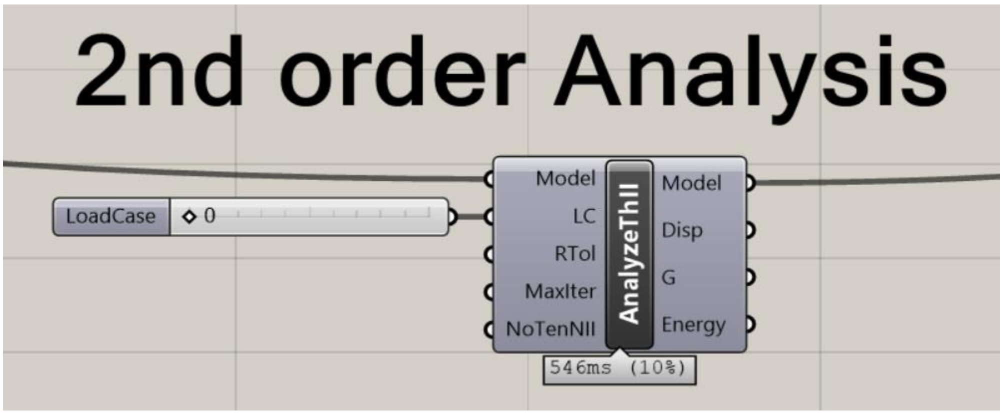
C# Utilisation Optimisation



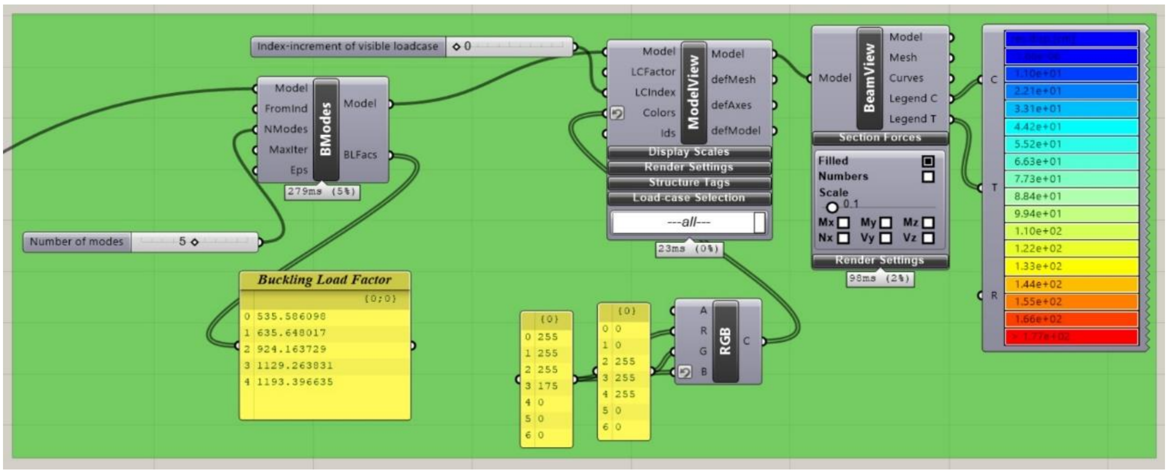
```
private void RunScript(double tall1, double tall2, ref object A, ref object B)
{

  //First: For maximising utilisation
  double Var1;
  //Avoid solution when the utilisation is bigger than 1
  if (tall1 >= 1 || tall2 >= 1)
  {
    Var1 = Math.Max(tall1, tall2);
    A = Var1 - 10000000;
  }
    //Choose this solution
  else
  {
    Var1 = Math.Max(tall1, tall2);
    A = Var1;
  }
  // Second: For minimising utilisation together with displacement and mass
  // if utilisation is between 0.6 and 0.99 it is good.
  if (tall1 < 0.99 && tall1 > 0.6 && tall2 > 0.6 && tall2 < 0.99 )
  { B = 0;}
    //else: we don't want galapagos to choose this
  else
  { B = 1000000;
  }
}
```

PART 5

*2nd order analysis



*Calculation of Buckling Modes



Index-increment of visible loadcase ◇ 0

Number of modes 5 ◇

Buckling Load Factor

	{0;0}
0	535.586098
1	635.648017
2	924.163729
3	1129.263831
4	1193.396635

	{0}
0	255
1	255
2	255
3	175
4	0
5	0
6	0

	{0}
0	0
1	0
2	255
3	255
4	255
5	0
6	0

Filled ▣
Numbers □
Scale ○ 0.1
Mx □ My □ Mz □
Nx □ Vy □ Vz □

D | Utilisation of timber elements

Cross Section			
	h	80	mm
	b	80	mm
	Area	6400	mm^2
	W_y	85333	mm^3
	W_z	85333	mm^3
	I_y	3413333	mm^4
	I_z	3413333	mm^4
	Lenght	0,897	m

Input values: (Arbitrary values are taken as inputs)

Forces	N_t	1	kN
	N_c	1	kN
	M_y	0,77	kNm
	M_z	2,78	kNm
	V_y	0,92	kN
	V_z	-6,06	kN

Material properties

Glued Laminated Timber (GL)

GL32c

Table 4 [NS-EN 14080]	f_mk	32	MPa
	f_c0k	24,5	MPa
	f_t0k	19,5	MPa
	f_vk	3,5	MPa
	E_0,05	11800	MPa

Table NA. 2.3

Partial factor for material property

GL ϒm 1,15

Table NA.901

Climate conditions:

Assume: load-bearing beam element that usually are not heated, but ventilated

Hence; Service Class 2

Table 3.1

Modification Factor, k_mod:
include the duration of the load and moisture content

Load combination:	k_mod
Selfweigt + Snow + Wind: Instantaneous action	1,10
Selfweigt + Snow : Short term action	0,9

In this calculation sheet utilisation calculations are presented for both load combinations.
This is to illustrate how k_mod effects the result.
However, it should be noted that the internal forces will not be the same for both situations.
This is due to the snow load comb. giving a different design load compared to wind load comb.

In Karamba only the load combination involving snow load will be considered, and this calculation sheet is simply presented to give the procedure of how the calculations are done with Eurocode 5.
Note: The values of the internal forces in this calculation sheet are arbitrary.
However by inputting the correct values corresponding to a beam element in the model,
the excel code can be used to double check the utlisation calculations in Karamba.

2.4.1(1)

The design value of strength properties

$$X_d = k_{mod} * \frac{X_k}{\gamma_M}$$

3.3 (3)

If the height, h, of the rectangular cross section is below 600 mm
f_m,k and f_t0k may be increased with the factor k_h

$$k_h = min\left\{\begin{matrix}\left(\frac{600}{h}\right)^{0.1} \\ 1.1\end{matrix}\right\}$$

k_h 1,100

Design value of material strength

	SnowLoad	*WindLoad*	
f_md	27,55	33,67	MPa
f_c0d	19,2	23,43	MPa
f_t0d	16,8	20,52	MPa
f_vd	2,7	3,35	MPa

6.1.2(1) **Tension parallell to grain direction**

$$\sigma_{t,0,d} \leq f_{t,0,d,}$$

$$\sigma_{t,0,d} = \frac{N_t}{A}$$

σ_t0d 0,15625 MPa

	Snow	*Wind*
Utilization Nt	0,01	0,01

6.1.4(1) **Compression parallell to grain direction**

$$\sigma_{c,0,d} \leq f_{c,0,d,}$$

$$\sigma_{c,0,d} = \frac{N_c}{A}$$

σ_c0d 0,15625 MPa

	Snow	Wind
Utilization Nc	0,01	0,01

6.1.6(1) **Bending**

Eq. 6.11

$$\frac{\sigma_{m,y,d}}{f_{m,y,d}} + k_m \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1$$

Eq. 6.12

$$k_m \frac{\sigma_{m,y,d}}{f_{m,y,d}} + \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1$$

$$\sigma_{m,d} = \frac{M}{W}$$

6.1.6(2) k_m 0,7 (rectangular cross section)

σ_myd 9,051550781

	Snow	*Wind*
Utilization My	0,33	0,27

σ_mzd 32,60507813

	Snow	*Wind*
Utilization Mz	1,18	0,97

Total Utilization when considering My and Mz

	Snow	*Wind*
Eq. 6.11	1,16	0,95
Eq. 6.12	1,41	1,16

6.1.7 (1) **Shear**

$$\tau_d \leq f_{v,d}$$

6.1.7(2)

$$\tau_d = \frac{V_d}{k_{cr}bh}$$

k_cr	0,8	(GL)

$$V_d = Max\{\, V_z; V_y\}$$

Vd	0,92	kN
τd	0,18	Mpa

	Snow	*Wind*
Utilization Vd	0,066	0,054

6.2.3 (1) **Combined bending and axial tension**

Eq. 6.17

$$\frac{\sigma_{t,0,d}}{f_{t,0,d}} + \frac{\sigma_{m,y,d}}{f_{m,y,d}} + k_m \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1$$

Eq. 6.18

$$\frac{\sigma_{t,0,d}}{f_{t,0,d}} + k_m \frac{\sigma_{m,y,d}}{f_{m,y,d}} + \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1$$

Utilization :	*Snow*	*Wind*
Eq. 6.17	1,17	0,95
Eq. 6.18	1,42	1,16

6.2.4 (1) **Combined bending and axial compression**

Eq. 6.19

$$\left(\frac{\sigma_{c,0,d}}{f_{c,0,d}}\right)^2 + \frac{\sigma_{m,y,d}}{f_{m,y,d}} + k_m \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1$$

Eq. 6.20

$$\left(\frac{\sigma_{c,0,d}}{f_{c,0,d}}\right)^2 + k_m \frac{\sigma_{m,y,d}}{f_{m,y,d}} + \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1$$

Utilization :	*Snow*	*Wind*
Eq. 6.17	1,17	0,95
Eq. 6.18	1,42	1,16

6.3.2(1) **Local Buckling of timber elements**

Assume buckling length of each member equals 0.5*L
Where L is the length of the beam member

l_ky	0,897	m
l_kz	0,897	m

Relative slenderness:

Eq. 6.21

$$\lambda_{rel,y} = \frac{\lambda_y}{\pi}\sqrt{\frac{f_{c,0,k}}{E_{0,05}}} \qquad \lambda_y = \frac{l_{k,y}}{\sqrt{\frac{I_y}{A}}}$$

Eq. 6.22

$$\lambda_{rel,z} = \frac{\lambda_z}{\pi}\sqrt{\frac{f_{c,0,k}}{E_{0,05}}} \qquad \lambda_z = \frac{l_{k,z}}{\sqrt{\frac{I_z}{A}}}$$

λ_y	38,84
λ_rel,y	0,56
λ_z	38,84
λ_rel,z	0,56

βc 0,1 GL

k-factor

Eq. 6.27

$$k_y = 0{,}5\ (1 + \beta_c\left(\lambda_{rel,y} - 0{,}3\right) + \ \lambda_{rel,y}^2)$$

k_y 0,671854638

Eq. 6.28

$$k_z = 0{,}5\ (1 + \beta_c\left(\lambda_{rel,z} - 0{,}3\right) + \ \lambda_{rel,z}^2)$$

k_z 0,671854638

Eq. 6.25

$$k_{c,y} = \frac{1}{k_y + \sqrt{k_y^2 - \lambda_{rel,y}^2}}$$

k_cy 0,65

Eq. 6.26

$$k_{c,z} = \frac{1}{k_z + \sqrt{k_z^2 - \lambda_{rel,z}^2}}$$

k_cz 0,65

Eq. 6.23

$$\frac{\sigma_{c,0,d}}{k_{c,y} f_{c,0,d}} + \frac{\sigma_{m,y,d}}{f_{m,y,d}} + k_m \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1$$

Eq. 6.24

$$\frac{\sigma_{c,0,d}}{k_{c,z} f_{c,0,d}} + k_m \frac{\sigma_{m,y,d}}{f_{m,y,d}} + \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1$$

Utilization Buckling

	Snow	*Wind*
Eq. 6.23	1,16	0,95
Eq. 6.24	1,41	1,16