Master's thesis

June-Marie Esjeholm
Sverre Magnus Haakonsen
Daniel May Instanes

# Design & Evaluation of Discrete Gridshells with Thrust Network Analysis

## Implementation of a parametric toolbox in Grasshopper

Master's thesis in Civil and Environmental Engineering

Supervisor: Anders Rønnquist

June 2020

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

NTNU
Norwegian University of
Science and Technology

June-Marie Esjeholm
Sverre Magnus Haakonsen
Daniel May Instanes

# Design & Evaluation of Discrete Gridshells with Thrust Network Analysis

Implementation of a parametric toolbox in Grasshopper

Master’s thesis in Civil and Environmental Engineering
Supervisor: Anders Rønnquist
June 2020

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

NTNU
Norwegian University of
Science and Technology

**Department of Structural Engineering**
Faculty of Engineering
**NTNU – Norwegian University of Science and Technology**

ACCESSIBILITY

# MASTER THESIS 2020

| | | |
|---|---|---|
| SUBJECT AREA:<br>Conceptual Structural Design | DATE:<br>10.06.2020 | NO. OF PAGES: |

TITLE:

**Design & Evaluation of Discrete Gridshells with Thrust Network Analysis**

Design og Evaluering av “Diskre Gitterskall” ved bruk av *Thrust Network Analysis*

BY:

June-Marie Esjeholm
Sverre Magnus Haakonsen
Daniel May Instanes

SUMMARY:
During the early stages of a building project, the structural efficiency of the building is seldom the main area of interest, and is often left for the structural engineer at a later stage. Consequently, the result often suffers from compromises between the architect’s vision and structural concerns.

Thrust network analysis is a form finding method developed by Block (2009). This thesis aims to investigate the functionality of the method by applying it to single layer gridshells. Using the visual programming language Grasshopper, a series of components are created to implement the method in a parametric environment. In addition, a proposal for a simplified global buckling analysis is presented. By using the projected geometry and results from the Thrust Network Analysis, the user gets an indication of the structures’ global stability without the need to establish a finite element analysis.

After the implementation, a series of case studies are conducted. The developed components are first tested on simple structures as a way of verifying the plausibility of the produced results. In addition, a case study is presented where the method is used as a part of a larger form finding procedure. Here the implemented components are used to create the initial global geometry before further optimisation is performed by the evolutionary solver *Galapagos*. Two design proposals for a timber gridshell hall is then developed using the created components. The components ability to create and modify a structure, before verifying the forces are evaluated here.

Finally, a series of arbitrary geometries are used to illustrate the flexibility of the created solver. Together, the case studies demonstrate how the created components can be used to create structural efficient gridshells in a conceptual design phase.

RESPONSIBLE TEACHER: Anders Rønnquist

SUPERVISOR(S): Steinar Hillersøy Dyvik & Marcin Luczkowski

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

# Abstract

During the early stages of a building project, the structural efficiency of the building is seldom the main area of interest, and is often left for the structural engineer at a later stage. As a consequence, the end result often suffers from compromises between the architect's vision and structural concerns.

Thrust network analysis is a form finding method developed by Block (2009) as a way of evaluating the stability of masonry structures. This thesis aims to investigate the functionality of the aforementioned method by applying it to single layer gridshells. Which in turn introduces new challenges where self-weight no longer acts as the governing load, and live load will consequently have a bigger impact on the design.

Using the visual programming language Grasshopper, a series of components are created in order to implement the method in a parametric environment. This enables both architect and engineers to explore the funicular shape of gridshells in early stage design with real time feedback on the viability of the given form.

In addition, a proposal for a simplified global buckling analysis is presented. By using the projected geometry and results from the Thrust Network Analysis, the user gets an indication of the structures' global stability without the need to establish a finite element analysis.

After the implementation, a series of case studies are conducted. The developed components are first tested on simple structures as a way of verifying the plausibility of the produced results. In addition, a case study is presented where the method is used as a part of a larger form finding procedure. Here the implemented components are used to create the initial global geometry, before further optimisation is performed by the evolutionary solver *Galapagos*.

Two design proposals for a timber gridshell hall is then developed using the components in collaboration with the PhD candidate in architecture Steinar Hillersøy Dyvik. The components ability to create and modify a structure, before verifying the forces are evaluated here.

Finally, a series of arbitrary geometries are used to illustrate the flexibility of the created solver. Together, the case studies demonstrate how the created components can be used to create structural efficient gridshells in a conceptual design phase.

# Sammendrag

I prosjekters startfase er det ofte arkitektens visjoner som spiller førstefiolin, og de lastbærende egenskapene er ikke nødvendigvis i fokus. Når ingeniører på et senere tidspunkt skal verifisere byggets egenskaper risikerer en at arkitektens opprinnelige ideer må ofres for fysikkens ufravikelige lover. En risikerer dermed et sluttresultat hvor visjon går på akkord med funksjon.

Thrust Network Analysis er en «formfinningsmetode» introdusert av Block (2009) . Opprinnelig ble metoden utviklet for å verifisere stabiliteten til mur- og steinkonstruksjoner. Denne oppgaven søker å evaluere nevnte metodes egnethet til bruk på «diskre gitterskall». Nye utfordringer introduseres når egenvekt i mindre grad dominerer lastbildet, og horisontallaster vil i større grad påvirke konstruksjonens form og funksjon.

Ved å programmere en serie komponenter som er kompatible med Grasshopper får både arkitekter og ingeniører muligheten til å utvikle materialeffektive gitterskall, som i hovedsak transporterer lastene gjennom membrankrefter, allerede fra designstart.

Videre introduseres en to-dimensjonal forenkling av en global stabilitetsanalyse. Gjennom å bruke den projiserte geometrien, og krefter fra Thrust Network Analysis, kan brukeren evaluere gitterskallets stabilitet uten å måtte etablere en komplett elementanalyse. Resultatene fra komponentene sammenlignes så med tall fra en elementanalyse for å kartlegge styrker og svakheter.

Oppgaven avslutter med ulike eksempelstudier. Først brukes verktøyet som en del av en større «formfinningsprosess» hvor den globale formen først etableres ved hjelp av de introduserte verktøyene, for så å optimaliseres ved hjelp av blant annet Galapagos. Etter dette blir det i samarbeid med Ph.d. kandidat i arkitektur, Steinar Hillersøy Dyvik, presentert et designforslag til en industrihall. Her brukes komponentene både til utforming og evaluering av konstruksjonen.

Avslutningsvis vil et variert utvalg former demonstrere fleksibiliteten til komponentene. Sammen med resten av eksempelstudiene viser denne masteroppgaven hvordan denne måten å jobbe på kan skape konstruksjoner som er både konstruktive og meningsfulle.

# Preface

This master thesis is written during the spring semester of 2020, as a concluding part of the Master of Science Degree at the Norwegian University of Science and Technology (NTNU), Department of Structural Engineering.

During our courses at NTNU we were fortunate enough to get a glimpse of the endless possibilities that existed when combining programming and parametric design. Intrigued by this and inspired by the courses *Architecture as a Technological Practice*(AAR4250) and *Structural Design, Advanced Course*(TKT4198), the goal of the thesis became clear. We wanted to develop something on our own that could also contribute to the shortening of the gap that exists between engineers and architects. Equipped with only basic programming knowledge, we ventured boldly into the fray notwithstanding; now we proudly present the culmination of our labour and intrepidness, id est this thesis.

We would like to offer our sincerest thanks to those who provided us with timely assistance: Anders Rønnquist for inspiring us to write about this topic and for all advice, guidance, and support that he so thoughtfully provided us. Steinar Hillersøy Dyvik for continuous help and for steering us along the right course. Bunji Izumi for constructive feedback and prudent recommendations. Bjørn Normann Sandaker for showing us engineers the ropes about architecture and all the practical help so selflessly given.

Also, a thanks to Marcin Luczkowski and Steinar Hillersøy Dyvik for encouraging us to write a conference paper from this thesis.

Finally, a great thanks to each other for all the motivation, support, and effortless teamwork.

# Introduction

What defines a great civil engineer? Besides a solid theoretical foundation, the proficiency to work together with the architect towards a common goal might be emphasised as the most important trait. A defining characteristic between the most successful building projects of today is the strong collaboration between engineers and architects throughout the entire process.

Following history, technological advances inspire new architecture which ,again, challenges engineers in a fresh way. With the immense amount of computational capacity available today, the need for digital competence is likely to increase significantly over the next years.

Algorithms-Aided Design as well as its growing popularity is both examples of this. By using flexible digital models based on algorithms, the behaviour of complex structures can be analysed, alternated, and optimised in a way static models would never allow, thus facilitating the creativity of new design methods.

Shell structures, either continuous or reticulated, are particularly suited for this type of design process. They are popular among architects for the possibility to create free form surfaces, and and among engineers for the material effective load transfer through membrane forces. Shell structures will often require different methods of design and verification than the typical "frame" often encountered in traditional buildings. Various methods of form finding is suitable for these structures as they seek to establish a shape based on equilibrium between external and internal forces. Starting from physical models, the technological revolution during the previous decades has enabled the development of several numerical form finding methods.

The aim for this thesis is to implement Thrust Network Analysis as a parametric form finding tool in Grasshopper as well as examine the possibilities and extensions of Thrust Network Analysis as a form finding tool for discrete timber gridshells. Since the global stability of single layer gridshell is a driving factor for design, the thesis includes a proposal to evaluate the global stability based on the projected geometry.

The components are then evaluated through a series of case studies where the components are both tested and used to create discrete gridshells. A suggestion to how the created tools may be included as a part of a complete form finding process is presented in Case Study III, while Case study IV proposes the design of an industry hall.

# Contents

# 1 Background

## 1.1 A brief history of form finding

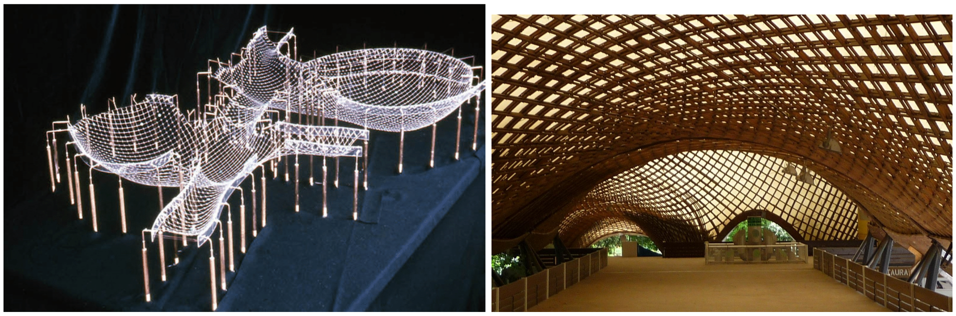


Figure 1.1: a) Hanging chain model of Mannheim Multihalle and b) the real construction. Images from: a) Liddell (2015) and b) Giel (2010).

Arches and curves are strong and stable constructions that have been widely used since the classical antiquity. The arch as a structural concept was known long before this, but it was the Romans who really utilised the structural opportunities that follow with the arch. At the time, stone and bricks were the main building materials. Due to its poor tensile capacity, long spanning structures were difficult to build. With arches carrying forces through compression, stone is a well-suited material for this task. This made the Romans able to build bridges, aqueducts, and large-scale architecture with stone arches, Shiveta Singh (2005).

From a scientific perspective, the mathematical theory did not describe the shape of the arches before the end of the 17th century. Robert Hook has gained much of the credit for what we today call catenaries. A catenary is the shape that occurs when a rope is freely suspended between two points. It achieves the shape that requires the least potential energy. This is a pure tensile form, with no moments. The arch is the steepest where it carries the most of the load, i.e closest to the support points. Hook figured that the most stable compression arch had to have the same shape as the catenary, but upside down. Just as the catenary is a pure tensile shape, the inverse arc becomes a pure compression shape without moments, Block et al. (2006).

By hanging lead weights in the catenary, it will always conform to the least potential energy principle. In this way, engineers and architects could hang weights in a rope and thus find the optimal curve for a given load. Further, this technique could then be used to design a scaled inverted shape of the final structure. The method eventually evolved into three-dimensional structures, and one could make models for more complex shell structures. Of course, this was a very time-consuming process.

The German architect Frei Otto, at the University of Stuttgart, was one of those who used such models on a large scale and explored the possibilities of form finding in the mid-20$^{th}$ century, Peteinarelis (2016). One of Frei Otto's largest project, Mannheim Multihalle, Figure 1.1, is based on this physical form finding. Mannheim Multihalle, a grid-shell structure in wood, was completed in 1975 and is still considered one of the world's largest and lightest compression structures Adriaenssens et al. (2014).

## 1.2 Different methods of form finding

"After the main introduction of computers in the 1960s, great advances in form finding were made from manual methods such as Antoni Gaudi's use of complex rope-systems to Frei Otto's (more unconventional use of) soap bubbles, and now digital methods have announced their coming with the computer's large computational capacity enabling form finding for anyone interested."

Listed below are the most common form finding methods reviewed. These are briefly presented, before an in-depth presentation of Thrust Network Analysis (TNA) is given in the next chapter.

### 1.2.1 Force Density Method

The Force Density Method was first introduced by Schek (1974) two years after the Munich Olympics, where it was developed to perform digital analysis of the cable network at Frei Otto's famous Olympic Stadium. The method finds an equilibrium state of the cable network, where the cables often have different elastic properties when external loads are applied.

The advantage of this method is the ability it provides to solve any general equilibrium state of a system through only a single set of linear equations; where the coordinates of the nodes are the unknowns. This solution is generated by introducing the relationship between the force and length of the branches ("force density") as the degrees of freedom of the system. Based on this one parameter, several different solutions can be generated quickly by prescribing different force densities in the system's branches.

The material properties are not introduced until after the introductory forms have been established. The method is thus material independent. Material can be introduced individually to each member of the system without deviating from the original form..

The method is still popular due to its applicability and simple numerical implementation. It is often used to generate various alternatives, which are then used for more accurate non-linear analyses where global loads such as snow, wind, and self-weight are introduced.

In recent years, extensions of the "Force Density Method" have been proposed, including "multi-step FDM" by Sánchez et al. (2007). Which, like the original method is intended for use in the initial design phase. This is an iterative equation solver that seeks an even distribution of stresses throughout the system.

### 1.2.2 Dynamic relaxation

Another important method in form finding is dynamic relaxation. This is a numerical method that was first introduced by Day (1965). Day used the method to analyse tension structures such as suspended ceilings. The method is based on following the oscillations of a structure over time for a given load. This way one can find the shape which causes the forces to be in equilibrium, and thus a stable structure. This is an iterative process where the geometry is optimised continuously Adriaenssens et al. (2014).

Gradually, dynamic relaxation has been found to have far more applications than just tensile structures; for example, the method was used early on for the analysis of thin shells and plates. More recently, the method has evolved and become more applicable to advanced problems, including reduced data storage needs. This has made the method suitable for analysis of structures with large deflections, elastic instability, and plastic problems. The method is also widely used for non-linear analyses of both non-linear materials and geometry Barnes (1999).

A disadvantage of dynamic relaxation is that it requires many input parameters, including time intervals for load and oscillations. In addition, Nouri-Baranger (2004) states that the values used for mass and damping are fictitious, and are therefore perceived as less significant as they do not have a physical, real explanation.

### 1.2.3 Particle-spring systems

Particle-spring systems are based on a series of lumped masses connected together by linear elastic springs. Each spring is then assigned a constant stiffness, initial length, and damping coefficient. When displayed from their initial position a force is generated by the springs. The solving procedure for equilibrium positions of the geometry can be implemented by both implicit and explicit solving techniques. A popular method, using an implicit Runge-Kutta solver, is presented by Kilian & Ochsendorf (2005).

Where the Force Density Method and Dynamic Relaxation are well suited for pre-stressed structures where a certain level of internal force is present, such as cable nets; the Particle-springs method performs better when it comes to statically determinate structures in pure compression or tension. For example the case of hanging chains exposed to only self weight from section 1.1.

Additionally, the method allows the user to change the form and force in real time. Hence the user quickly experiences how the force influences the form and vice versa. An easy-to-use, open source implementation of this is presented by Kilian (2004).

## 1.3 Shells and gridshells

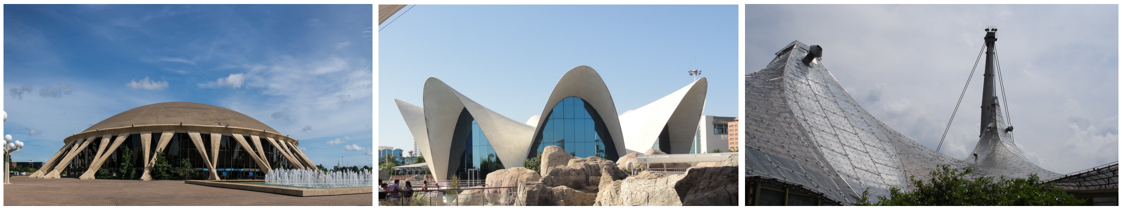


Figure 1.2: a) Palazzo dello sport by Nervi in Roma, Kirsch (2012). b)L'Oceanogràfic by Candela, Wang (2011). c) Olympic stadium by Frei Otto, Brown (2008).

Calladine (1989) defines shells as curved or doubly curved solid surfaces, carrying their loading through a combination of membrane and bending action. A thin shell is a shell with a high width to thickness ratio, and carries most of its loading through membrane forces. This makes it highly efficient with regard to both material usage as well as its ability to span over long distances. Through the $20^{th}$ century, architects and engineers such as Frei Otto, Pier Luigi Nervi, and Félix Candela have been using thin shells in some of their most well-known architectural features. Monuments such as "Palazzo dello Sport", "Hipódromo de la Zarzuela", and "Mannheim Multihalle", Figure 1.2, are all interesting structures not only from an architectural point of view, but also from an engineer's perspective.

Thin shells may generally be divided into three main categories:

- **Concrete shell structures**, e.g. Figure 1.2 (a) and (b) which displays Nervi's Palazzo dello Sport and Candela's l'Oceanografic. Both show how a traditionally heavy and substantial material such as concrete may be used in creation of light and airy structures.

- **Membrane structures**, i.e. material such as cloth or cable net where the loads are transferred purely by membrane forces illustrated by Frei Otto's Olympic Stadion in Figure 1.2 (c).

- **Lattice structures**, consisting of grid elements as opposed to the solid concrete shell. The materials used here include timber, steel, aluminium, cardboard, and composite material.

Lattice structures, also known as gridshells, share many of the same properties as a continuous concrete shell such as getting its strength from a curved or double curved surface, Douthe et al. (2006). The biggest difference in behaviour between the two is shown in Figure 1.3. For a regular thin shell element, all the membrane forces are present, while for a rectangular gridshell the lack of the membrane forces $N_{xy}$ and $N_{yx}$ makes a single element prone to large deformations and rotations. A gridshell element therefore has the need for diagonal bracing in order to resist in-plane shear.

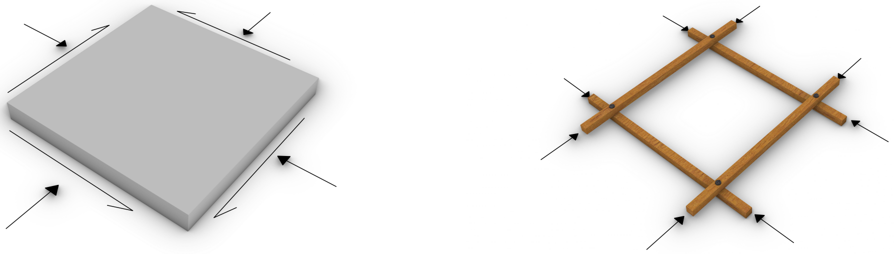


Figure 1.3: Load transfer of membrane forces in (a) concrete thin shell, and (b) timber gridshell. For the concrete gridshell all membrane forces $N_x$, $N_y$, $N_{xy}$, and $N_{yx}$ are present, while the timber gridshell transfers its loads through $N_x$ and $N_y$ only.

A great advantage of gridshells is the possibility to prefabricate all the individual elements off-site in a controlled environment before assembling it on site. A large number of simple elements can therefore create complex doubly curved geometry that would be difficult to achieve with concrete. By adding a little thought into the design process, they are also suitable for disassembling and reuse. The way the members are connected in such gridshells may be categorised into two types, as done by Harris et al. (2003):

- **Continuous grid members**, with long laths spanning over the whole structure, and overlapping each other at the nodes.
- **Discrete members**, with grid members that connect at each node.

The construction of continuous gridshells may be done by laying out an initial planar grid, before lifting it up to the desired shape. This technique was first developed by Professor Frei Otto, and used on a large scale for the Mannheim Multihalle in 1975 with Arup and Bruno Happold as structural engineers. The entire structure consists of two domes, spanning 40 m and 60 m respectively, with connecting pathways.The gridshell enclosing the interior is built by a series of 50 mm x 50 mm hemlock sections joined together in a double-layered gridshell configuration. The complete process is explained by Happold & Liddell (1975). Even with the limited computational power available back then, the potential of timber gridshells was clearly illustrated by mainly using manual form finding methods such as the inverted hanging chain when developing its form.

With the curvature of the continuous gridshells giving it the strength, the nodal connections are fairly simple to produce. Despite this, the use of such gridhsells are most common in temporary structures such as pavilions due to the limited availability of building materials having the required flexibility.

Discrete gridshells on the other hand consist of straight members, with the nodal connection ensuring the global curvature of the structure. This enables a larger variety of materials applicable to serve as structural members. However, as the complexity of the gridshell increases, the nodal connections quickly become costly and cumbersome to produce. With modern fabrication methods the ability to mass produce unique nodal connections enables discrete gridshell to be relevant for a broad range of buildings.

One such example is the Great Court roof at the British Museum. Developed by Foster + Partners between 1994-2000, the roof consists of almost 5000 grid elements connected to over 1500 nodes. All of them are prefabricated (Sischka (2000)). The elements and nodes were all numbered during the fabrication and thus converting the construction process to a huge three-dimensional puzzle.

Discrete gridshell has greater possibilities when it comes to complex structures, since the elements can be orientated in any direction and constructed in any lengths. Another advantage regarding timber gridshells, is the possibility to create large structures with elements where irregularities such as knots are removed. This enables the use of a higher material strength in the analysis. It is this type of gridshell that will be investigated throughout this thesis.

## 1.4 Global stability of gridshells

With a slender, and often long-spanning geometry, an important aspect when designing gridshells is the global stability. Especially single-layer gridshells where the nodal connections need to have a certain amount of rotational stiffness in order to work as a uniform surface. The idealisation of rigid/pinned connections therefore quickly becomes inaccurate in these structures; rigid joints in timber gridshells are close to impossible to achieve.

There is a shortcoming on research about global buckling of gridshells. One of the aspects that has not been in focus is how to accurately model the joints which in reality behave somewhere between the rigid and pinned assumptions. Multiple factors influence this; from the number of elements connected to the joint, the lengths and stiffness of these as well as the geometry and material of the joint itself. All these factors influence the performance of the structure's global stability.

It is not possible to estimate an accurate value of the joint stiffness in the early design phase, as it will depend on the specific case. Therefore, an initial analysis using pinned and rigid joints can give an estimate of whether the structure needs bracing or not. Mesnil et al. (2017) states that "quadrangular gridshells rely on the bending stiffness of connections, whereas triangulated gridshells benefit from a shell-like behaviour without the need for rigid connections". Accordingly. the chosen mesh geometry is also important in the case of global stability.

Bulenda & Knippers (2001) describes the typical buckling types for gridshells as:

- Local member buckling.
- Local instability (snap through of one or several nodes).
- Global instability.
- Combinations of the types above.

When analysing the stability of gridshell structures, they found that the only way to consider all the buckling modes was by analysing the structures' imperfections in an FEA program. These imperfections include system imperfections such as semi-rigid joints, structural imperfections due to cross sectional area, curvature of members, loading, and geometrical deviations. The article suggests analysing the structure with applied eccentricities based on scaling the structures eigenmodes.

The extended research from this have mainly been focused on dome-like structures. Malek et al. (2014) did a study on a closed spherical cap, and developed a framework for a quick assessment of the buckling performance of gridshells, using closed form solutions of continuous shells. They evaluated grid density and height to span ratios and found that:

- Different gridshell topologies and curvatures can have the same structural efficiency.
- Definitions of the equivalent continuum from continuous shells can provide quick estimates for the buckling load of gridshell structures.
- Shallow shells require a denser grid to maintain its stability.
- Triangular gridshells are sensitive to geometric imperfections; quadrilaterals are not.

Hwang et al. (2009) did a thorough study on how various types of nodes affected the global stability of steel gridshells. They found that high-rise gridshells are more sensitive to changes in nodal stiffness, and that low-rise structures needed very high nodal stiffness to avoid buckling.

## 1.5 Applied software

During the development and verification of the components created throughout the thesis a variety of digital software is used. A brief introduction to these is presented below.

**Rhinoceros 6**
*Rhinoceros* (2020), often abbreviated as Rhino, is a commercial, computer-aided design (CAD) program developed by Robert McNeel & Associates. Based on the NURBS mathematical model, it produces mathematically precise representations of complex geometry and free form surfaces, making it a popular tool for architects as well as designer, and thus a suitable platform for developing a tool which aims to make structural design more readily available for architects.

**Grasshopper**
*Grasshopper* (2020) is an official package included in Rhino. Allowing users without knowledge of scripting the possibility to create generators and algorithms through pre-made code blocks, the program is popular among engineers and architects alike. The program also allows third-party contributors to create and share their self-made components.

Along with the core components of Grasshopper, the following packages have been used:

- *Galapagos* (2020): An easy-to-use genetic algorithm component created by David Rutten. As an evolutionary solver it uses a series of variables called "genes" as input, and modifies these to maximise or minimise a fitness function by iterating multiple generations of different solutions. Each generation creates multiple solutions where the ones closest to the fitness function are used to breed the next generation. This process goes on until a satisfactory solution is found.
- *Silvereye*: An optimisation tool based on Particle Swarm Optimisation developed by Cichocka et al. (2017). Used as an alternative to the *Galapagos* evolutionary solver when a faster and more computational effective solution is sought.
- *Karamba3D* (2020): Enables a full finite element analysis (FEA) in the parametric environment of Grasshopper. The plug-in is a powerful tool for fast and efficient evaluations of multiple geometries in the conceptual phase of a project.
- *Kangaroo* (2020): Live physics engine which can be used for mesh optimisation and form finding.
- *Lunchbox* (2020): A plug-in containing useful geometry tools for panelling surfaces, structures etc.
- *Weaverbird* (2020): Mesh editing plug-in which simplifies the creating and editing of meshes.

**Visual Studio**
The development of components is done with the integrated development environment Visual Studio from Microsoft. Supporting C#, which is the native language of Rhino, a full analysis and debugging session can be performed directly in Rhino via Visual Studio. Some external packages are used to implement mesh modifications, and linear optimisation:

- *Google.OR-Tools* (n.d.): An open source software used to integrate a linear optimisation routine in the components.
- *Plankton* (2017): Library implementing the half-edge data structure used to store information about polygon meshes.

# 2 Theory

In this chapter the theoretical background behind the TNA, and relevant theory are first introduced, before an introduction to local and global stability problems is provided.

## 2.1 Thrust Network Analysis

Introduced by Block (2009), the TNA was developed to verify the structural performance of funicular masonry structures. The greatest strength of the method compared to for instance the Force Density Method , which it is based upon, is the use of reciprocal diagrams. Which enables a graphical explanation on how the forces affect the form and vice versa.

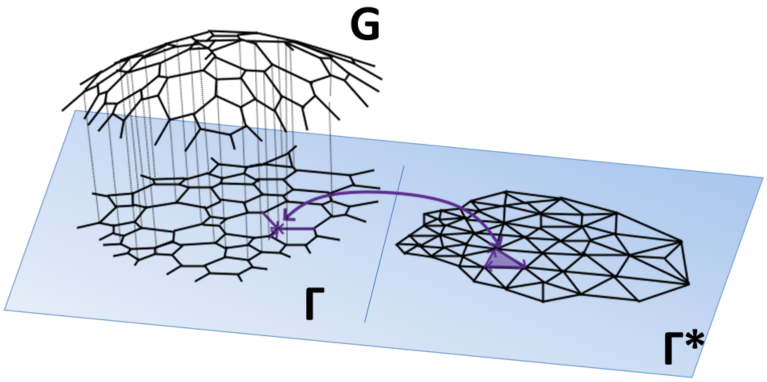


Figure 2.1: TNA: Relationship between Thrust network (**G**), primal grid (**Γ**), and dual grid (**Γ***).

Figure 2.1 illustrates the three domains used within the TNA framework. The Primal grid, **Γ**, which is the two-dimensional projection of the spatial thrust network, **G**, and the Dual grid, **Γ***. All of the three domains are reciprocal, meaning that a change in one domain will affect the others.

A physical interpretation for the three domains may be expressed as follows:

- The thrust network, **G**, represents the global 3D shape using nodes and branches.
- The primal grid, **Γ**, obtained by projecting the branches from **G** down to the xy-plane.
- The dual grid, **Γ***, is the reciprocal of **Γ**. The branches coming into a node in the primal grid will form a closed polygon in the dual grid. **Γ*** thus represents the horizontal forces in the corresponding branches in **Γ**.

## 2.2 Reciprocal diagrams

An important part of the TNA framework developed by Block (2009) is the use of reciprocal diagrams. It gives the user an intuitive understanding of force distribution, and through modifications of the force diagram, a new form will follow. This is a principle based on a pair of figures where the properties of the first figure relative to the second one, are the same as those of the second one to the first, Maxwell (1864). By definition, two figures are reciprocal when they have an equal number of lines, with the lines being parallel. The lines which in one Figure converges towards a point in one plane, must form a closed polygon in the other plane.

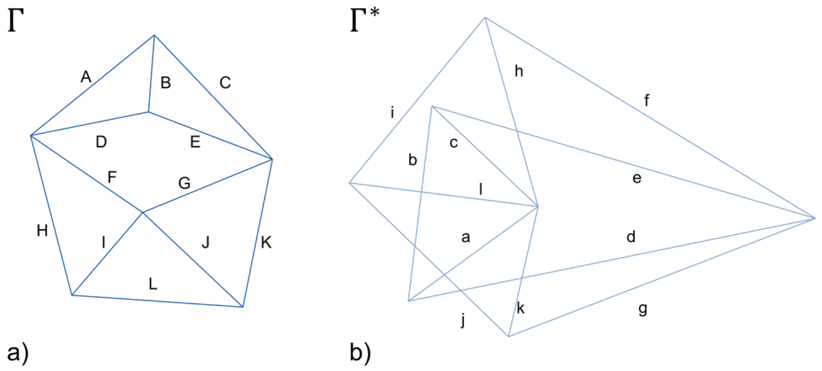


Figure 2.2: a) Primal grid **Γ** and b) dual grid **Γ***. All lines denoted in upper-case letters are parallel to the corresponding lower-case letters in the dual grid.

For the example in Figure 2.2, the primal grid **Γ** consists of seven points and twelve lines. To construct the dual grid **Γ*** the easiest way is to start with a point in the primal grid with three branches connected to it. Since every line in the primal grid only converges toward two points, the corresponding line in the dual grid can only belong to two polygons.

In statics, the primal grid **Γ** would be the orientation of for instance, bars; then the dual grid **Γ*** becomes the corresponding force diagram. For each point in the primal grid, a closed polygon in the dual grid consisting of the lines converging to said point would ensure the point being in static equilibrium. In the TNA, reciprocal diagrams gives the user a visual representation of forces, making the method very intuitive to use.

### Determinate and indeterminate networks

Let $s$ be the number of points, $e$ the number of lines, and $f$ the number of polygons or faces in the primal grid. Assuming that the first line determines two points, the remaining $s-2$ points are determined by $2(s-2)$ lines. Hence if the number of lines equals:

$$e = 2s - 3 \tag{2.1}$$

then every point in the primal grid may be determined, and the system itself would then be statically determined. Should the number of lines be less, the system is statically indeterminate. For more lines than in (2.1), construction of the system is impossible unless certain conditions are fulfilled.

The above formulation describes the conditions for drawing any diagram with an arbitrary direction of lines. For the case of the TNA, the primal grid is first established before seeking a valid dual grid, thus ensuring horizontal equilibrium of the system (section 2.1). For such systems, when $e > 2s - 3$, the direction of lines are no longer arbitrary, but rather subjected to $e - (2s - 3)$ conditions.

Defining *s'*, *e'*, and *f'* as the values of *s*, *e* and *f* in the reciprocal diagram $\mathbf{\Gamma}^*$

$$e = e', \quad s = f', \quad f = s', \tag{2.2a}$$

$$e = s + f - 2, \quad e' = s' + f' - 2. \tag{2.2b}$$

If $s = f$, then $e = 2i - 2$. This ensures that at least one condition connecting the directions of the lines in the primal grid is met/is fulfilled, consequently ensuring the existence of a dual grid. If

$$s > f, \quad e > 2s - 2, \ \ and \ \ e' < 2s' - 2; \tag{2.3}$$

then the construction of a dual grid is possible, but indeterminate to the extent of $s - f$ variables. For $s < f$ the dual grid will be impossible to construct unless (s-f) conditions are fulfilled in the original diagram.

Going back to the example in Figure 2.2, the number of lines $e = 12$, meaning that from eq. 2.1 the system is indeterminate and subjected to $12 - (2 * 7 - 3) \ = \ 1$ condition. Since the primal grid is already drawn, the principles described above may be used to create a dual grid with the properties shown in Table 2.1

Table 2.1: Properties of the system in Figure 2.2.

| Properties of Figure 2.2 | | | |
|---|---|---|---|
| **Γ** | s = 7 | e = 12 | f = 7 |
| **Γ***  | s' = 7 | e = 12 | f' = 7 |

## 2.3 Computational set-up of Thrust Network Analysis

The computational structure to implement the TNA is parted into the equilibrium equations, branch node structure, matrix notation, and the implementation. The formulas introduced by Block (2009), are presented according to chapter 7 in *Shell Structure for architecture*, Adriaenssens et al. (2014).

### 2.3.1 Geometry

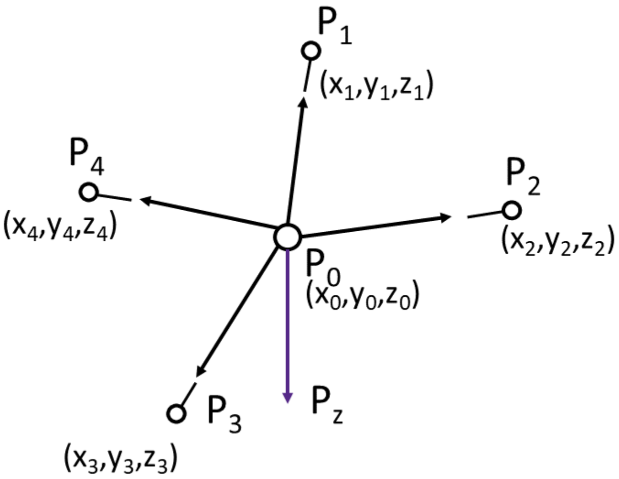


Figure 2.3: A single node $P_0$ and the connected nodes.

The structure is divided in nodes and branches. Both the nodes and branches are numerated, and sorted into matrices during computation. The coordinates of the nodes are collected in coordinate-vectors:

$$\boldsymbol{x} = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 & x_4 \end{bmatrix} \tag{2.4a}$$

$$\boldsymbol{y} = \begin{bmatrix} y_0 & y_1 & y_2 & y_3 & y_4 \end{bmatrix} \tag{2.4b}$$

$$\boldsymbol{z} = \begin{bmatrix} z_0 & z_1 & z_2 & z_3 & z_4 \end{bmatrix} \tag{2.4c}$$

The length of each branch is found by first computing the coordinate-differences, as such:

$$\boldsymbol{u} = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 & x_4 - x_0 \end{bmatrix}^T \tag{2.5a}$$

$$\boldsymbol{v} = \begin{bmatrix} y_1 - y_0 & y_2 - y_0 & y_3 - y_0 & y_4 - x_0 \end{bmatrix}^T \tag{2.5b}$$

$$\boldsymbol{w} = \begin{bmatrix} z_1 - z_0 & z_2 - z_0 & z_3 - z_0 & z_4 - z_0 \end{bmatrix}^T \tag{2.5c}$$

The length is then given by:

$$l_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \tag{2.6}$$

The branch node matrix **C** is constructed in order to maintain information about the connectivity of branches and nodes during computations. The matrix does not contain any information about the geometry . It only stores the connection between branches and nodes. For each branch *j* connected to node *i* the C-component is:

$$C_{ij} = \begin{cases} +1 & \text{if the branch begins in node i} \\ -1 & \text{if the branch ends in node i} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

For the example in Figure 2.3, the C-matrix is given as:

$$\boldsymbol{C} = \begin{bmatrix} +1 & -1 & 0 & 0 & 0 \\ +1 & 0 & -1 & 0 & 0 \\ +1 & 0 & 0 & -1 & 0 \\ +1 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{2.8}$$

The coordinate-difference vectors in (2.5) can now expressed by the **C**-matrix and coordinates of each node:

$$\boldsymbol{u} = \boldsymbol{C}\boldsymbol{x} \tag{2.9a}$$

$$\boldsymbol{v} = \boldsymbol{C}\boldsymbol{y} \tag{2.9b}$$

$$\boldsymbol{w} = \boldsymbol{C}\boldsymbol{z} \tag{2.9c}$$

With **U**, **V**, and **W** being the corresponding diagonal matrices used to express the branch length matrix **L**.

$$\boldsymbol{L} = \sqrt{\boldsymbol{U}^2 + \boldsymbol{V}^2 + \boldsymbol{W}^2} \tag{2.10}$$

Finally the nodes are sorted as free and fixed, with the matrices being sorted according to this:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_N & \boldsymbol{x}_F \end{bmatrix} \tag{2.11a}$$

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_N & \boldsymbol{y}_F \end{bmatrix} \tag{2.11b}$$

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_N & \boldsymbol{z}_F \end{bmatrix} \tag{2.11c}$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_N & \boldsymbol{C}_F \end{bmatrix} \tag{2.12}$$

### 2.3.2 Connection between primal and dual grid

As explained in section 2.2 the primal and dual grid are reciprocal to each other. In the TNA this is used to present the horizontal projection of branches in the primal grid (form diagram) and the corresponding branch forces in the dual grid (force diagram).

A node in the primal grid will belong to a space in the dual grid and is labelled with numbers (1,2,3...). Each corresponding branch in the primal and dual grid is parallel to the other and labelled in roman numerals (I, II, III...). The faces in the primal grid correspond to an associated node in the dual grid, and are labelled by lower- and upper-case letters (a A, b B...). Meaning that each node in the form diagram is represented as a face in the force diagram. All forces connected to the node will be the branches that make up the shape of the face. This is illustrated in Figure 2.4.

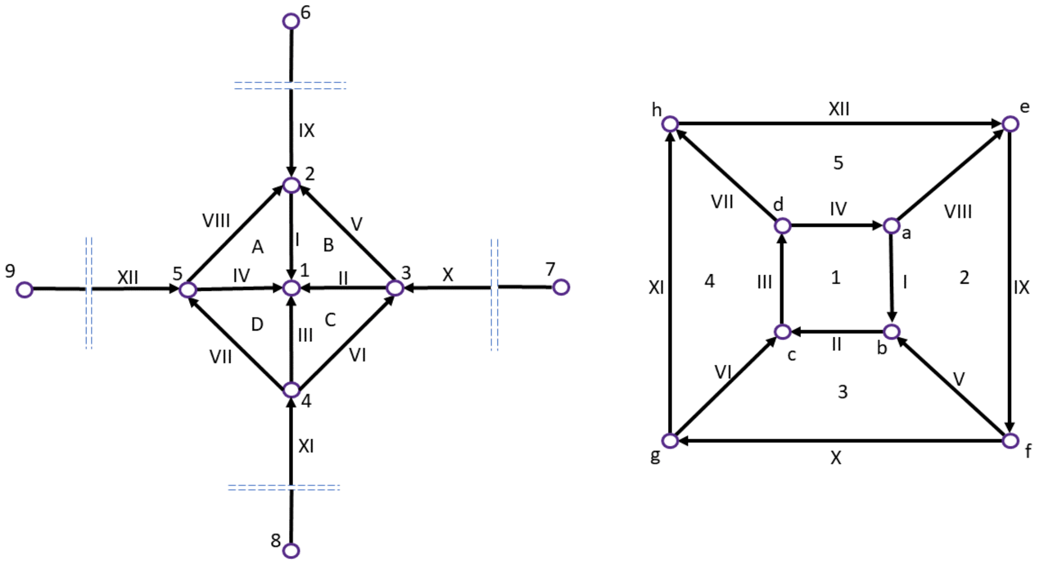


Figure 2.4: Form diagram (primal grid $\mathbf{\Gamma}$) and force diagram(dual grid $\mathbf{\Gamma}^*$).

Just as for the primal grid $\Gamma$, which has the connectivity store in the branch node matrix $\mathbf{C}$, the connectivity of the dual grid $\mathbf{\Gamma}^*$ is stored in the $\mathbf{C}^*$ matrix. It is built such that each branch in the primal grid is connected to a face rather than a node, which is the case for the $\mathbf{C}$-matrix. If a branch is oriented clockwise to the face, then its entry will be 1, and -1 if it is counterclockwise. This is illustrated in Figure 2.5.

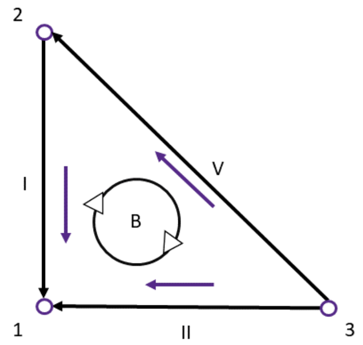


Figure 2.5: The branches of the grid can either be oriented clockwise or counterclockwise to the face.

$$C^*_{ij} = \begin{cases} +1 & \text{if the branch is oriented clockwise to the face} \\ -1 & \text{if the branch is oriented counterclockwise to the face} \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

Using the definition in equation (2.13) for Figure 2.4, the (m x n) **C**-matrix becomes:

$$C = \left[\begin{array}{ccccc|cccc}
1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
1 & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
1 & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
1 & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & 1 & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & 1 & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & 1 & \cdot & \cdot & \cdot & -1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & -1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & -1
\end{array}\right] \tag{2.14}$$

In the dual grid there is $m$ branches and $n^*$ nodes, and the $\mathbf{C}^*$-matrix (m x n$^*$) is constructed in the same way as the C-matrix.

The **C**$^*$-matrix for Figure 2.4 becomes:

$$C^* = \begin{bmatrix} -1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & -1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & -1 & 1 & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & -1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & -1 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & -1 \\ -1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & -1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & -1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & 1 \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & -1 \end{bmatrix} \tag{2.15}$$

### 2.3.3 From single node to TNA

By only considering the external vertical forces, the equilibrium equations are simplified such that the horizontal forces become independent of external forces. As a result of this, the form finding process may be divided into two parts:

1. Solve horizontal equilibrium.
2. Find the nodal heights based on the loads.

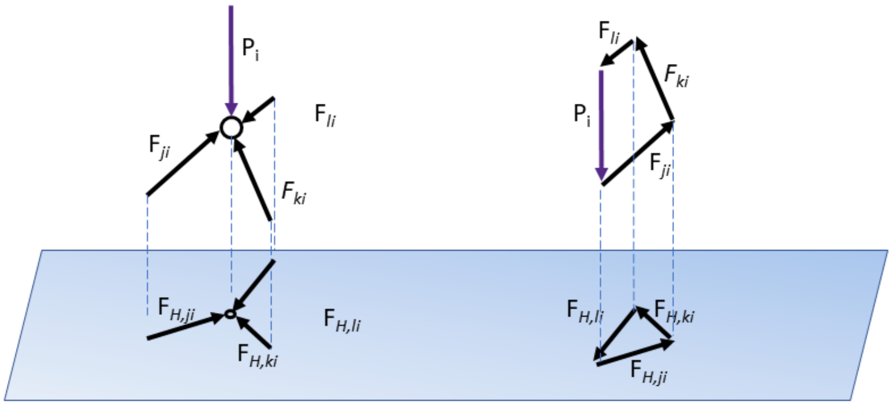


Figure 2.6: Static equilibrium of a node under vertical loading.

For the node in Figure 2.6 the equilibrium is given by:

$$F_{H,ji} + F_{H,ki} + F_{H,li} = 0 \tag{2.16a}$$

$$F_{V,ji} + F_{V,ki} + F_{V,li} = P_i \tag{2.16b}$$

where $F_{H,ji}$ is the horizontal component and $F_{V,ji}$ is the vertical component of the axial force $F_{ji}$; $P_i$ is the vertical force applied in the node. The horizontal components $F_{H,ji}$ are the forces in the force diagram $\mathbf{\Gamma}^*$. Since form ($\mathbf{\Gamma}$) and forces ($\mathbf{\Gamma}^*$) are reciprocal, the force $F_{H,ji}$ can be scaled by a factor $r$ to find the lengths of the lines in $\Gamma^*$:

$$F_{H,ji} = \frac{1}{r} \times l^*_{H,ji} \tag{2.17}$$

where the reciprocal branch length is given by:

$$l^*_{H,ji} = \sqrt{(x^*_i - x^*_j)^2 + (y^*_i - y^*_j)^2} \tag{2.18}$$

In the original domain $\Gamma$, the branch length is given as:

$$l_{H,ji} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2.19}$$

Vertical equilibrium is then expressed as:

$$F_{H,ji} \times \frac{(z_i - z_j)}{l_{H,ji}} + F_{H,ki} \times \frac{(z_i - z_k)}{l_{H,ki}} + F_{H,li} \times \frac{(z_i - z_l)}{l_{H,li}} = P_i \tag{2.20}$$

and further multiplied with $r$:

$$l^*_{H,ji} \times \frac{(z_i - z_j)}{l_{H,ji}} + l^*_{H,ki} \times \frac{(z_i - z_k)}{l_{H,ki}} + l^*_{H,li} \times \frac{(z_i - z_l)}{l_{H,li}} = P_i \times r \tag{2.21}$$

and rearranged to:

$$\left(\frac{l^*_{H,ji}}{l_{H,ji}} + \frac{l^*_{H,ki}}{l_{H,ki}} + \frac{l^*_{H,li}}{l_{H,li}}\right) \times z_i - \frac{l^*_{H,ji}}{l_{H,ji}} \times z_j - \frac{l^*_{H,ki}}{l_{H,ki}} \times z_k - \frac{l^*_{H,li}}{l_{H,li}} \times z_l - P_i \times r = 0 \tag{2.22}$$

and expressed with constants $d$:

$$d_i \times z_i - d_j \times z_j - d_k \times z_k - d_l \times z_l - P_i \times r = 0 \tag{2.23}$$

When the geometry of the structure is expressed on matrix-form as given in chapter 2.3.1, (2.22) is accordingly rewritten as:

$$\mathbf{C}_N^T(\mathbf{L}_H^{-1}\mathbf{L}_H^*)\mathbf{C}\mathbf{z} - r\mathbf{p} = \mathbf{C}_N^T(\mathbf{T})\mathbf{C}\mathbf{z} - r\mathbf{p} = 0 \tag{2.24}$$

and by using $\mathbf{D}_N = \mathbf{C}_N^T\mathbf{T}\mathbf{C}_N$ and $\mathbf{D}_F = \mathbf{C}_N^T\mathbf{T}\mathbf{C}_F$, equation (2.24) can be simplified to:

$$\mathbf{D}\mathbf{z} - r\mathbf{p} = \mathbf{D}_N\mathbf{z}_N - \mathbf{D}_F\mathbf{z}_F - r\mathbf{p} = 0 \tag{2.25}$$

Thereby the unknown z-coordinates of the free nodes are given by:

$$\mathbf{z}_N = \mathbf{D}_N^{-1}(p\mathbf{r} - \mathbf{D}_F\mathbf{z}_F) \tag{2.26}$$

### 2.3.4 Workflow

Figure 2.7 illustrates the different steps in the TNA-process. Starting with the user defining a form diagram, the connectivity matrices **C** and $\boldsymbol{C}^*$ are constructed from this, and the force diagram is then produced. The force diagram represents one of the possible states of equilibrium in **G**. The loads are then calculated before the nodal heights are found; finally the result is visualised in **G**.

If the result is not satisfactory, the user can go back and change both heights in $\boldsymbol{z_N}$, scale factor **r**, form diagram **Γ**, or force diagram $\boldsymbol{\Gamma}^*$.

Form finding by the TNA is therefore an iterative process where different factors are adjusted until a satisfying end-result is met; consequently the method suited for parametric modelling.

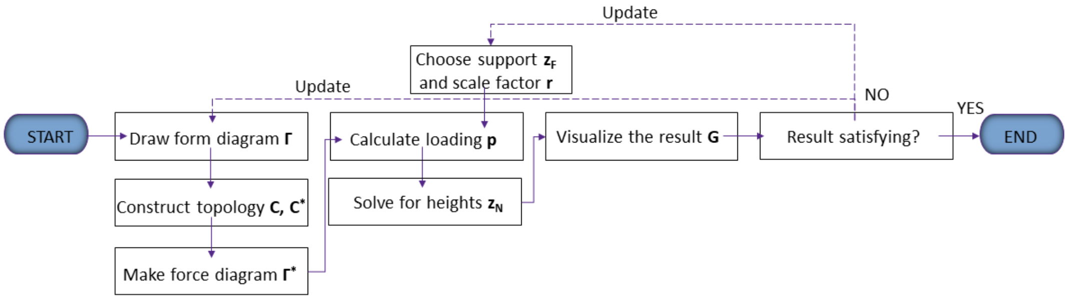


Figure 2.7: General workflow of the TNA.

### 2.3.5 Linear optimisation

Statically indeterminate networks have an infinite number of possible solutions. The problem must therefore be solved by linear optimisation.

Equation (2.26) will not take into account either the constrains on nodal heights or the scale factor. The problem is formulated to a linear optimisation problem, such that the solution lies between the given nodal boundaries, and the scaling factor is minimised. First the equation must hold; then the z values are given its boundaries. The linear optimisation problem then becomes:

$$\min_{z,r} \quad \pm r \quad \text{subject to} \quad \begin{cases} \mathbf{Dz} - r\mathbf{P} = 0 \\ \mathbf{z}^{LB} \leq \mathbf{z} \leq \mathbf{z}^{UB} \\ 0 \leq r \leq +\infty \end{cases} \tag{2.27}$$

The branch forces **S** may then be obtained from the relation:

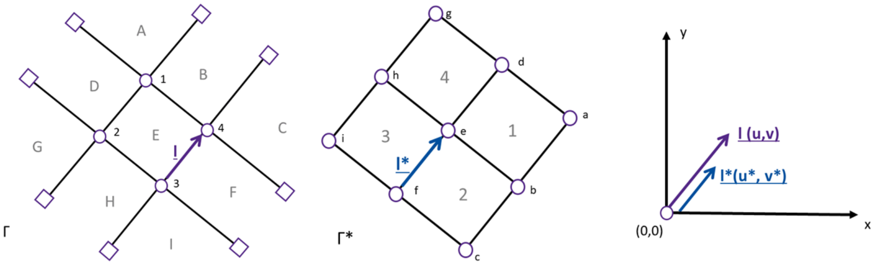
$$\boldsymbol{S} = \boldsymbol{Lq} = \boldsymbol{LL}_{\boldsymbol{H}}^{-1}\boldsymbol{L}_{\boldsymbol{H}}^{*}\zeta \tag{2.28}$$

To solve this, the reciprocal branch lengths ($\mathbf{L_H}$)$^{-1}$ has to be established through a new set of linear optimisation problems. If the network is indeterminate, there is an infinite number of solutions. To find a possible solution that satisfies the conditions in (2.2), three linear optimisation problems are defined.

In the computation of the reciprocal diagrams only the free nodes are considered, as the fixed nodes remain unaltered.

Both in the primal and dual graph the branches can be written as vectors.

$$\mathbf{l}_i = \begin{bmatrix} l_{i,x} \\ l_{i,x} \end{bmatrix}^t = \begin{bmatrix} u_i \\ v_i \end{bmatrix}^t = \begin{bmatrix} C_{ij}x_j \\ C_{ij}y_j \end{bmatrix}^t \quad , \quad \mathbf{l}_i^* = \begin{bmatrix} l_{i,x}^* \\ l_{i,x}^* \end{bmatrix}^{t^*} = \begin{bmatrix} u_i^* \\ v_i^* \end{bmatrix}^{t^*} = \begin{bmatrix} C_{ij}^*x_j^* \\ C_{ij}^*y_j^* \end{bmatrix}^{t^*} \tag{2.29}$$



a) Primal Grid Branch

b) Dual Grid Branch

c) Both Branches in same axis system

Figure 2.8: Connected branches in Primal and Dual grid are parallel to each other.

As illustrated in Figure 2.8, the corresponding branches in the primal and dual grid will have the direction vector, and is therefore only scaled in comparison to each other. Now $\mathbf{l}_i^*$ can be written as a function of $\mathbf{l}_i$.

$$\mathbf{l}_i^* = t_i * \mathbf{l}_i \tag{2.30}$$

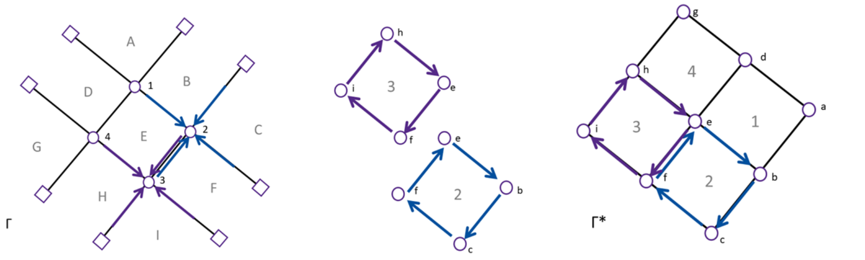
This results in the following constraints for the linear optimisation problem:

$$\begin{cases} u_i^* - u_i * t_i = 0 \\ v_i^* - v_i * t_i = 0 \\ t_i \geq 0 \end{cases} \tag{2.31}$$

where both $u_i$ and $v_i$ are known from the primal grid. This can be rewritten to matrix form as:

$$\begin{cases} \mathbf{U}^* - \mathbf{Ut} = \mathbf{I}_m\mathbf{u}^* - \mathbf{Ut} = \mathbf{0} \\ \mathbf{V}^* - \mathbf{Vt} = \mathbf{I}_m\mathbf{v}^* - \mathbf{Vt} = \mathbf{0} \\ \mathbf{t} \geq \mathbf{0} \end{cases} \tag{2.32}$$

These constraints will result in the branches being oriented the same way in both the primal and dual grid. Furthermore, the constraints guarantee that a closed polygon in the dual grid must be maintained. This is illustrated in Figure 2.9.



a) Node forces in form diagram

b) Nodal forces creates closed polygon

c) Force diagram made of closed polygons

Figure 2.9: Connection between nodal forces and equilibrium represented by closed polygons in the force diagram.

Figure 2.9 shows the relationship between the primal grid, closed polygons, and the dual grid. The problem here is that a single branch is only represented once in both the primal and dual grid and has but one direction. However, a single branch will have two opposite directions in the closed polygons for each node. Note that in Figure 2.9 (a) the branch is directed from node 2 to node 4. But in the closed polygon of node 2, the direction needs to be shifted. This is handled as follows:

$$-l_{be}^* + l_{bc}^* + l_{cf}^* + l_{fe}^* = 0 \tag{2.33}$$

or

$$\begin{cases} -u^*_{be} + u^*_{bc} + u^*_{cf} + u^*_{fe} = 0 \\ -v^*_{be} + v^*_{bc} + v^*_{cf} + v^*_{fe} = 0 \end{cases} \tag{2.34}$$

This needs to be done with all $n_i$ nodes. On matrix form this is sorted by the row space from the **C**-matrix, given as the transpose of the matrix. Since it only needs to be done with internal nodes (represented with $\mathbf{C}_i$) we get:

$$\begin{cases} \mathbf{C}_i^t \mathbf{u}^* = \mathbf{0} \\ \mathbf{C}_i^t \mathbf{v}^* = \mathbf{0} \end{cases} \tag{2.35}$$

Finally, to solve the optimisation, a cost vector needs to be defined. As given by Block (2009), the cost function aims to achieve the most equalised thrust network possible. The objective will be to minimise $t_i$, given as:

$$\min_{t_i} \qquad \sum t_i \qquad \text{with} \qquad t_i \geq \frac{d}{l_i} \tag{2.36}$$

Summarising, the second linear optimisation problem then becomes:

$$\min_{\mathbf{u}^*,\mathbf{v}^*,\mathbf{t}} \qquad \sum t_i \qquad \text{subject to} \qquad \begin{cases} \mathbf{I}_m \mathbf{u}^* - \mathbf{U}\mathbf{t} = \mathbf{0} \\ \mathbf{I}_m \mathbf{v}^* - \mathbf{V} = \mathbf{0} \\ \mathbf{C}_i^t \mathbf{u}^* = \mathbf{0} \\ \mathbf{C}_i^t \mathbf{v}^* = \mathbf{0} \\ \mathbf{L}^{-1}\mathbf{d} \leq \mathbf{t} \leq +\infty \end{cases} \tag{2.37}$$

where the bounds are given as:

$$\begin{array}{cc} \hline & \text{[lb,ub]} \\ \hline \mathbf{u}^* & [-\infty,\infty] \\ \mathbf{v}^* & [-\infty,\infty] \\ \mathbf{t} & [\mathrm{d/l}\,,\infty] \\ \hline \end{array} \tag{2.38}$$

The second linear optimisation problem will give $\mathbf{u}^*$ and $\mathbf{v}^*$, but the dual grid needs $\mathbf{x}^*$ and $\mathbf{y}^*$. Since the dual branch-node matrix cannot be inverted, this also needs to be solved as a linear problem. Starting with initial coordinates $x_0$ and $y_0$, we get:

$$\min_{x*} \qquad x_1^* \qquad \text{subject to} \qquad \begin{cases} \mathbf{C}^*\mathbf{x}^* = \mathbf{u}^* \\ x_1^* = x_0 \end{cases} \tag{2.39}$$

and

$$\min_{y*} \quad y_1^* \quad \text{subject to} \quad \begin{cases} \mathbf{C}^*\mathbf{y}^* = \mathbf{y}^* \\ y_1^* = y_0 \end{cases} \tag{2.40}$$

This will yield all the necessary coordinates to construct the dual grid.

The linear optimisation problem of (2.37) seems to be the bottleneck of the entire TNA system regarding computational time. Structural systems with more than ca 700 branches increase the computational cost dramatically, Block (2009). A solution to this problem is to use coarser grids with fewer branches or split the structure in smaller parts, which later merges together during a new optimisation process.

## 2.4 Free edges

The computational setup presented up to this point is only valid for systems where all the boundary edges are fixed. The opportunity to release these in order to create openings should also be included in the implementation phase. This also requires two additional steps in the process: 1) Ensuring correct shape of the primal grid; 2) expanding the C*-matrix.

As stated so far, only the internal lines of the grid are taken into account when creating the dual grid. With openings, the outer edge will also transfer forces, and need to be evaluated in the dual grid. However, the outer edge is not a node, meaning that a closed polygon will not ensure equilibrium. Block (2009) found that by enforcing the outer edge to have a funicular shape in the horizontal plane, then equilibrium is guaranteed. Furthermore, all branches connected to the free outer edge will have forces going to a single point. This is illustrated in Figure 2.10 and 2.11.

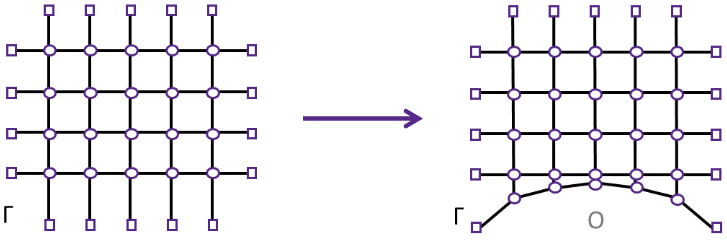


Figure 2.10: Introducing an open edge in the primal grid. The edge needs to be funicular in the plane.

The dual grid will then have a single point for each open edge. Illustrated with one edge in Figure 2.11.

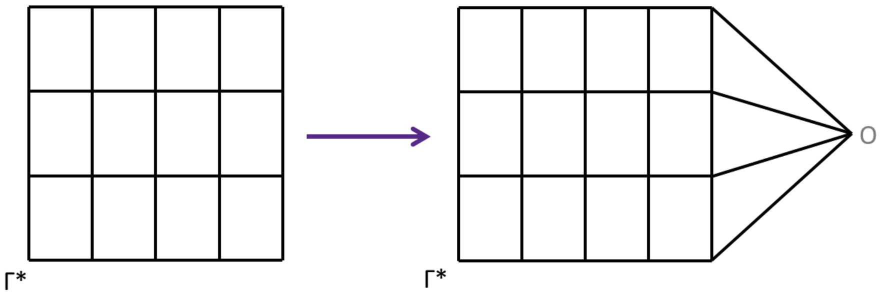


Figure 2.11: When introducing an open edge in the primal grid, the edge branches are connected to a single point in the dual grid.

For each free edge, the $\mathbf{C}^*$-matrix needs to be expanded by adding the new branches and their connectivity to the edge face.

## 2.5 Local and global stability

A stable equilibrium of a structural element, structure, or mechanical system is defined by Jones (2006) as a system where a small disturbance only results in a small response. While an unstable system which is exposed to a small disturbance will undergo a sudden change in deformation or displacement. The term "small" is a relative term in this case. This definition is illustrated in Figure 2.12, where the ball at the lower point is in a stable equilibrium, because it returns to its original position if exposed to a small disturbance. For the ball at the higher point, a small displacement will move the ball away from its original position. Hence the equilibrium is unstable.

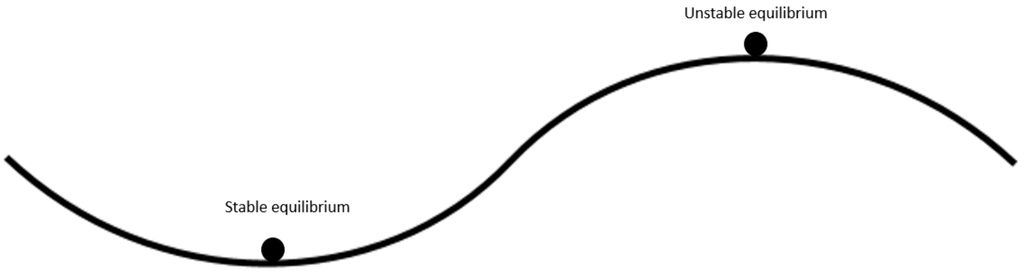


Figure 2.12: Visual explanation of stable and unstable equilibrium.

When a structural element or a structure starts deforming in an equilibrium shape different from the original equilibrium shape, the structure has buckled. Meaning that the curve of deformation changes direction. The most straightforward example of this ia a straight, axially loaded, beam element which is pinned at both ends, Figure 2.13. As the load increases, the beam is deformed, but initially remains straight. However, for a certain load, a slight disturbance will cause the beam to bend laterally into a new stable equilibrium, the buckled shape. The load that causes a stable element or structure to suddenly deform into a buckled shape, is called the buckling load, Jones (2006). The buckling shape may lead to second order forces, due to larger deformations, that may cause failure.

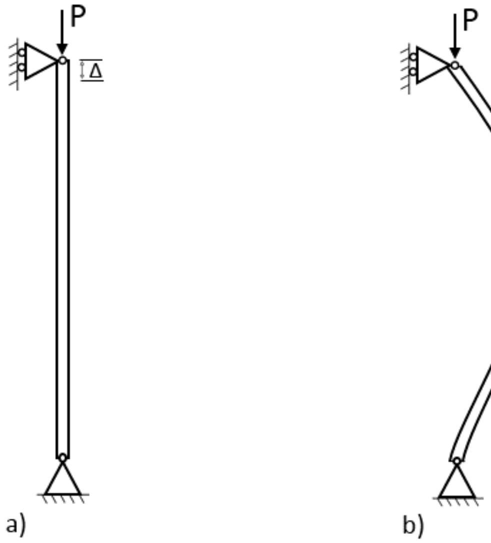


Figure 2.13: Axial loaded beam. In a) the beam element is in its original stable equilibrium. In b) the beam has buckled into a new stable equilibrium.

### 2.5.1 Local buckling

Local buckling is an instability problem limited to a single element in compression. Buckling is independent of the material strength. This means that a member exposed to compression forces may fail due to buckling even though the level of stresses are far from the material strength. As presented in Hibbeler (2014) the critical buckling load $P_{cr}$, also referred to as the *Euler load*, is given by:

$$P_{cr} = \frac{\pi^2 \times EI}{(K \times L)^2} \tag{2.41}$$

Where $E$ is the modulus of elasticity for the material and $I$ the moment of inertia about the weakest axis. $K$ is the effective length factor depending on the boundary conditions for both of the ends of the column. When both ends are pinned, as for the bar in Figure 2.13, $K$ is equal 1.0.

An important factor for the buckling analysis is the slenderness of the element. The slenderness ratio $\lambda$ is given by:

$$\lambda = \frac{L}{r} \tag{2.42}$$

Where $L$ is the length of the bar and $r$ is the radius of gyration which is given by:

$$r = \sqrt{\frac{I}{A}} \tag{2.43}$$

Elements with high slenderness ratios are more likely to buckle and are often defined as "slender columns". These members are likely to become unstable and buckle before the material capacity is reached. A rule of thumb in preliminary design is to use the ratio between height and smallest cross section width. For rectangular and circular columns with pinned connections, a ratio of 15-30 will most likely yield reasonable slenderness ratios, Orton (2013).

### 2.5.2 Finite Element Analysis of global stability

When the global stability of a system is evaluated, the contributions from all elements have to be established before they are introduced in a global system. This section presents the global stability problem for two-dimensional beam- and bar elements, Bell (2018).

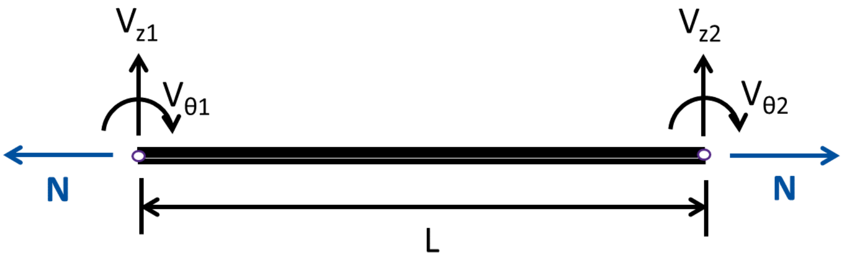


Figure 2.14: A two-noded beam element with 4 DOFs.

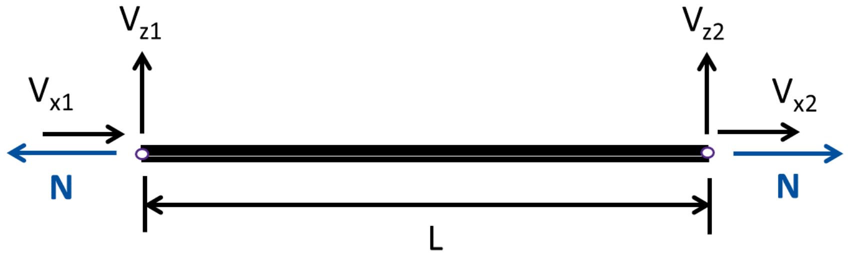


Figure 2.15: A two-noded bar element with 4 DOFs.

Starting with a single element, the forces **S** may be expressed using the relation between stiffness **k** and displacement **v**:

$$\boldsymbol{S} = \boldsymbol{k} * \boldsymbol{v} \tag{2.44}$$

From this the global system may be established from the contribution of each element:

$$\boldsymbol{R} = \boldsymbol{K} * \boldsymbol{r} \tag{2.45}$$

Starting with the beam element in Figure 2.14 with four degrees of freedom and contribution from second order stiffness, equation 2.45 may be expressed as follows:

$$\mathbf{S} = \begin{bmatrix} S_{z1} \\ S_{\theta 1} \\ S_{z2} \\ S_{\theta 2} \end{bmatrix} = \frac{2EI}{L^3} \begin{bmatrix} 6 & -3L & -6 & -3L \\ -3L & 2L^2 & 3L & L^2 \\ -6 & 3L & 6 & 3L \\ -3L & L^2 & 3L & 2L^2 \end{bmatrix} + \tag{2.46}$$

$$\frac{N}{30L} \begin{bmatrix} 36 & -3L & -36 & -3L \\ -3L & 4L^2 & 3L & -L^2 \\ -36 & 3L & 36 & 3L \\ -3L & -L^2 & 3L & 4L^2 \end{bmatrix} * \begin{bmatrix} v_{z1} \\ v_{\theta 1} \\ v_{z2} \\ v_{\theta 2} \end{bmatrix} = (\mathbf{k}_1 + \mathbf{k}_G)\, v = \mathbf{k}_2 \mathbf{v} \tag{2.47}$$

Before adding the element contribution to the global system, a transformation matrix has to be established:

$$\mathbf{v}_i = \begin{bmatrix} v_{zi} \\ v_{\theta i} \end{bmatrix} = \begin{bmatrix} -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \overline{v}_{xi} \\ \overline{v}_{zi} \\ \overline{v}_{\theta i} \end{bmatrix} = \mathbf{t}_r \overline{\mathbf{v}}_i \tag{2.48}$$

where c = cos $\phi$ and s = sin $\phi$.

The transformation matrix from local to global system is given as:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \mathbf{t}_r & 0 \\ 0 & \mathbf{t}_r \end{bmatrix} * \begin{bmatrix} \overline{v}_1 \\ \overline{v}_2 \end{bmatrix} = \mathbf{T}_r \overline{\mathbf{v}} \tag{2.49}$$

The stiffness matrices for the element is transformed using this equation:

$$\overline{\mathbf{k}} = \mathbf{T}_r^T \mathbf{k} \mathbf{T}_r \tag{2.50}$$

According to 2.order theory, the stiffness matrices for each element in global coordinates are established by using (2.50):

$$\overline{\mathbf{k}}_1 = \frac{2EI}{L^3} \begin{bmatrix} 6s^2 & -6cs & 3Ls & | & -6s^2 & 6cs & 3Ls \\ -6cs & 6c^2 & -3Lc & | & 6cs & -6c^2 & -3Lc \\ 3Ls & -3Lc & 2L^2 & | & -3Ls & 3Lc & L^2 \\ - & - & - & - & - & - & - \\ -6s^2 & 6cs & -3Ls & | & 6s^2 & -6cs & -3Ls \\ 6cs & -6c^2 & 3Lc & | & -6cs & 6c^2 & 3Lc \\ 3Ls & -3Lc & L^2 & | & -3Ls & 3Lc & 2L^2 \end{bmatrix} \tag{2.51}$$

$$\overline{\mathbf{k}}_g = \frac{N}{30L} \begin{bmatrix} 36s^2 & -36cs & 3Ls & -36s^2 & 36cs & 3Ls \\ -36cs & 36c^2 & -3Lc & 36cs & -36c^2 & -3Lc \\ 3Ls & -3Lc & 4L^2 & -3Ls & 3Lc & -L^2 \\ -36s^2 & 36cs & -3Ls & 36s^2 & -36cs & -3Ls \\ 36cs & -36c^2 & 3Lc & -36cs & 36c^2 & 3Lc \\ 3Ls & -3Lc & -L^2 & -3Ls & 3Lc & 4L^2 \end{bmatrix} \tag{2.52}$$

A similar procedure is established for the bar element with two degrees of freedom in Figure 2.15:

$$\mathbf{S} = \begin{bmatrix} S_{x1} \\ S_{z1} \\ S_{x2} \\ S_{z2} \end{bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \frac{N}{L} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} v_{x1} \\ v_{z1} \\ v_{x2} \\ v_{z2} \end{bmatrix} = (\mathbf{k}_1 + \mathbf{k}_G)\mathbf{v} = \mathbf{k}_2 \mathbf{v} \tag{2.53}$$

The connection between the local and global translations is given as:

$$\mathbf{v}_i = \begin{bmatrix} v_{xi} \\ v_{zi} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} * \begin{bmatrix} \overline{v}_{xi} \\ \overline{v}_{zi} \end{bmatrix} = \mathbf{t}_r \overline{\mathbf{v}} i \tag{2.54}$$

where c = cos $\phi$ and s = sin $\phi$.

Using (2.50), the stiffness relation after 2.order theory for the bar element with global coordinates is then given as:

$$\overline{\mathbf{S}} = \begin{bmatrix} \overline{S}_{x1} \\ \overline{S}_{z1} \\ \overline{S}_{x2} \\ \overline{S}_{z2} \end{bmatrix} = \frac{EA}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} + \tag{2.55}$$

$$\frac{N}{L} \begin{bmatrix} s^2 & -cs & -s^2 & cs \\ -cs & c^2 & cs & -c^2 \\ -s^2 & cs & s^2 & -cs \\ cs & -c^2 & -cs & c^2 \end{bmatrix} * \begin{bmatrix} \overline{v}_{x1} \\ \overline{v}_{z1} \\ \overline{v}_{x2} \\ \overline{v}_{z2} \end{bmatrix} = (\overline{\mathbf{k}}_1 + \overline{\mathbf{k}}_G)\overline{\mathbf{v}} = \overline{\mathbf{k}}_2 \overline{\mathbf{v}} \tag{2.56}$$

Buckling denotes instability, which is characterised by the system going from being stable to becoming unstable at so-called bifurcation or branch points on the equilibrium curve. This happens when the resulting stiffness matrix $\boldsymbol{K}_2$ becomes singular,

$$det(\boldsymbol{K}_2) = det(\boldsymbol{K}_1 + \boldsymbol{K}_g) = 0 \tag{2.57}$$

A physical interpretation is that buckling happens when the compressive axial forces reach such a level that the structure becomes unstable without applying any outer forces:

$$(\boldsymbol{K}_1 + \boldsymbol{K}_g) * \boldsymbol{r} = 0 \tag{2.58}$$

This problem can be solved by using the eigenvalue problem. Since $\mathbf{K}_g$ is a function of N, the value of normal forces which causes the structure to buckle may be established. The eigenvalue is expressed as:

$$(\mathbf{K}_1 - \lambda_i * \mathbf{K}_g) * \phi_i = 0 \tag{2.59}$$

where $\lambda_i$ is an eigenvalue and $\phi_i$ is the corresponding eigenvector (buckling shape). The lowest value of $\lambda$ represent the buckling load, since $\lambda_i$ is the factor to scale the axial forces in the system to make the structure buckle.

$$\mathbf{R}_{kr} = \lambda_i * \mathbf{R} \tag{2.60}$$

As shown with eq. (2.54) the bar element can only represent buckling modes with pure translation in two directions (here x and z), but the computational time will be faster than the beam element due to the fewer degrees of freedom. The beam element is able to represent both translation and rotations in the nodes (eq.(2.48)), but at the cost of increased computational time.

# 3 Implementation of TNA in 2D

This chapter contains the implementation of TNA in 2D. Two components are developed in Visual Studio using the C# language. The solver, which implements the TNA, and a verification component that controls the utilisation and local buckling of the elements in the final geometry.

In order to verify the components, and investigate the effect of different construction materials, two examples are carried out at the end of the chapter. Renzo Piano's Chiesa de Padre Pio, with stone arches is first recreated before a conceptual timber bridge is designed.

## 3.1 Assumptions and simplifications

The systems are assumed to be simply supported at both end nodes. Only vertical loads are considered. These are lumped into the free nodes as both self weight and external loads. All external loads are assumed to be of equal magnitude and present in all the free nodes.

The relationship between the length of the form and force branches is given as:

$$L_{H,i} = \frac{1}{r} L^*_{H,i} \tag{3.1}$$

Where $\frac{1}{r}$ is the scale factor, $\zeta$, which defines the relationship between the form an force diagrams. For the 2D solution, presented in this chapter, the ratio r, which in theory is initially unknown, can be changed by the user to quickly generate several shapes in the preliminary design.

The branches $L^*_{H,i}$ in the force diagram for the 2D case only consist of straight lines with origin in the same point $O$. In order for the polygons of the force diagram to be closed, and the structure consequently in equilibrium, the vertical forces from the self weight and nodal forces must thus be included.

## 3.2 Implementation

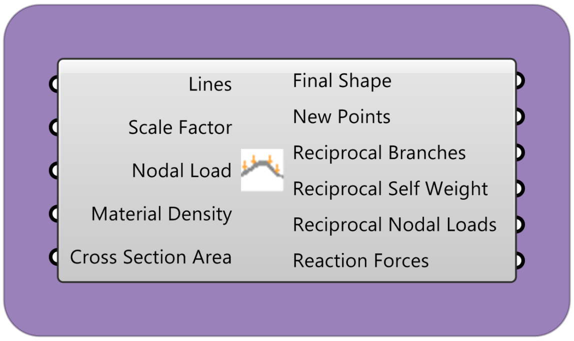


Figure 3.1: Component *TNA2D Solver*.

**Input**

- Lines: List of lines stretching between two support points.
- Scale Factor: Scale factor $r$.
- Nodal Load: Load in distributed to all nodes, [N].
- Material Density: Double, [kg/m$^3$].
- Cross Section Area Density: Double, [mm$^2$].

**Output**

- Final Shape: List of lines forming the new shape.
- New Points: List of nodes forming the new shape.
- Reciprocal Branches: List of the branch forces in the reciprocal diagram, [N].
- Reciprocal Self Weight: List of self weight lines in the reciprocal diagram.
- Reciprocal Nodal Loads: List of the nodal forces in the reciprocal diagram, [N].
- Reaction Forces: List of horizontal and vertical reaction forces at the two support points, [N].

### Calculations

Initially the *TNA2D Solver* component, Figure 3.1, finds the start point and end points from the given input lines before sorting these into free and fixed nodes based on the assumptions in section 3.1 that all end nodes are fixed. Following this the C-matrix can be found. Both the initial coordinates of the points as well as the external nodal loads are known from the input.

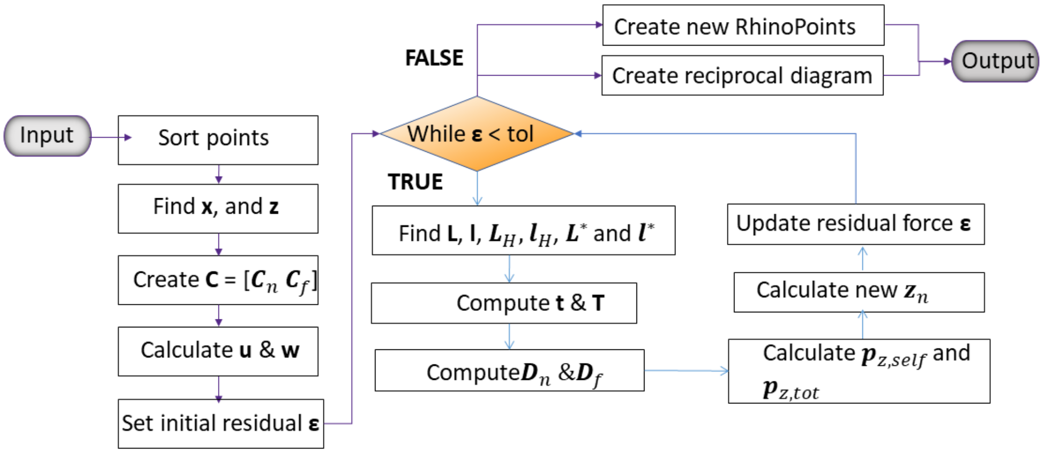

Input
Sort points
Find **x**, and **z**
Create **C** = [$\boldsymbol{C}_n$ $\boldsymbol{C}_f$]
Calculate **u** & **w**
Set initial residual **ε**
While **ε** < tol
FALSE
TRUE
Create new RhinoPoints
Create reciprocal diagram
Output
Find **L**, **l**, $\boldsymbol{L}_H$, $\boldsymbol{l}_H$, $\boldsymbol{L}^*$ and $\boldsymbol{l}^*$
Compute **t** & **T**
Compute$\boldsymbol{D}_n$ &$\boldsymbol{D}_f$
Calculate $\boldsymbol{p}_{z,self}$ and $\boldsymbol{p}_{z,tot}$
Calculate new $\boldsymbol{z}_n$
Update residual force **ε**

Figure 3.2: Pseudo-code for the *TNA2D Solver* component showing the calculation procedure.

Self weight is accounted for by lumping the mass from the branches onto the free nodes by the assumption that each node gets half the weight from each coinciding bar. For a situation where all the bards initially are aligned horizontally with equal lengths, the loads will be equal in all free nodes. After one iteration of the *TNA2D Solver*, all the bars will get a new length, in addition to a new self weight. To account for this, an iterative solver of the problem is implemented where the new lengths and loads are updated before a new calculation is done. As shown by Adriaenssens et al. (2014) this self weight may be presented on a matrix form as:

$$\mathbf{p}_y = \frac{1}{2}\rho g A|\mathbf{C}_n|^T\mathbf{l}, \tag{3.2}$$

where $\rho$ is the material density [kg/m$^3$], $g$ is the gravitational constant [m/s$^2$], and $A$ the cross sectional area of the bars [m$^2$]. In 2D the cross sectional area and material density will be direct user input to the component as shown in Figure 3.1.

Residual forces between the new and old self weight are used as a criteria for closing the loop. This is initially set to be 0.01, but may be further specified by the user. The pseudo-code showing the flow of calculations is illustrated in Figure 3.2.

**Calculations of reciprocal diagram**

For the design of the reciprocal diagram the theory states that the force- and form-branches should be parallel to each other, with the relation given in eq.2.17. This is based on the assumption that in 3D where all external loads act vertically, and will, therefore, not appear in the force diagram. For a 2D case, however, the vertical loads are needed in order to get a closed polygon.

Figure 3.3 (a) and (b) illustrates this for a structure where the only input difference is the scale factor of $r = 0.5$ and $r = 1.0$ respectively. From this it may be shown that the form and force branches will only be parallel in a situation where $r = 1$.

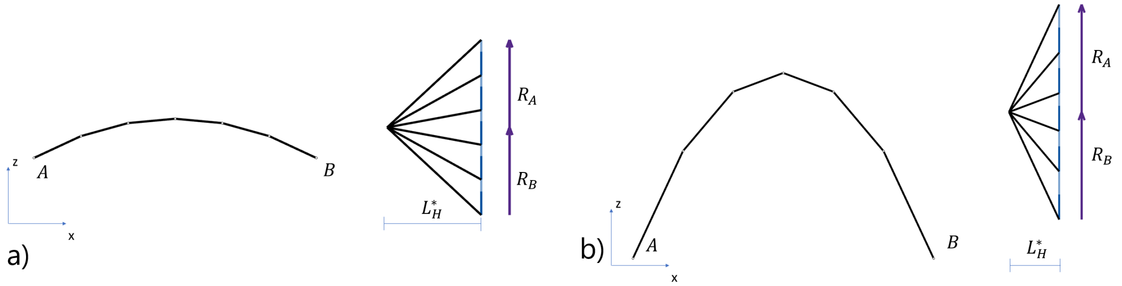


Figure 3.3: Figure a) shows the solution from a TNA with the scale factor $r = 0.5$ and Figure b) where $r = 1.0$. This clearly shows that for equilibrium to be maintained while the external loads are constant, the form and force branches will no longer be parallel.

The reason for the non-parallel branches is from the assumption of the horizontal length of the branches in the force diagram being proportional to the scale factor $r$ in equation 3.1. For the self weight this proportionality is reciprocal as the force depends on the length of the branches in the form diagram. The nodal loads, on the other hand, are completely independent of the force diagram. In practice, this means that the only way to close the force diagram is to increase the length of force branches when the nodal loads, self weight, and horizontal branch length are given.

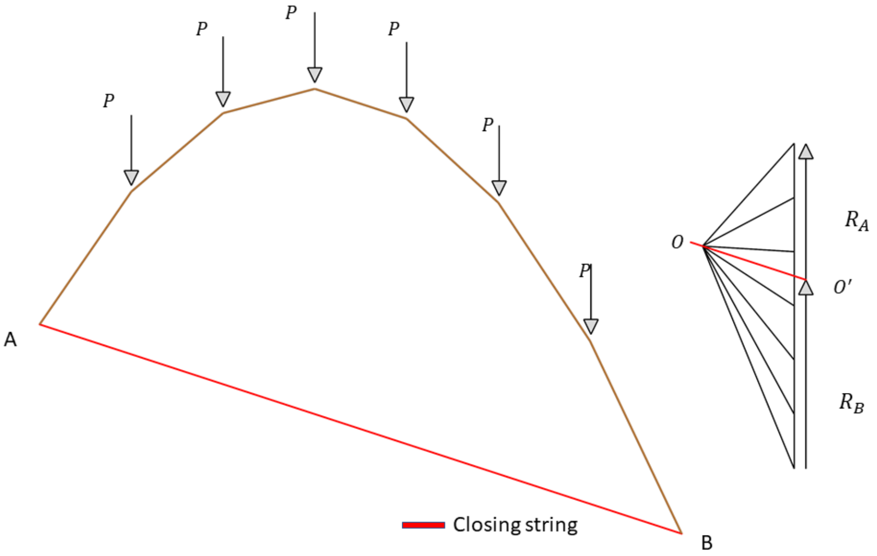


Figure 3.4: The closing string in red in both the primal and the dual grid. The closed polygons above and beyond the O-O' line equal the force polygon at each support.

For the calculations of the force diagram, the closing string between the two supports are first found. Using the theory presented by Kindem (1979), a closing string between the two supports is drawn. The inclination of this line will be equal to the O-O' line in the reciprocal diagram. When all the external forces act vertically the horizontal forces will be equal in all the branches.

The length of the vertical lines in the reciprocal diagram above the O’ point represents the reaction forces $R_A$, while the length beneath equals $R_B$, Figure. 3.4. These support forces are found by calculating the moment equilibrium about one support.

From this information the reciprocal diagram is created. The highest point of the vertical lines in the force diagram will be the coordinates of the O’-point with an additional height equal to the left support forces $\mathrm{R_A}$. From here all the force lines from vertical loads are drawn head to tail downwards. Finally, the polygons are closed by drawing the branch forces from the O-point to the tails of each total nodal load. The pseudo-code for this process is shown in Figure 3.5.

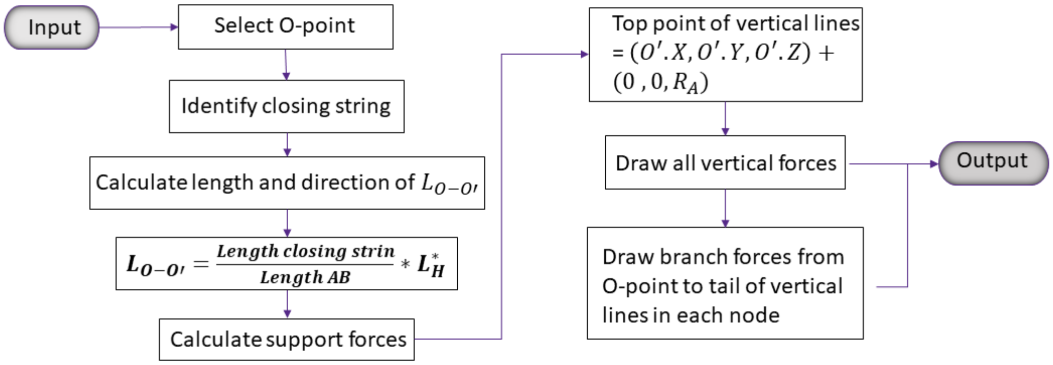

Input
Select O-point
Identify closing string
Calculate length and direction of $L_{O-O'}$
$L_{O-O'} = \frac{Length\ closing\ strin}{Length\ AB} * L_H^*$
Calculate support forces
Top point of vertical lines = $(O'.X, O'.Y, O'.Z) + (0\ , 0, R_A)$
Draw all vertical forces
Draw branch forces from O-point to tail of vertical lines in each node
Output

Figure 3.5: The pseudo-code used for generating the reciprocal force diagram in 2D.

### Visualisation and interpretation

In order to give the user a visual interpretation of the results, a component which calculates the utilisation ratio of normal stresses in all branches is created. The component is shown in Figure 3.6. A colour map representing the bars which have a utilisation ratio above 0.85 as yellow and above 1.0 as red, gives the user an overview of the structure's efficiency.

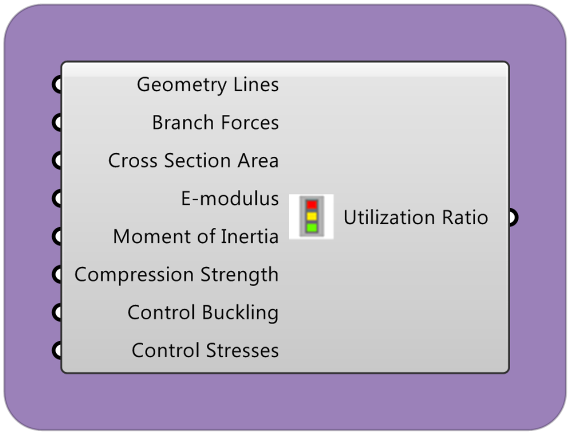


Figure 3.6: Component *Capacity Control 2D*.

**Input**

- Geometry Lines: List of lines representing the new shape.
- Branch forces: List of lines from the reciprocal diagram.
- Cross sectional area: Double, [$\text{mm}^2$].
- E-modulus: Double, [$\text{N/mm}^2$].
- Moment of inertia: Double, [$\text{mm}^4$].
- Compression strength: Double, [$\text{N/mm}^2$].
- Control Buckling: Boolean, true if the buckling should be controlled.
- Control stresses: Boolean, true if the stresses should be controlled.

**Output**

- Utilisation ratio: Colour map of the structure showing utilisation of the structure.

## 3.3 Verification

In order to verify the results from the *TNA2D Solver*, two examples where the solution is known will be evaluated. Starting with a classic inverted hanging chain loaded only by point loads, before moving on to verify the iterative solution of the self weight.

### 3.3.1 Simple inverted hanging chain

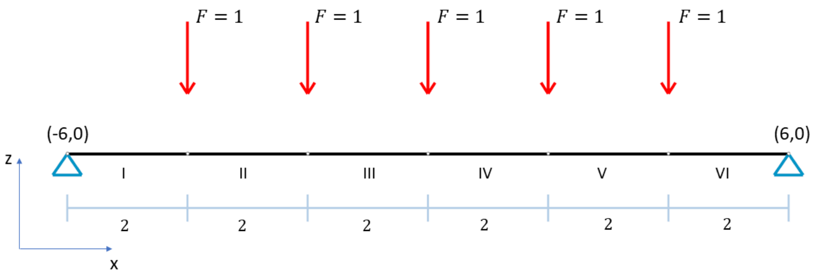


Figure 3.7: Geometry for simple inverted chain example with loading and geometry displayed.

The *TNA2D Solver* component is first tested by comparing it with a simple hanging chain presented in Adriaenssens et al. (2014), p. 117. In order for the component to match the results from the book, a small modification is made by neglecting the self weight of the system, so that only the nodal loads are accounted for. Furthermore the scale factor is set to $r = 1$, meaning $\boldsymbol{L}_H = \boldsymbol{L}_H^*$.

$$\boldsymbol{C} = \left[\begin{array}{c|c} \boldsymbol{C_N} & \boldsymbol{C_F} \end{array}\right] \tag{3.3}$$

For the example in Figure 3.7, the nodal loads in the free nodes are assumed to be equal to 1 in the vertical direction. The simple chain has the following branch node matrix:

$$\boldsymbol{C} = \left[\begin{array}{ccccc|cc} 1 & \cdot & \cdot & \cdot & \cdot & -1 & \cdot \\ 1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & -1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & -1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & -1 \end{array}\right] \tag{3.4}$$

with the coordinate vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ sorted in free and fixed nodes expressed as:

$$\boldsymbol{x} = \left[\begin{array}{c|c} \boldsymbol{x_N} & \boldsymbol{x_F} \end{array}\right] = \left[\begin{array}{ccccc|cc} -4 & -2 & 0 & 2 & 4 & -6 & 6 \end{array}\right]^T, \tag{3.5}$$

$$\boldsymbol{z} = \left[\begin{array}{c|c} \boldsymbol{z_N} & \boldsymbol{z_F} \end{array}\right]^T = \left[\begin{array}{ccccc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]^T \tag{3.6}$$

From this, the coordinate differences $\boldsymbol{u}$ and $\boldsymbol{w}$ may be calculated as:

$$\boldsymbol{u} = \boldsymbol{C}\boldsymbol{x} = \begin{bmatrix} 2 & -2 & -2 & -2 & -2 & -2 \end{bmatrix}^T \tag{3.7}$$

$$\boldsymbol{w} = \boldsymbol{C}\boldsymbol{z} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{3.8}$$

The $m\ x\ m$ matrices $\boldsymbol{U}$ and $\boldsymbol{W}$ belonging to vectors $\boldsymbol{u}$ and $\boldsymbol{w}$ give us the branch lengths $\boldsymbol{L}$ from equation 2.10. For the 2D case this may be expressed as

$$\boldsymbol{L} = \sqrt{\boldsymbol{U}^2 + \boldsymbol{W}^2} \tag{3.9}$$

with the length vector $\boldsymbol{l}$ from the diagonal of $\boldsymbol{L}$

$$\boldsymbol{l} = \begin{bmatrix} 2 & -2 & -2 & -2 & -2 & -2 \end{bmatrix} \tag{3.10}$$

From the information above, all the necessary geometry and relations needed to perform the 2D TNA analysis are given.

By inserting the now known data from Figure 3.7 into eq.(2.24) and eq.(2.26), the new z-coordinates of the system become:

$$\boldsymbol{z}_N = \begin{bmatrix} 2.5 & 4 & 4.5 & 4 & 2.5 \end{bmatrix} \tag{3.11}$$

The $\boldsymbol{x}$ vector remains unchanged through the whole process as the earlier assumption of the horizontal lengths being fixed and constant is still valid. The final funicular shape of the system is shown in Figure 3.8.

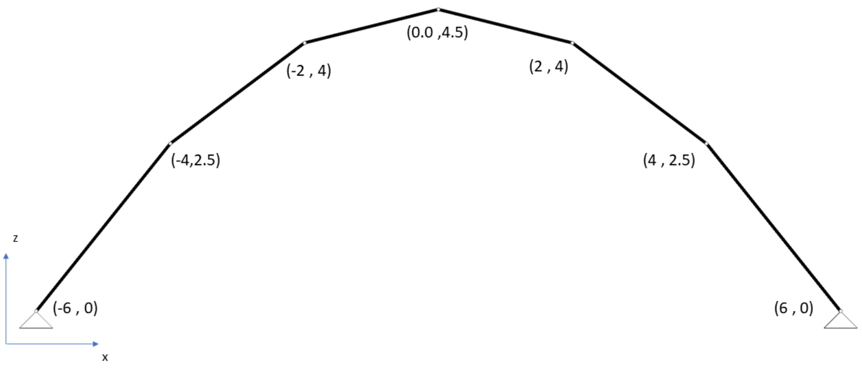


Figure 3.8: The results from the *Camel* component *TNA2D Solver*. When self weight are neglected the results match the theoretical ones from (3.11).

Calculations from the component in Figure 3.1 are identical to the theoretical ones presented in this section, and the resulting shape is parabolic. A practical interpretation of this may be a situation where the nodal loads are significantly greater than the self weight. One example of this may be a wooden arch bridge carrying a horizontal bridge deck through vertical ties.

### 3.3.2 Catenary from self weight

According to the theory Carlson (2015), a chain only loaded by its self weight should obtain the shape of a catenary given by the equation:

$$y = a * cosh(\frac{x}{a}) \tag{3.12}$$

where $a$ is a constant and $cosh$ is a hyperbolic function. The *TNA2D Solver* is tested with the catenary as a target shape when only self weight is present. The component is fed with data for the input as shown in fig 3.9. The scale factor is set to 1.0, and no nodal loads are added. The material is assumed to be concrete with a density of 2400 kg/m$^3$. A cross sectional area of 30 000 mm$^2$ (i.e. 300 mm x 100 mm) is chosen. The horizontal length between the supports is set to 12 000 mm.

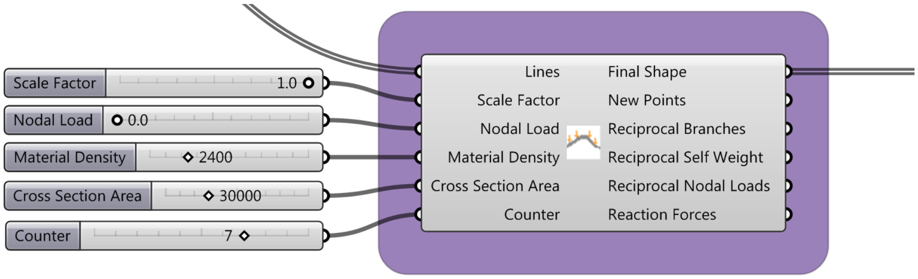

Scale Factor 1.0
Nodal Load 0.0
Material Density 2400
Cross Section Area 30000
Counter 7

Figure 3.9: Component used to test the self weight. Input parameter "counter" is only used for the purpose of illustrating the effect of several iterations, and for this reason not a part of the final component.

The shapes calculated by the component are compared to a catenary shape drawn directly with a function in Rhino, as shown in Figure 3.10. With an increasing number of iterations it becomes clear that the solution converges towards a catenary shape. Together with the first verification, the created component seems to handle both nodal loads and self weight.

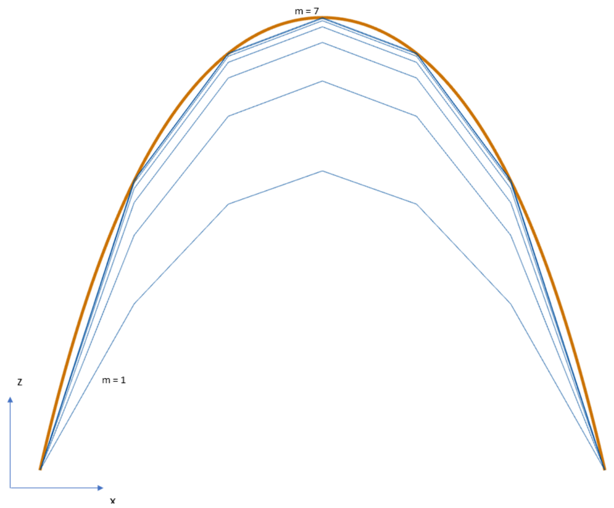


Figure 3.10: Development of shape from seven iterations of the *TNA2D Solver* component. This shows how the form converges towards a catenary shape illustrated by the brown line.

## 3.4 Test case: Chiesa di Padre Pio

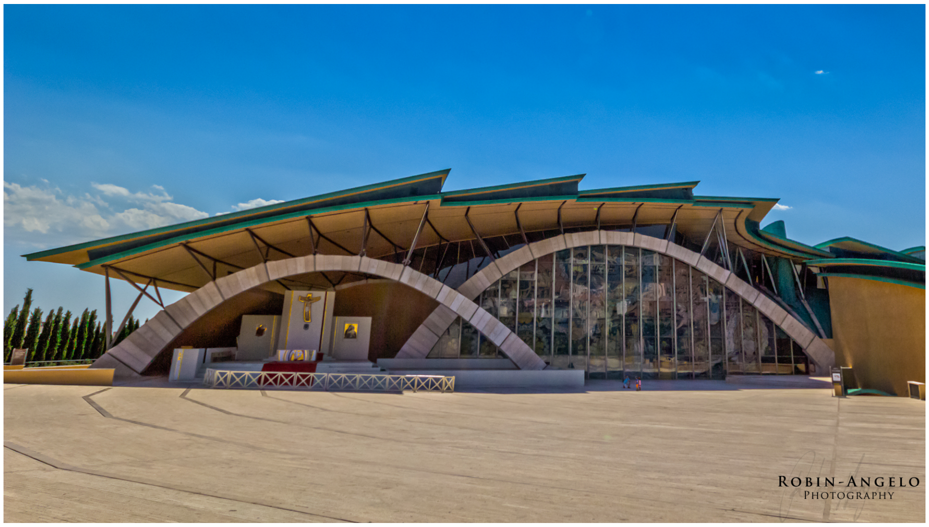


Figure 3.11: The Church of Padre Pio. The whole building consists of a series of stone arches arranged in two half-circles around a centre point. Image from: Robin-Angelo (2012).

To test the functionality of the components when both nodal load and self weight are present, a case study is performed on Renzo Piano Building workshop's church from 2004, Chiesa di Padre Pio.

The church is located in the east of Italy, and was constructed due to an increasing amount of pilgrimage demanding a bigger capacity. It is recognised by its vast open spaces and atmosphere. It consists of a total of 21 monumental stone arches divided into two three-quarter circles, where the outer circle represents a scaled-down version of the inner one. The loads from the roof, which are made of copper, are transferred to the arches via steel struts connected together in discrete points, ArchDaily (2018). This resembles the assumptions and basis of analysis presented in this section, and makes it a fitting case study for the developed components.

### 3.4.1 Specifications

The material used for the arches is a local Apulian limestone typical for the region. The density for limestone is approximately 2700 kg/m$^3$. The copper roof has a density of 8960 kg/m$^3$. The weight of the struts connecting the roof and arch is initially neglected.

With a span of 50 metres, and height of 16 metres, the largest arches will be used as a benchmark for the 2D components.

First, an estimate of the loading applied to each arch from the roof is made. Copper plates for roofing are usually just a couple of millimetre thick, and from studying images some sort of wooden plates to support them. An assumption of 1 kg/m$^2$ as the total roof load is considered to be appropriate.

Since the TNA framework neglects horizontal load, wind loads are not considered here. Snow loads are also neglected based on the geographical location of the building.

When assuming the self weight of the arch, the size of the bricks is estimated to have a constant cross-section of 1400 mm x 700 mm. The roof load is transferred via 12 equal-sized nodal loads with a size of 62.5 kN, and the self weight is evenly distributed at 26.46 kN/m in the vertical direction.

### 3.4.2 Designing the arch

#### TNA in Grasshopper

The script in Grasshopper starts by creating two points with a distance of 50 metres. From these points a line is drawn, and then subdivided into 13 straight line segments of equal lengths. The input obtained from the initial estimates are then specified, as shown in Table 3.1. With the necessary input given, the objective will be to find the scale factor $r$ for which the height is roughly 16 metre. The scale factor $r = 0.06612$ is found to achieve this. Next, the forces should be read directly from the reciprocal diagram. As explained in section 3.2, the solver component yields three different parts from the reciprocal diagram, as shown in Figure 3.1 . The length of these branches corresponds to the forces in Newton. For the first and last branch near the supports, which have the largest forces, the values from the TNA are $1097.4kN$, resulting in compressive stresses around 1.12 N/mm$^2$. This is expected to be somewhat lower than the results expected from the FEA. There are two main reasons for this.

Firstly, is the way the TNA calculates the self weight by lumping it onto nodes as explained in section 3.1. As a result, half of the weight from the first and last branch is transported directly to the supports, meaning that some of the self weight disappears from the calculations. For a course division of branches such as the one used here, this will cause a significant discrepancy between the TNA and the initial estimates.

Table 3.1: The input parameters for TNA-analysis.

| Properties | Value |
|---|---|
| Arch span [mm] | 50 000 |
| Line segments | 13 |
| Cross section [mm$^2$] | 700 x 1400 |
| Nodal Load [kN] | 62.5 |
| Self weight [kN/m] | 26.46 |
| Material Density [kg/m$^3$] | 2700 |
| Scale factor $r$ | 0.06612 |

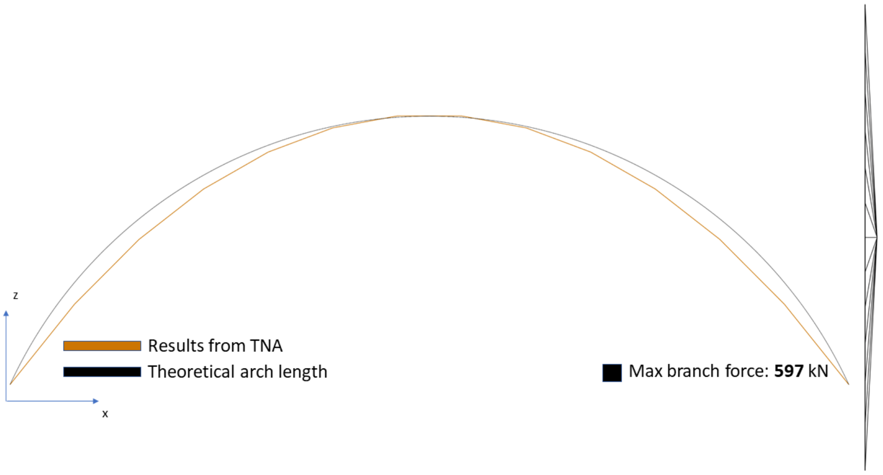


Figure 3.12: The resulting shape from the *TNA2D Solver* (brown) compared to the theoretical arch used for the hand calculations (black). To the right the reciprocal diagram is shown. This is scaled to fit the Figure. It clearly shows how the vertical component of the forces are significantly larger than the horizontal one.

Secondly, the total length of the TNA-arch is 1 m shorter than the theoretical one, giving another reduction in the arch's self weight compared to the hand calculations in Figure 3.12. As the number of branches increase, these discrepancies will diminish.

Overall, the solution seems to be slightly non-conservative, but given the early stage of the design it still yields a reasonable result, that may be further reviewed in an FEA in order to verify if it is a funicular shape.

**Evaluation with finite element analysis**

From the thrust lines obtained with the TNA, an FEA in the *Karamba* plug-in is performed to verify the results. Materials, cross sections, and supports are then defined before a first order analysis is performed. No load factors are used for this. Results from this analysis and the TNA are presented in Table 3.2 along with the initial estimate.

Since beam elements are used to analyse the arch, the arch's self weight is lumped into the nodes in order to avoid the particular solution coming from the evenly distributed self weight between the nodes. The resulting force diagrams from this analysis is shown on the left side in Figure 3.13. With the self weight as lumped nodes, the use of only 13 elements for a 60 metre long arch is a very coarse assumption, especially since the weight of the stone is a substantial part of the total load. In order to achieve more accurate results, each element is further subdivided into 10 elements, resulting in a total of 130 elements. The self weight remains lumped, but divided into more nodes, while the roof loads work in the same 13 nodes as before. This yields the force diagrams on the right side in Figure 3.13.

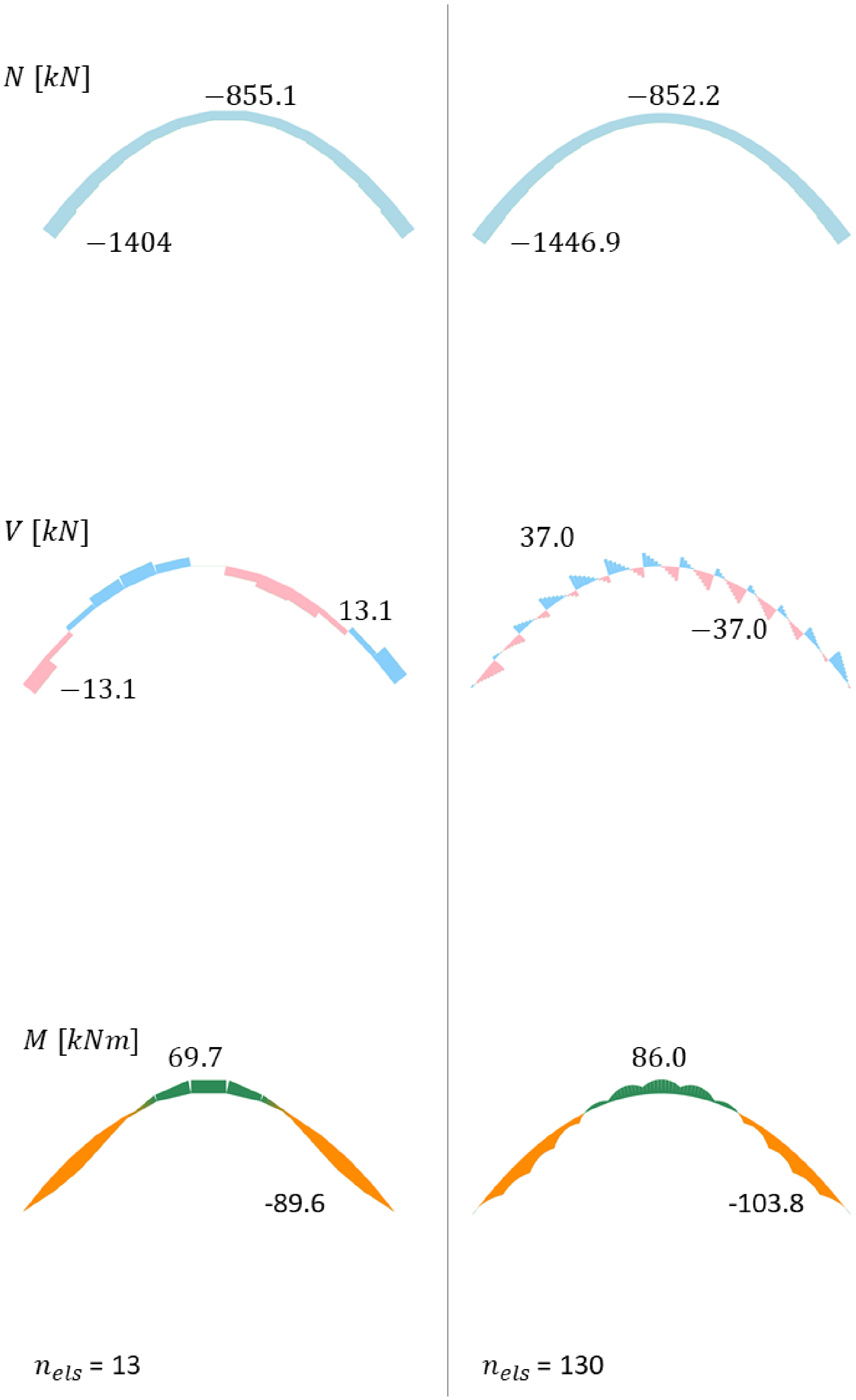


Figure 3.13: Results from FEA with *Karamba*. Left side shows the arch modelled with 13 elements and the right side with 130 elements. Both analyses lump the contribution from self weight into the nodes.

Table 3.2: Results from the TNA and the FEA performed in this section. Note that the total axial stresses from the FEA are obtained by summing the maximum contributions from the normal forces and moments, which is a conservative solution since they are not present at the same location. Negative axial forces indicates compression

| | TNA 2D | FEA-Karamba |
|---|---|---|
| $R_V$ [kN] | 1096.9 | 1169.3 |
| $R_H$ [kN] | 58.25 kN | 852.21 |
| $N_{Max}$ [kN] | -1097.4 | -1446.9 |
| $V_{Max}$ [kN] | - | 37.0 |
| $M_{Max}$ [kNm] | - | 103.8 |
| $\sigma_{\text{N.x}}$ [N/mm$^2$] | 1.12 | 1.48 |
| $\sigma_{\text{M.x}}$ [N/mm$^2$] | - | 0.91 |
| $\sigma_{\text{x.Tot}}$ [N/mm$^2$] | 1.12 | 2.39 |
| UR [%] | 3.73 | 10.1 |

When comparing the TNA to the FEA there are some important differences one should notice. Starting with the reaction forces at the supports, the vertical forces is fairly similar in magnitude. The horizontal force however, has a discrepancy of 794 kN between the TNA and the FEA. This causes the maximum force of the TNA to be 24 % smaller than the FEA. Due to the fact that the TNA solution only has two branch forces and one vertical load in each node, the system is statically determined. The assumptions made in eq. (3.1) "fixes" the horizontal forces for the *TNA2D Solver* in order to get equilibrium at each node; meaning that the original element size has a great influence on the final horizontal forces. Hence a greater number of elements will reduce the horizontal forces in the system. Consequently, the number of elements in the analysis has a large impact on the final solution. Furthermore, the horizontal force remains constant throughout the whole arch (see reciprocal diagram in Figure 3.12).

Results from the FEA also show that the original shape created with the TNA produces some bending moment. The greatest difference might be the horizontal reaction forces, which in the FEA are considerably larger than those from the TNA. Resulting in both an increase in the normal forces, as well as the presence of moments with a magnitude of 103.8 kNm.

When looking at the difference in utilisation and axial stresses for the two solutions, a compressive strength of 30 MPa is assumed, MatWeb (2020). Starting with the TNA, only normal forces are present, the normal stresses have a maximum value of 1.12 N/mm$^2$. This gives an utilisation ratio of only 3.7 %. The FEA has both a higher maximum normal force, as well as bending moments. The respective stresses are 1.48 N/mm$^2$ and 0.91 N/mm$^2$, resulting in a total axial stress of 2.39 N/mm$^2$. The utilisation then becomes 8 %, which is caused by the bending moments and larger normal forces in the FEA. Even though the total stresses in the arch are only compressive throughout the arch, the difference between the TNA and FEA is larger than desired.

### 3.4.3 Summary

In this section it was shown that the TNA component created an arch which had a larger presence of bending moments than expected. With the case aiming to recreate an existing structure, the height to span ratio was fixed and equal to 0.32. The scale factor $r$ is controlling the magnitude of horizontal forces. A smaller scale factor yields a lower arch and a larger horizontal force. As the arch gets too low it will start to behave more like a beam, due to the introduction of bending moments. Based on the results, this ratio should be increased for the arch to become funicular. This relationship will be further investigated in the next case study.

The case study is concluded with the following remarks:

- The *TNA2D Solver* is able to recreate the shape of the stone arches.
- Some bending moments are present, this is most likely due to the low height of the arch.
- The FEA yields an arch in compression throughout the entire structure.
- The height to span ratio's effect on the funicular solutions should be further investigated in the next section.

## 3.5 Test case: Timber arch bridge

In addition to the Chiesa di Padre Pio, another case with an arch in compression will be presented in this chapter. The arches for the Padre Pio Church had a loading situation where the self weight had a big impact on the total load, due to the heavy material. For a timber structure, the variable loads are assumed to be the driving aspect in design. Instability problems like buckling may be more relevant for this kind of material and structures as well.

For this case the objective is to use the components as an early stage design tool for creating a plausible shape for the glue laminated timber arches in Figure 3.14. Spanning over a road beneath it, two timber arches carry the bridge deck through six tension bars, hence resembling the way nodal loads are applied in the *TNA2D Solver*. After establishing a shape and recommending a cross section, the results will be verified by an FEA.

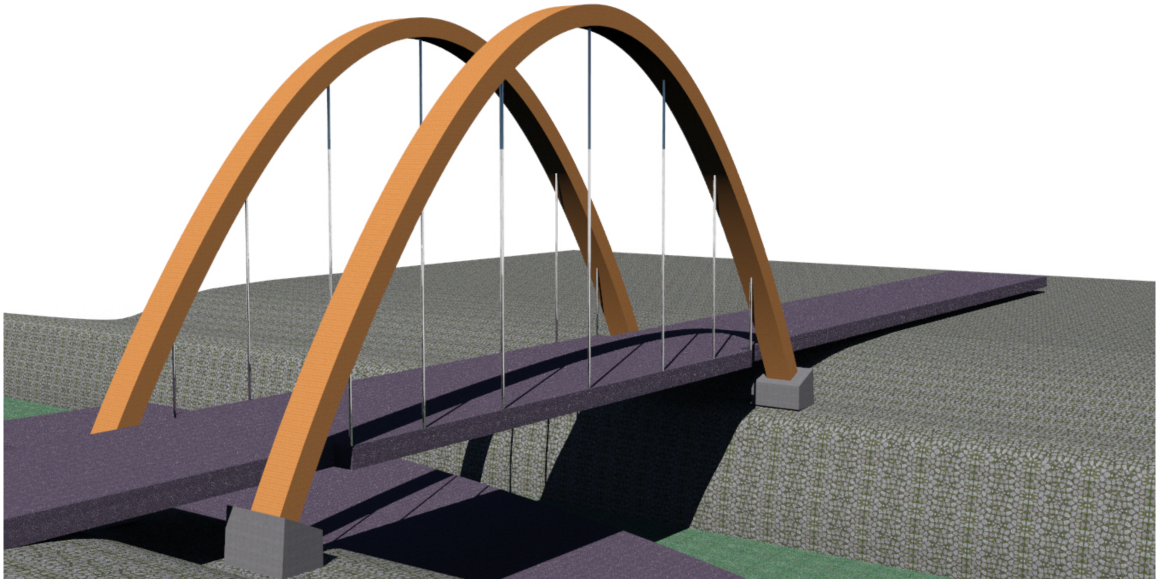


Figure 3.14: Arch bridge carrying the deck with compression arches and tension bars.

### 3.5.1 Assumptions

The bridge is assumed to span 20 metres and have a width of five metres. With traffic in just one direction at the time.

The number of tension bars that will carry the bridge deck is a parametric input to the calculations. The number of tension bars is set to seven in the beginning. This means that the *TNA2D Solver* will have seven nodes with loads and the model will consist of eight line segments. As presented in chapter 3.1, one of the assumptions for the *TNA2D Solver* is that all loads are vertical.

The arches are assumed to be constructed of GL32h, a high strength glue laminated timber with material properties taken from NS-EN 14080 (Table 5), Timber Structures (2016), as presented in Table 3.3.

Table 3.3: Mateiral properties for GL32h used in the case study.

| Properties | Value |
|---|---|
| Compression strength [N/mm$^2$] | 32 |
| E-modulus [N/mm$^2$] | 14 200 |
| Shear-modulus [N/mm$^2$] | 650 |
| Density [kg/m$^3$] | 490 |

All loads are assumed to be vertical, meaning all horizontal loads are neglected. The wind load is also neglected for simplicity. The self weight of the bridge deck is assumed to be 10 kN/m$^2$. Total variable loads, including both snow and traffic, are assumed to be 20 kN/m$^2$. Both these loads are distributed equally to each node of the arch. This gives 300 kN at each node, while the self weight of the arch will be less than 1 % of this at each node. The initial cross section is estimated to 800 mm x 500 mm.

In this case study both material and load factors are used. Material factors are taken from NS-EN 1995-1-1, Eurocode 5 (2010). Load factors are set to 1.2 for self weight loads and 1.5 for variables loads.

### 3.5.2 Designing the arch

**Creating the geometry**

Eight straight lines of equal length between the supports are first created. Together with the nodal loads and the cross section properties, these are used as input for the *TNA2D Solver*. As the case study in section 3.4 mentioned, lower arches are prone to larger bending moments. This case is thus initiated by an investigation of the structural performance of the three arches in Figure 3.15 which is created by adjusting the scale factor *r*.

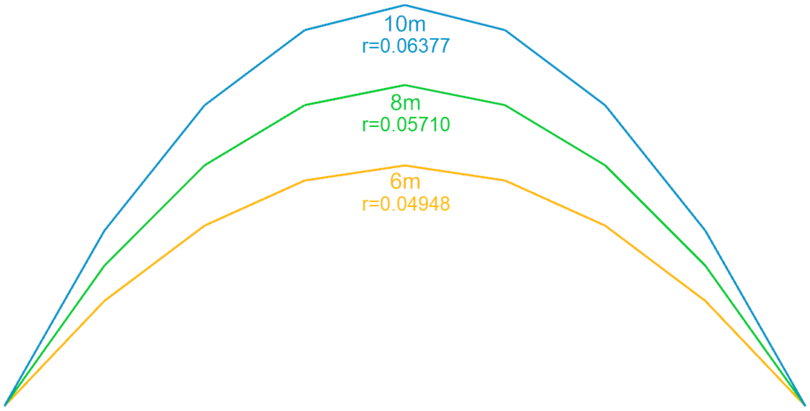


Figure 3.15: Different arch heights are obtained by adjusting the scale factor *r*.

Table 3.4 shows the reaction forces for each arch compared to results from an FEA. The difference between the FEA and the TNA are acceptable for the vertical reaction forces in all of the three cases, with a maximum deviation of 0.53 %. As expected, the horizontal force varies largely between the TNA and the FEA. All three cases of TNA have a fairly similar solution since the external loads remain unchanged and dominate the loading situation.

Figure 3.16 illustrates how the horizontal force increases as the branch force angle to a horizontal line decreases. Hence, as the branch force approach a vertical position, the accuracy of the TNA will increase. This corresponds well with the results from the FEA, where the horizontal forces decrease as the height increases, making the discrepancy between the normal forces from the two methods smaller.

Before selecting an arch height, a comparison between the different height to span ratios and discrepancy of forces are shown in Table 3.5. For the horizontal forces, the difference varies between 95-91 %, which is not accurate at all. With regards to the normal forces in the branches, a height to span ratio of 0.5 gives an acceptable solution when looking for a plausible result. With a difference of 13.1 % before considering the presence of moments, this solution is selected when going forward with the design process.

Table 3.4: The reaction forces for the three different arches in 3.15 with a cross section of 800 mm x 500 mm.

| | TNA 2D | | | FEA | | |
|---|---|---|---|---|---|---|
| | $R_V$ [kN] | $R_H$ [kN] | $N_{Max}$ [kN] | $R_V$ [kN] | $R_H$ [kN] | $N_{Max}$ [kN] |
| h = 6 m | 1073.6 | 50.1 | 1074.7 | 1077.7 | 1019.9 | 1480.1 |
| h = 8 m | 1075.8 | 50.52 | 1076.6 | 1080.7 | 766 | 1320.7 |
| h = 10 m | 1078.2 | 50.52 | 1078.9 | 1084.0 | 614.1 | 1240.8 |

Table 3.5: Difference ($\epsilon$) between the TNA and the FEA for different height to span ratios.

|  | h/s | $\epsilon_H$ [%] | $\epsilon_V$ [%] | $\epsilon_N$ [%] |
|---|---|---|---|---|
| h = 6 m | 0.3 | 95.1 | 0.38 | 27.4 |
| h = 8 m | 0.4 | 93.4 | 0.45 | 18.5 |
| h = 10 m | 0.5 | 91.2 | 0.54 | 13.1 |

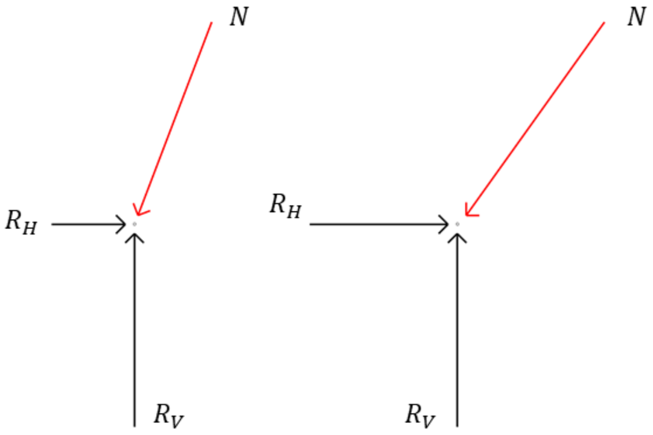


Figure 3.16: When decreasing the angle of the arch, the horizontal force increases. A low arch will, therefore, give high horizontal forces.

### Evaluation with TNA

Using the TNA tools to analyse the arch with a height of 10 metres, axial forces, stresses, and local buckling are investigated. The results are then verified and evaluated by an FEA of the arch.

Starting with the TNA, the *Capacity check* component from Figure 3.6 is used to control the utilisation and element buckling of the branches. As Figure 3.17 shows, the utilisation ratio in the arch is low, with a maximum of 0.11 in the branches near the supports. None of the elements shows a tendency for local buckling, as illustrated by the green arrows. Table 3.6 shows all results from the TNA.

Table 3.6: All relevant results from the *TNA 2D Solver*.

| Results from the TNA 2D | |
|---|---|
| $N_{max}$ [kN] | 1078.9 |
| $\sigma_{n.max}$ [N/mm$^2$] | 2.705 |
| $f_{yd}$ [N/mm$^2$] | 25.04 |
| UR | 0.11 |

**Evaluation with finite element analysis**

As for the FEA in section 3.4, the self weight of the arch is lumped onto the free nodes based on the length of the associated branches. The loads are calculated with the material density given in Table 3.3. Since the weight of the material is almost negligible compared to the stone arch, a subdivision of the elements is not performed here.

The utilisation ratio of the material after the FEA is shown together with the one from the TNA in Figure 3.17. Note that this only accounts for the normal stresses compared to the material capacity given as input, and not for the reduction factors from slenderness, which is looked at later. Hence making it a fitting comparison to the TNA. The utilisation ratio near supports are almost identical for the TNA and the FEA, indicating that a plausible shape has been found. The largest deviance occurs at the top where the inclination of the branches are lowest.

Force diagrams from this analysis are shown in Figure 3.18. The values show an arch in almost pure compression with a maximum bending moment of only 2.92 kNm at the middle. Table 3.7 shows the maximum values of shear and axial stresses, as well as the utilisation ratios. The results indicate a structure very close to the goal of a funicular arch. The difference in the utilisation ratio of 0.015 % is considered to be acceptable. Even though moments are present, they are of such a small magnitude that the residual stresses are compressive throughout the entire arch. Despite the reduction of material capacity from Eurocode, only 56 % of the material is utilised.

Table 3.7: Results from FEA in *Karamba* compared to the TNA. All stresses are given in N/mm$^2$. The first utilisation ratio is comparable to the one presented in 3.6 where only material stresses are considered. The latter one takes into account reduced capacity to account for e.g. buckling and LTB. Note that no extra material or load factors are used for this.

| | $\sigma_{\text{N.max}}$ | $\sigma_{\text{My.max}}$ | $\tau_{\text{Z.max}}$ | UR | $\text{UR}_{EC}$ | Buckling factor $\lambda$ | $w$ [mm] |
|---|---|---|---|---|---|---|---|
| FEA | 3.1 | 0.068 | $7.6 * 10^{-5}$ | 0.125 | 0.56 | 4.64 | 3.05 |
| TNA 2D | 2.705 | - | - | 0.110 | - | - | - |

Deflection, $w$, with the same load factors from section 3.5.1 are found to be only 3.05 mm. From NS-EN 1995-1-1:2004/NA:2010 Table NA.7.2 the maximum allowed deflection of a beam is L/300, which for a span of 20 metres gives a boundary of 66.7 mm. This is of course a simplification, but it is still a reasonable approach. The chosen cross section seems like a decent choice with this in mind.

Finally, global stability and global buckling are evaluated to get an indication of the arch's stability. This returns a loaf factor of 4.64,i.e. the arch is not prone to buckling for the given loading situation.

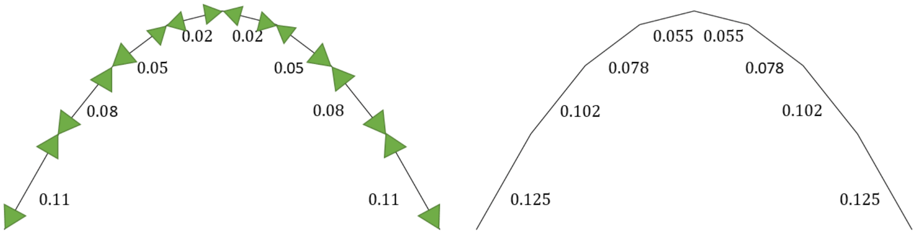


Figure 3.17: The arch is checked for local buckling and utilisation with 800mm x 500mm cross sections. The green arrows in the left arch indicate that the branches are not disposed to local Euler buckling. The numbers represents the utilisation ratio of the material. The arch to the right shows the utilisation from the FEA.

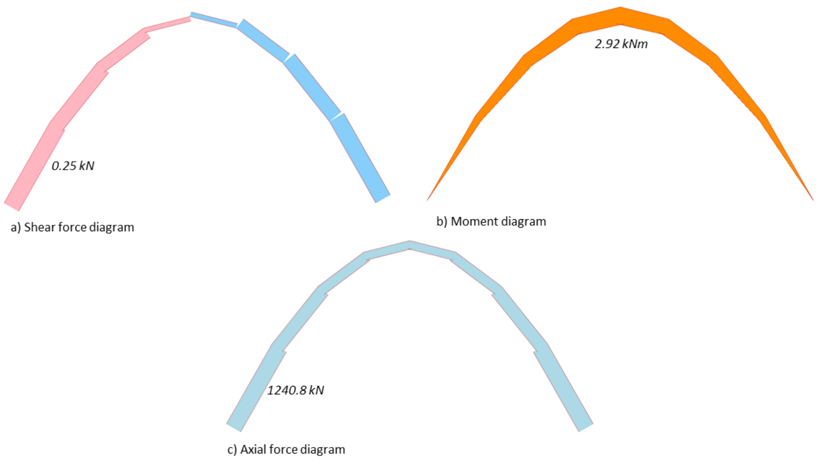


Figure 3.18: Force and moment diagrams from *Karamba*. The moment diagram is underneath the arch, meaning that there is tension at the underside of the arch.

### Comparing to initial theory

To conclude this section, a final control of the result is performed. In section 3.3.1, it was shown that an inverted hanging chain subjected only to nodal loads would obtain a parabola shape. The arch in this section has a load situation where the external nodal loads amount to 97.4 % of the total load, meaning the final shape should resemble that of a parabola for the theory to be valid. This is confirmed by creating an arch with the same span and height and comparing it to the arch's final shape, shown in Figure 3.19.

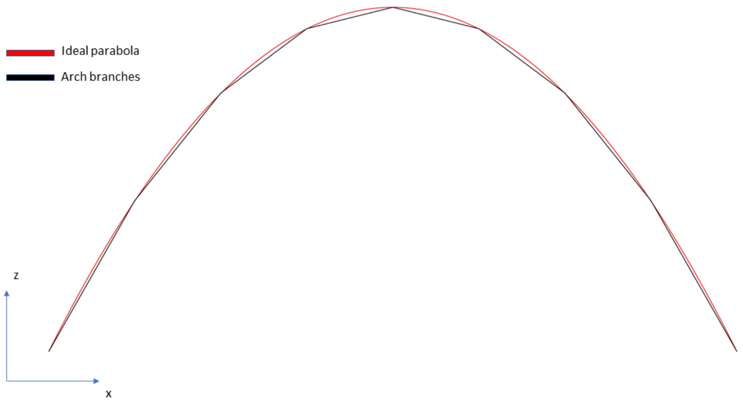


Figure 3.19: The black arch is a perfect parabola, while the blue arch is the obtained arch from *TNA2D Solver* with $r = 0.06440$.

### 3.5.3 Summary

Through this section a shape and cross section for a bridge spanning twenty metres has been suggested using the TNA, and then evaluated by an FEA. The results from this prove that for an arch with a height to span ratio larger than 0.5, the transfer of loads will primarily happen through axial forces. Since this is more efficient than through bending action, the utilisation due to axial stresses is significantly lower than a beam spanning the same length. However, the designer should be attentive to factors such as global buckling being more likely for slender constructions when choosing a cross section.

Furthermore, out-of-plane bending from wind loads will cause moment in the arch, and yet another reason to keep the initial utilisation from the TNA well below the material capacity.

Overall, the *TNA2D Solver* is able to create plausible arches with little effort in the initial design phase. The *Capacity Check* component produces material utilisation close to the ones from FEA. The local buckling of elements from this component was found to be ineffective in 2D, since the assumption of pinned connections between the branches in the TNA is invalid for a 2D arch.

## 3.6 Chapter summary

In this chapter, a tool for designing funicular two-dimensional arches has been designed, verified, and tested. Here the component is first trying to recreate an existing masonry structure, before a wooden arch bridge is conceptually designed. The resulting shapes were then analysed by finite elements to verify the plausibility of the design.

From the two test cases, the accuracy of the TNA-solver seemed to be dependent on the ratio between height and span of the created arches. In section 3.4, the results indicated the presence of bending moments with a height to span ratio of 0.32; the timber arch in section 3.5 was almost funicular with a height to span ratio equal to 0.5. Therefore, the latter seems like a minimum height to span ratio if a funicular arch is wanted.

For both case studies the TNA2D's tendency to reproduce too low horizontal forces was discussed, resulting in non-conservative normal forces.

Additionally, some flaws and needs for improvements for the development of a fully functional 3D tool were identified. In the Padre Pio Church, the need for individual loading of the free nodes presents itself, as nodal loads do not occur in every node of the arch. This is not done for the 2D case and limits the possibilities for playing with the shapes' response to different loading.

The value of a component controlling for local buckling of branches was found to be inefficient for arches which in reality acts as a monotonic element, and not hinged trusses. In Table 3.6, the critical buckling load was vastly larger than the normal forces. This component could prove more important when implementing it in three-dimensional gridshells, where the connections between the elements are more like hinges.

Despite some minor issues, the component shows potential for being a useful design-tool in early stages, for both architects and engineers, as long as considerations and precautions are taken.

# 4 Implementation of TNA in 3D

Moving on from 2D to 3D, this chapter describes the implementation of the TNA through a complete set of components in Grasshopper. All the components are developed using Visual Studio and written in the C# language. The function of each component is described as well as its required inputs and outputs. Together, these components will enable an intuitive workflow for early stage design of single layer gridshells in compression.

To continue the trend of giving animal names to Grasshopper plugins, this collection of components is given the name Camel. The camel with its humps is a reference to curved shapes and indeed only too appropriate an appellation for the form finding of gridshells. The Camel plugin, with all its components and how it appears in Grasshopper, is shown in Figure 4.1.

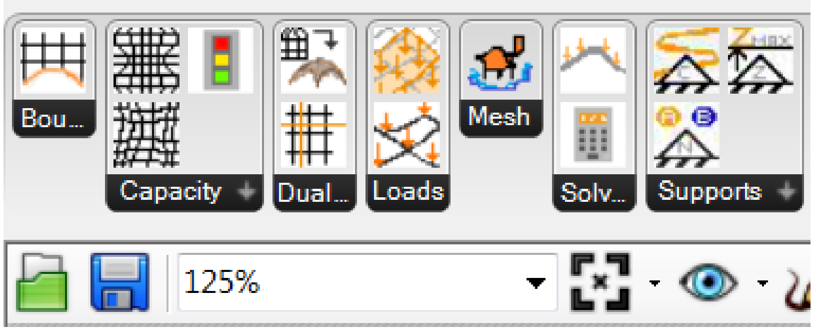


Figure 4.1: Illustration of the Camel components in the *Grasshopper* toolbar.

## 4.1 Mesh Generator

Starting from the beginning, an initial form diagram has to be created in order for the TNA to run. The *Mesh Generator*, Figure 4.2, uses a surface to establish a mesh with the lines and edge points required for the TNA.

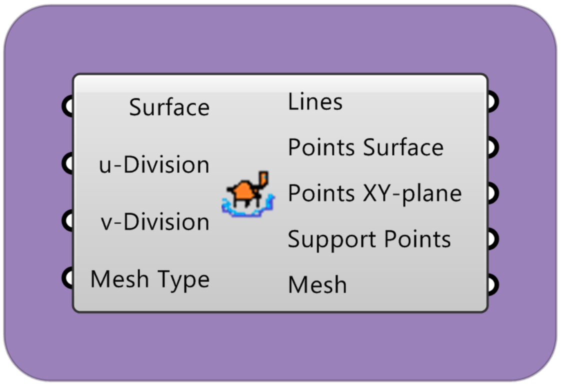


Figure 4.2: Camel component *Mesh Generator*.

**Input**

- Surface: Initial surface to be meshed.
- u-Division: Integer setting the mesh division in u-direction.
- v-Division: Integer setting the mesh division in v-direction.
- Mesh Type: An integer ranging from 1-8 to select the meshes in Figure 4.3.

**Output**

- Lines: Mesh lines used as input in *TNA3D Solver*.
- Points Surface: All the mesh vertices.
- Points XY-plane: Mesh vertices projected down to the XY-plane.
- Support Points: Points located along the surface edges.
- Mesh: Initial mesh used in *TNA3D Solver*.

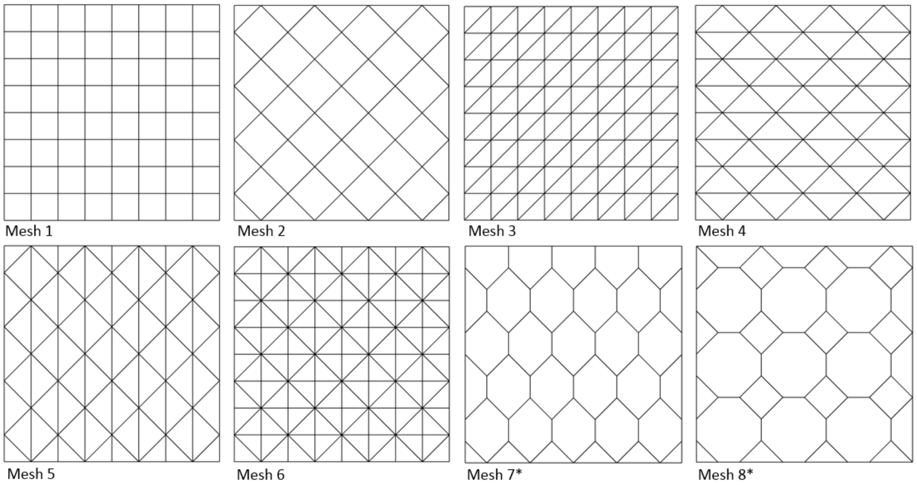


Figure 4.3: Mesh type 1-8. **Mesh 7 and 8 are divided into a grid with uCount = vCount = 10; while the rest is divided with uCount = vCount = 8.*

As Figure 4.3 shows, the component creates eight different geometries. After initial testing, a limitation in Rhino and Grasshopper appeared. In Rhino mesh-faces are only defined as triangles and rectangles, making it difficult to create polygon meshes with more than four vertices. Programming the lines for mesh 7 and 8 was possible, but the mesh could not be easily established, consequently making them unavailable for the TNA. Since timber gridshell usually consists of square, diamond, triangular, and hexagonal grid, (Pourebrahimi et al. (2015)), this issue was not further considered.

Arbitrary meshes created outside the *Mesh Generator* may also be used in the TNA, exemplified in the case study presented in section 5.4.2 and section 5.5.2.

## 4.2 Support selection

Apart from the support points generated from *Mesh Generator* the ability to specify support points are ensured in three different ways. The user can select all points below a specified height, all points that are enclosed by curves, or points close to specific Rhino points. These support points are used in the same manner as the output from *Mesh Generator* in the TNA analysis. Should duplicate support points enter the *TNA3D Solver*, they will be removed there.

### Support z-value

The *Support z-value* component, Figure 4.4, makes it possible to select all points in a list which has a z-coordinate lower than the specified *Zmax* value. The points fulfilling this are returned.

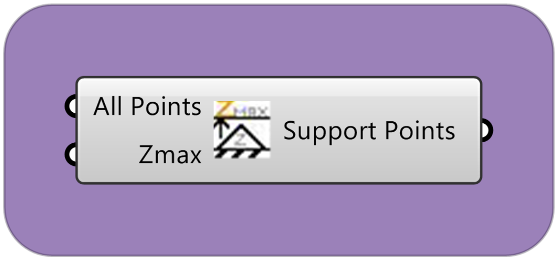


Figure 4.4: Camel component *Support z-value*.

**Input**

- All Points: All the surface points from *Mesh Generator* or a list of arbitrary points.
- Zmax: The upper limit for the z-coordinates values selected as supports. By default this is set to 0.0001.

**Output**

- Support Points: All points from the list *All Points* fulfilling the criteria in *Zmax*.

### Support Curves

Another alternative is to draw a curve or multiple curves in Rhino, enclosing the surface points intended as supports, using the *Support Curves* component, Figure 4.5. The component returns the selected points.

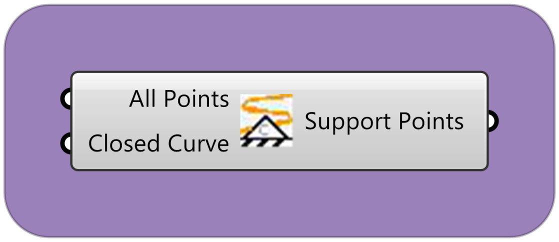


Figure 4.5: Camel component *Support Curves*.

**Input**

- All Points: All the surface points from *Mesh Generator* or a list of arbitrary points.
- Closed Curve: Single curve or list of curves.

**Output**

- Support Points: A list of points enclosed by the input curve(s).

### Support Nearest Point

Finally, the support points may be selected by using Rhino points close to the preview geometry from Grasshopper as input, with the *Support Nearest Point* component, Figure 4.6. The closest point to each drawn point will become a support point. This can be useful for placement of discrete supports such as columns.

The input Rhino points may also be a list of points generated with Grasshopper. This can be quite convenient if one wants a parametric list of points.

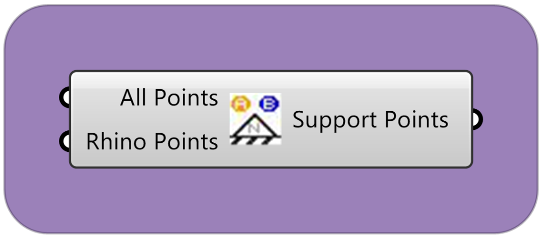


Figure 4.6: Camel component *Support Nearest Point*.

**Input**

- All Points: All the surface points from *Mesh Generator* or a list of arbitrary points.
- Rhino Points: Point or list of points drawn in Rhino or generated from Grasshopper.

**Output**

- Support Points: A list of the points closest to the Rhino points.

## 4.3 External loads

External vertical loads such as self weight from cladding and snow loads are added to the model in two different ways: nodal loads or area loads. Self weight of the branches are implemented directly in the *TNA3D Solver* component.

### Nodal Loads

The *Nodal Loads* component, Figure 4.7, applies a user-specified nodal load to all the mesh vertices enclosed by a curve. These curves may enclose the whole structure or only parts of it, depending on the desired load situation.

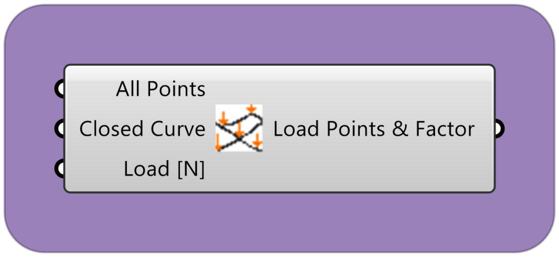


Figure 4.7: Camel component *Nodal loads*.

**Input**

- All Points: List of surface points projected to XY-plane from *Mesh Generator* or a list of arbitrary points.
- Closed Curve: Single curve or list of curves.
- Load [N]: Number setting the load to be applied at each node.

**Output**

- Load Points & Factor: A list containing the nodal coordinates and corresponding load.

### Area Loads

The *Area loads* component, Figure 4.8, applies a uniformly distributed load along the branches enclosed by the selected curves. The loads are then lumped to the nodes based on the coinciding branches' lengths.

Note that this component is only valid for certain meshes, because of the complex geometry for some of the meshes. The valid meshes are mesh 1, 2, and 3.

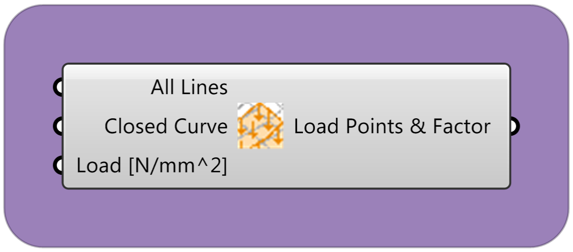


Figure 4.8: Camel component *Area Loads*.

---

**Input**

- All Points: List of surface points projected to XY-plane from *Mesh Generator* or a list of arbitrary points.
- Closed Curve: Single curve or list of curves.
- Load [N/mm$^2$]: Number setting the evenly distributed load.

**Output**

- Load Points and factor: A list containing the nodal coordinates and corresponding load.

---

## 4.4 TNA 3D Solver

This is the component which implements the TNA theory from section 2.1. The components' input and output are first presented, before an in-depth description of its functionalities follows.

**TNA 3D Solver**

From the inputs given by the components presented in section 4.1, 4.2, and 4.3, the *TNA3D Solver* component, Figure 4.9, finds a funicular solution for the structure, if one exists.

This component is used for the initial analysis where all edges are supported, and then the user has several modifications opportunities presented in section 4.5 and 4.6.

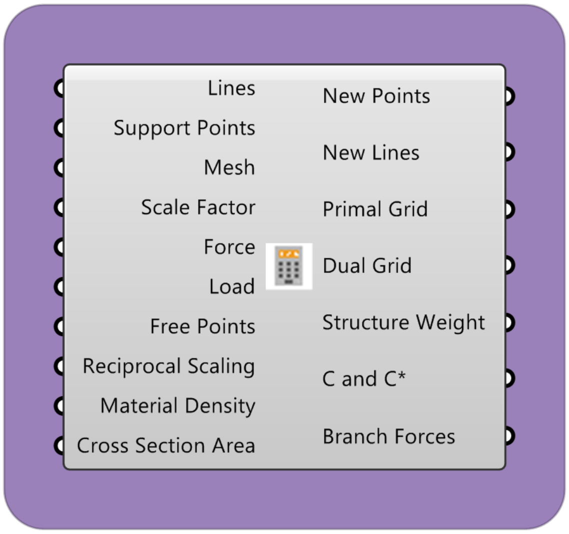


Figure 4.9: Camel component *TNA3D Solver*.

**Input**

- Lines: List of lines from the mesh used as branches in the analysis.
- Support Points: List of points which are to be restrained from translation during the analysis. In the first analysis, these should be all the edge points.
- Mesh: One, or multiple, meshes representing the base surface.
- Scale Factor: Scale factor r, double. Describes the relationship between the form and force diagram. Can be used to adjust the height.
- Load: List of coordinates and values calculated in sec. 4.3.
- Free Points: Dictionary containing the edges with free points from sec. 4.6.
- Reciprocal scaling: Double. Used as a constraint for the linear optimisation in eq. 2.37.
- Material Density: Double. Weight of material to be used for calculations of self weight [kg/m$^3$].
- Cross section area: Used in calculations of self weight [mm$^2$].

**Output**

- New Points: List containing the new surface points after the solution is found.
- New Lines: List of lines representing the geometry of branches.
- Form Lines: List of lines representing the primal grid.
- Reciprocal Lines: List of lines representing the dual grid.
- Structure Weight: Double. The total weight of all branches [N].
- **C** and **C*** matrix: Matrix used as input for the modification of the reciprocal diagram in sec. 4.5.
- Branch forces: List of doubles representing the forces in Newton for each branch. Sorted according to the *New Lines* output.

### Calculations of self weight

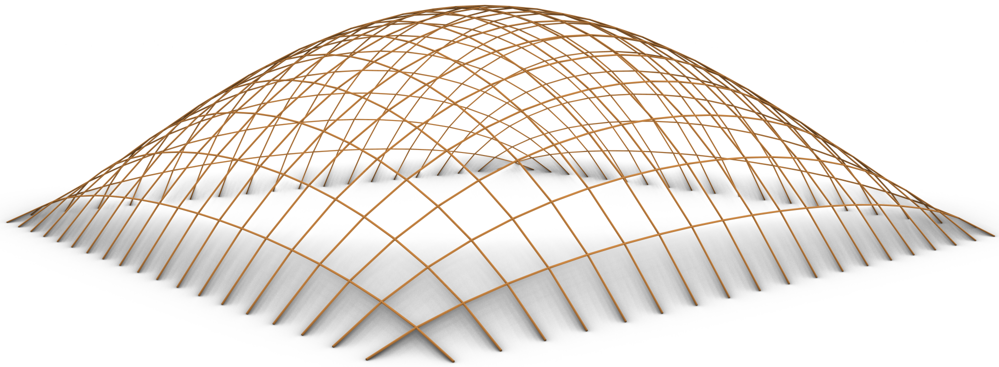


Figure 4.10: The reference model for evaluating the effect different numbers of iterations have on the approximated self weight. Square base of 50 m x 50 m with a target height of 15 m and a total of 760 branches.

Self weight from the bars are in the *TNA3D Solver* calculated from equation (3.2). The branch length matrix $\boldsymbol{L}$ are needed for the first optimisation problem, and calculated from (2.37). An iterative solver that recalculates the solution for each time the branch lengths change (such as the one in sec. 3.2) would quickly become a costly procedure in terms of computational time.

Since the first iteration calculates the branch length based on their length in the xy-plane, the second iteration would most likely have the greatest impact on the accuracy of the solution, since the actual branch lengths will be used for this solving procedure. To test the effect of updating the self weights before recalculating the solution, a simple test with a rectangular surface meshed into 760 branches of 50 mm x 50 mm timber elements is performed. The scale factor $r$ is adjusted for each case in order to get the same target height of 15 metres. The increased accuracy after each iteration are compared to the computational cost in order to find a number of iterations that yields a satisfying result, without slowing the component more than necessary.

Table 4.1 shows the effect of recalculating (2.26) after all the coefficient matrices are updated. The original calculations with one iteration use 83.9 % of the actual self weight in the calculations with a computational time of 5.7 seconds. Rerunning the solver with the updated branch lengths one time increases the accuracy to 92.5 % with a 3.0 seconds increase in computational time. For three iterations just 1.0 % increase in calculated self weight is achieved compared to two iterations; whereas the computation takes 2.65 times longer than the original.

Table 4.1: The effects of updating the self weight after finding new nodal heights $\boldsymbol{z}_N$ from equation (2.26).

| | i = 1 | i = 2 | i = 3 |
|---|---|---|---|
| Solver time [$s$] | 5.7 | 8.7 | 15.1 |
| Final height [m] | 15.08 | 15.06 | 15.01 |
| Scale factor $r$ | 0.132 | 0.124 | 0.123 |
| Total self weight [kN] | 16.82 | 18.60 | 18.80 |
| Actual self weight [kN] | 20.05 | 20.10 | 20.10 |
| Self weight in calculations [%] | 83.9 | 92.5 | 93.5 |
| Relative increase in time | 1.0 | 1.53 | 2.65 |

Furthermore, from (3.2), the bars with one end connected to a fixed node will "lose" half of the self weight to the supports; indicating that even for a hundred iterations the solver would not reach the exact value for self weight. From this example, two iterations for the solver are found acceptable in order to catch the biggest discrepancy between actual and horizontal branch lengths while retaining an acceptable computational time.

**Solving procedure**

The pseudo-code in Figure 4.12 displays how the component implements the theory in section 2.1. After the geometry has been sorted into matrices, the branch node matrix (**C**), coordinate differences (**u**, **v**, **w**), and branch lengths (**L** and $\mathbf{L}_\mathrm{H}$)of the primal grid are calculated from equations (2.13) - (2.10).

To solve (2.26), the **D** matrix has to be established by finding the reciprocal diagram's branch lengths. Due to the indeterminacy of most gridshells, the linear optimisation procedure in (2.37) is used to determine the coordinate differences **u*** and **v*** of the dual grid. An open source .NET wrapper of the Google.OR.Tools permits all the calculations to be performed within the component without the need for external computations.

When the coordinate differences are found, another round of optimisation (2.39) and (2.40) is needed to find the dual coordinates **x*** and **y*** due to the **C*** being non-invertible. If a solution exist, the dual grid is created and the **D** matrix is calculated; a new geometry is then found from (2.26).

### Half-edge data structure

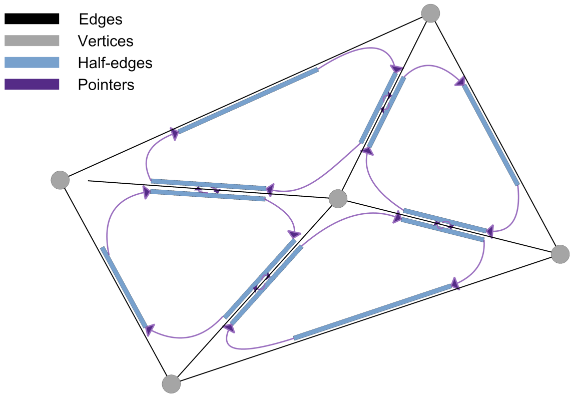


Figure 4.11: Illustration of half-edge data structure.

While the branch node matrix **C** may be established in a straightforward manner based on the input lines, dual matrix **C*** demands a different approach. Established according to section 2.3.2, the dual matrix is dependent on the area enclosed by the primal branches. In Rhino, these areas may be represented by a mesh face enclosed by the branch lines. By rotating counterclockwise around each face, the columns of **C*** are determined from (**??**). The most efficient way of working through each face of the input mesh is by transforming the mesh from Rhino to a mesh using a half-edge data structure.

The half-edge data structure represents the mesh by a list of shared vertices and list of faces storing pointers for each vertex. Instead for storing the faces' edges, half-edges are created by subdividing the edges. Each half-edge stores a pointer to its adjacent face, the next half-edge, and the opposite half edge as shown in Figure 4.11. By rotating counterclockwise around each faces' half-edges the **C*** matrix is established in an efficient way. A free and open library for C# named *Plankton* from *Plankton* (2017) is used in the component.

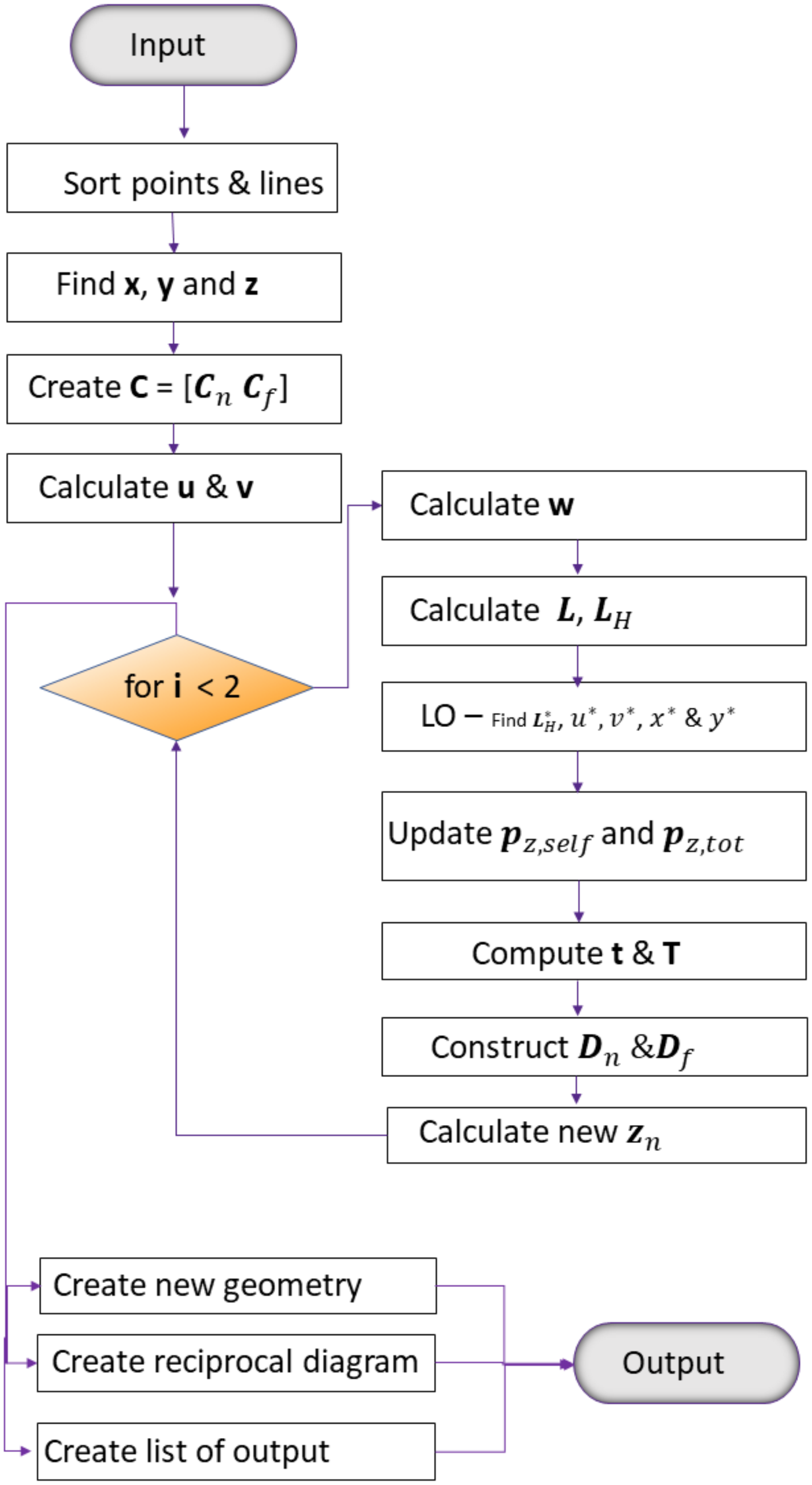

Input
Sort points & lines
Find **x**, **y** and **z**
Create **C** = [$\boldsymbol{C}_n$ $\boldsymbol{C}_f$]
Calculate **u** & **v**
for **i** < 2
Calculate **w**
Calculate $\boldsymbol{L}$, $\boldsymbol{L}_H$
LO – Find $\boldsymbol{L}_H^*$, $u^*$, $v^*$, $x^*$ & $y^*$
Update $\boldsymbol{p}_{z,self}$ and $\boldsymbol{p}_{z,tot}$
Compute **t** & **T**
Construct $\boldsymbol{D}_n$ &$\boldsymbol{D}_f$
Calculate new $\boldsymbol{z}_n$
Create new geometry
Create reciprocal diagram
Create list of output
Output

Figure 4.12: Pseudo-code for the implementation of *TNA3D Solver* including the linear optimisation (LO).

## 4.5 Manipulation of reciprocal diagram

The reciprocal diagram generated from the TNA analysis displays the horizontal force distribution in each branch. From section 2.2, a grid where four or more branches are connected to the same node will be statically indeterminate. By utilising this, the shape of the geometry, **G**, could be altered by manipulating the reciprocal diagram before recalculating the TNA.

### Internal Arch

The first created component manipulates the dual grid by stretching the lines intersected by an input line, resulting in a change of the dual grid similar to the one in Figure 4.15 b). The component is named *Internal Arch*, Figure 4.13.

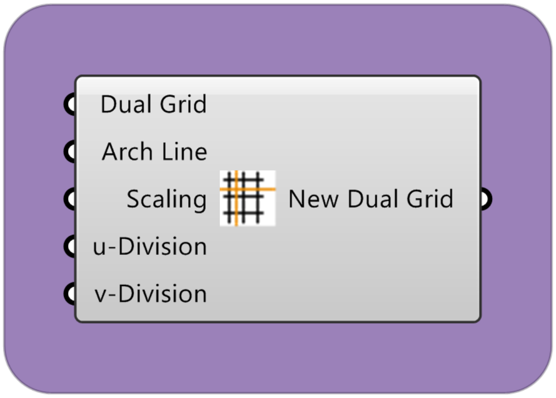


Figure 4.13: Camel component *Internal Arch*.

**Input**

- Force diagram: List of lines from *TNA3D* output.
- Arch line: Line or list of lines intersectin the dual grid where the modifications should happen.
- Scaling: Double. Adjusting the stretching of the reciprocal.
- u-Division: Integer. The same number as used in *Mesh Generator*.
- v-Division: Integer. The same number as used in *Mesh Generator*.

**Output**

- New Dual Grid: List of lines representing the new dual grid.

### Force To Form

A component calculating the geometry after modifying the dual grid is also created, Figure 4.14. Since the number of lines and thus the **C** and **C*** matrix remain unaltered throughout the entire operation, the component takes these as input from the *TNA3D Solver*. This, in turn, saves a new LO-problem, resulting in a faster solution time.

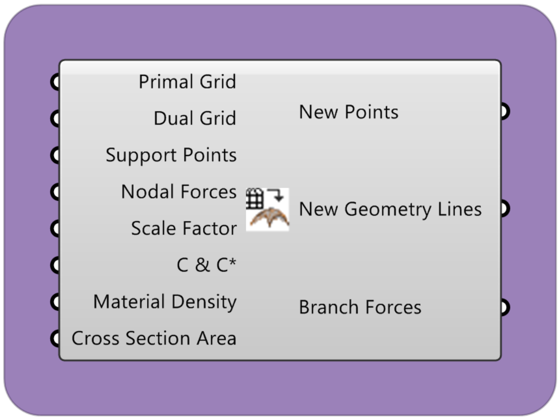


Figure 4.14: Camel component *Force To Form*.

**Input**

- Primal Grid: List of lines from *TNA3D Solver*.
- Dual Grid: List of lines from *InternalArch*.
- Support points: List of points restrained from translation.
- Nodal forces: List of coordinates and values from load components in section 4.3.
- Scale factor: Scale factor $r$, double. Used in the TNA analysis, eq.2.26.
- C and C*: Matrices **C** and **C*** from the *TNA3D solver*.
- Material density: Double, weight of materials kg/m$^3$.
- Cross section area: Double, area in mm$^2$.

**Output**

- New Points: List of new geometry points.
- New Geometry Lines: List of the new lines.

If modifications of the dual grid are done without using the *Internal Arch* component, it is important that the sorting of the dual grid remains unaltered when set as input to the *Force To Form* component. If this is not the case, the branch matrices will no longer be correct, resulting in an invalid solution.

### Effect of manipulating the force diagram

Fig.4.15 shows the resulting changes in geometry when using the two components presented in this section. The primal grid covers an area of 10 m x 5 m and consists of a total of 232 branches.

By drawing a single line through the initial line as in fig 4.15 b) the length of the intersected branches increase.Altering $\boldsymbol{L}_H^*$, equation 2.22 shows that a larger $\boldsymbol{L}_{ji}^*$ will increase the value of related indices in the $\boldsymbol{D}_N$. The only change in equation 2.26 will be said $\boldsymbol{D}_N$ matrix, meaning that a larger branch length yields lower z-values for the free nodes. This creates an internal "arch" in the places where the dual grid is modified. Figure 4.15 c) and d) illustrate the resulting shape when adding several lines to the created components introduced in this section; perpendicular and also parallel to each other.

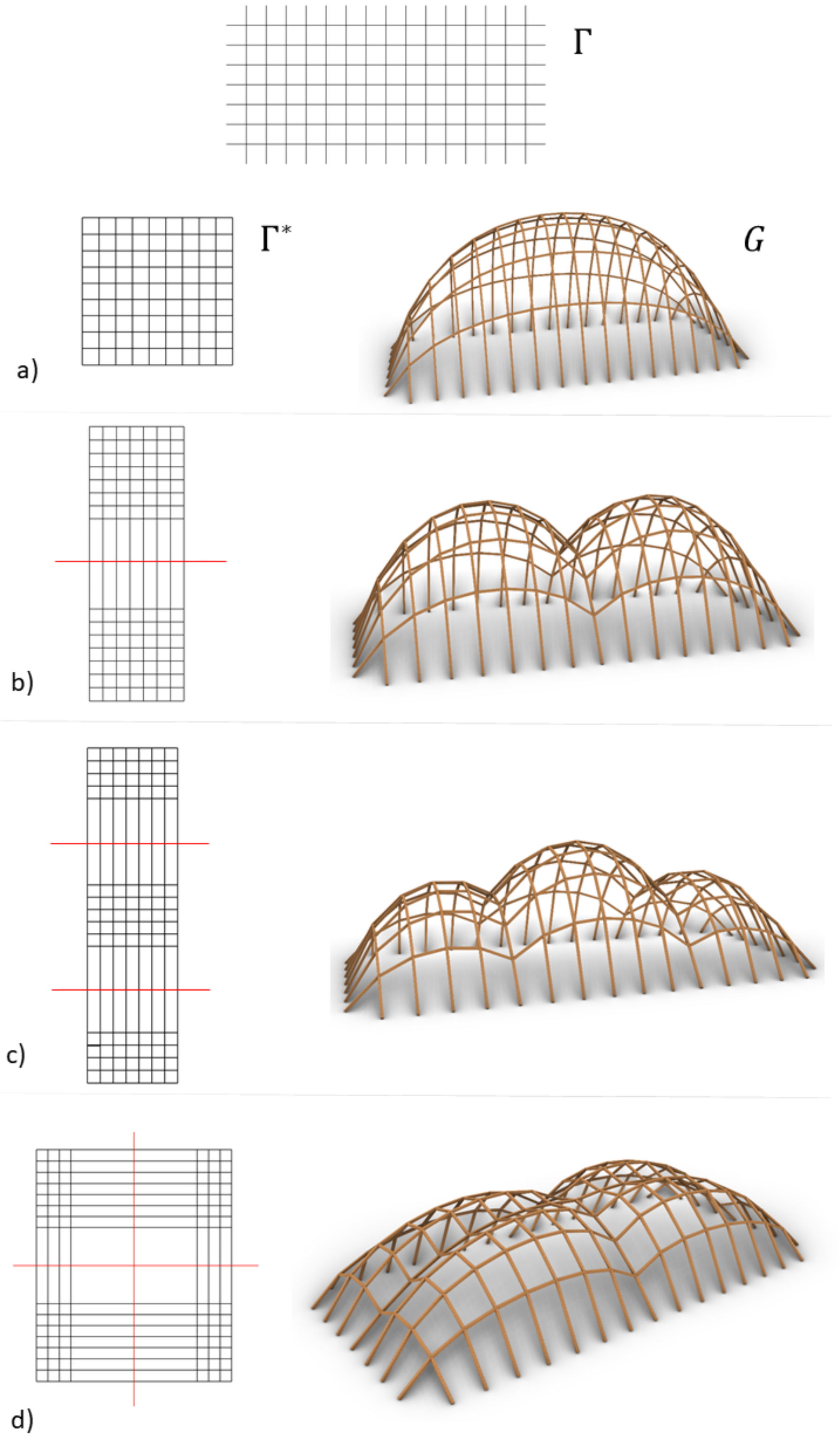


Figure 4.15: Figure a) shows the primal grid **Γ**, dual grid **Γ***, and geometry **G** after an initial TNA. By drawing intersecting lines (displayed in red) through the force diagram, the magnitude of forces in the reciprocal branches will increase by the selected scale factor. The dual grids are rotated 90 degrees counterclockwise compared to the primal grid.

## 4.6 Free edges and openings

For the situations where the boundary conditions are short lines or a number of discrete points resembling for example support walls or columns, the equilibrium of the resulting funicular edge arches must be considered. As described by Block (2009) there are two alternatives for doing this:

- First choose the internal force distribution of the mesh, with the form of the arch following from this.
- Secondly, define the geometry of the edge arch in the horizontal plane and thereby dictating the internal force distribution

For the first case the shape of the edge arch in the primal grid is funicular for the loading caused by the connected branches, as illustrated in Figure 2.10.

The latter alternative where the horizontal projection of the arch shape, which is decided by the user, needs to be funicular in the plane in order for the structure to be in global equilibrium. This funicular shape dictates the force branches ending in it. Hence the lengths of the corresponding dual branches will be fixed. This funicular equilibrium is represented by attaching a funicular polygon to the side of the reciprocal diagram, as shown in Figure 2.11. A component creating the internal arches from the first method is presented here.

The initial *TNA3D Solver* component will solve the system for a condition with all boundary edges fixed. The horizontal branch forces from the dual grid are then used as nodal loads in a two-dimensional TNA similar to the one explained in section 3.2. The designer draws a curve in Rhino around the points which are to be released.

Figure 4.17 illustrates how the component can be used for one or multiple edges. A problem encountered when introducing the free edge was the lack of feasible solutions. This is caused by the linear optimisation implemented in the *TNA3D Solver*. For some u- and v-divisions of the grid, no feasible solutions can be found. The user may also experiment with the reciprocal scaling $d$ used as a constraint in equation 2.37. This problem becomes more relevant for each free edge added to the solution.

### Free Edge

The *Free Edge* component, Figure 4.16, makes it possible to create openings in the structure. By drawing one or multiple curves in Rhino around the points to be released, the component returns an updated primal grid with funicular edges to be used in the *TNA3D Solver*.

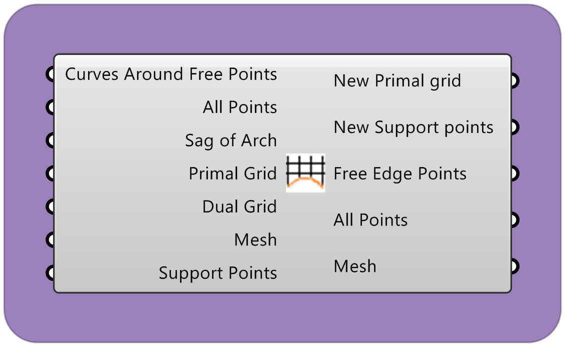


Figure 4.16: Camel component *Free Edge*.

**Input**

- Curves Around Free Points: Curve or list of curves enclosing the points to be modified.
- All Points: List of all surface points.
- Sag of Arch: Double, adjusting the height of the arch similarly to the scale factor, $r$, in section 3.2.
- Primal Grid: List of lines from the *TNA3D Solver*.
- Dual Grid: List of lines from the *TNA3D Solver*.
- Mesh: One mesh. Either user specified or from the *Mesh Generator*.
- Support Points: List of all support points.

**Output**

- New Primal grid: List of lines in the updated primal grid.
- New Support points: List of updated support points.
- Free Points: Dictionary with each free edge as a key and the corresponding points as a list.
- All Points: List of all points on the new geometry.
- Mesh: Updated mesh.

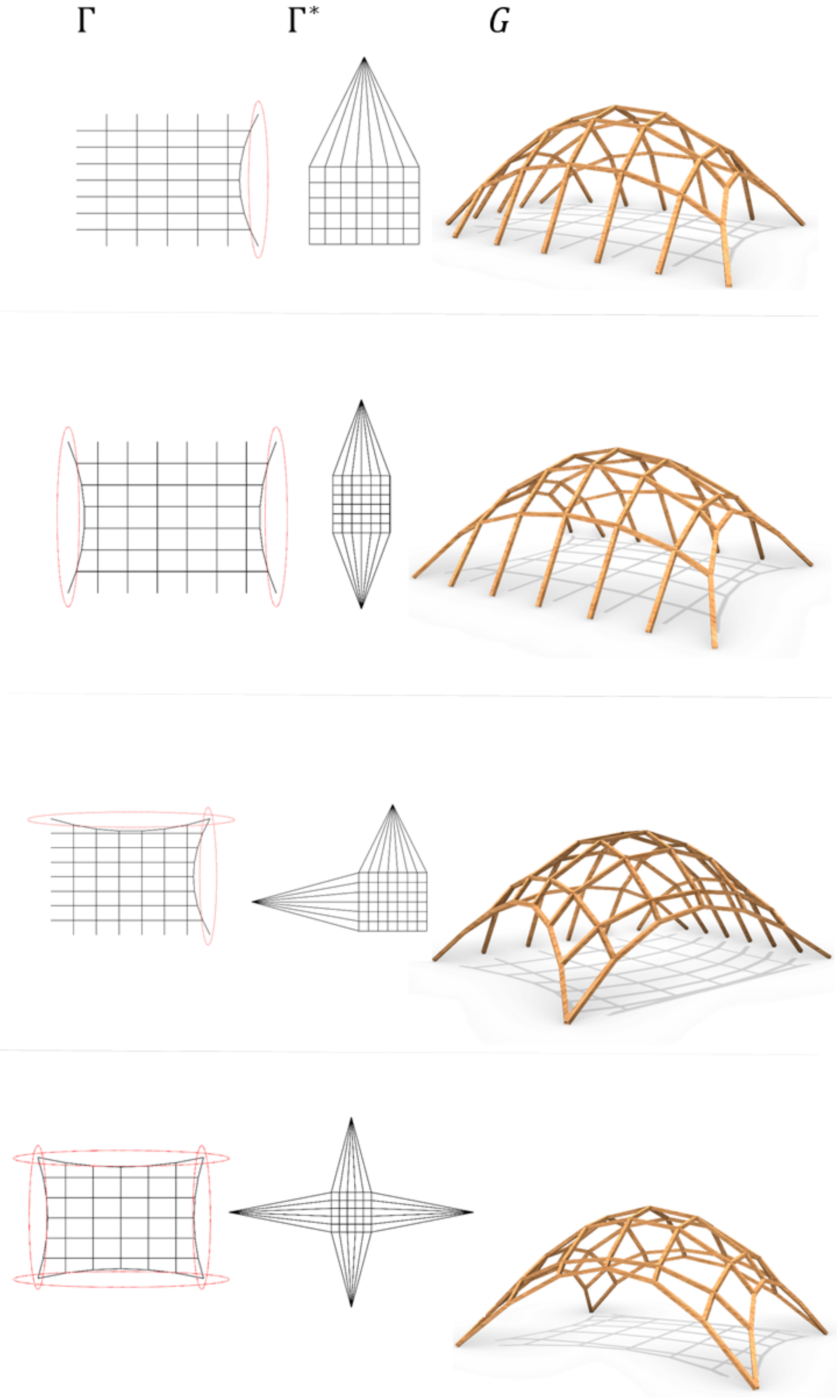


Figure 4.17: Illustration of how free edges in the primal grid change both the dual grid **Γ**∗ and the final geometry **G**. The dual grids are rotated 90 degrees counterclockwise to the primal grid.

## 4.7 Structural Performance

The components presented in section 4.1 - 4.6 are developed to establish a shape. When the shape meets the designer's goal, the next step is displaying the structural performance of the gridshell in an intuitive way. This is first done by extending the component from section 3.2 to 3D. Showing the utilisation and stability of the structure.

**Capacity Control**

Starting with local stability and utilisation, the *Capacity Control* component, Figure 4.18, controls the utilisation and the critical Euler buckling load of each member locally.

For the implementation of local buckling in the analysis component, it is assumed that all connections are pinned. This gives $K = 1$ in equation (2.41). or simplicity's sake it is also assumed that all bars in the construction are slender. This means that in some cases equation (2.41) will give a wrong Euler load. Therefore, buckling may occur even though the analysis does not indicate it. However, since this is not definite verification, but more of a guidance for finding a better form, this is not seen as critical simplification.

The visualisation of buckling is indicated with coloured arrows at each of of the branches in Rhino. If a bar is exposed to a force higher than the Euler load, the arrows become red. If the force is between 80-100 % of the Euler load, the arrows turn yellow. This is done with consideration to cases with bars with low slenderness ratios. If the force is smaller than 80 percent of the Euler load, then the colour will be green.

The utilisation of the branches is calculated by simply dividing the axial stresses on the material capacity from the input. For a utilisation lower than 80 %, the branches will appear green in the geometry preview; if the utilisation is between 80-100 % a yellow colour will instead appear; additionally, all bars with normal stresses higher than the material capacity will be displayed as red.

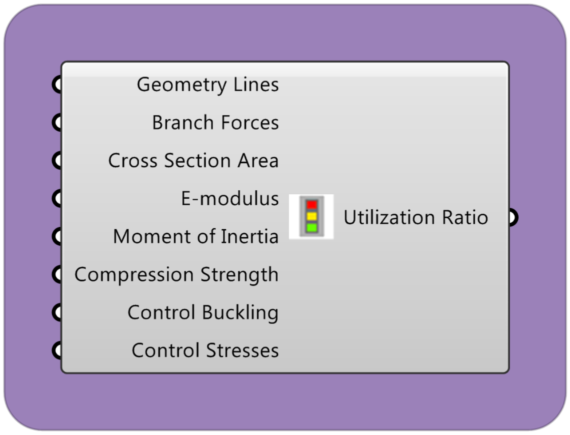


Figure 4.18: Camel component *Capacity Control*.

**Input**

- Geometry Lines: List of the branches from *TNA 3D Solver*.
- Branch Forces: List of branch forces from *TNA3D Solver*.
- Cross Section Area: Double, area of branches in $[mm^2]$.
- E-modulus: Double, modulus of elasticity in $[\mathrm{N/mm^2}]$.
- Moment of Inertia: Double, $[\mathrm{mm^4}]$.
- Compression Strength: Double, material capacity in compression, $[\mathrm{N/mm^2}]$. No load factors are used within the component.
- Control Buckling: Boolean, true if the component should control the member buckling.
- Control Stresses: Boolean, true if the component should control the member utilisation.

**Output**

- Utilisation Ratio: List of doubles displaying the utilisation of each branch. Colour map of buckling and/or utilisation.

## 4.8 Global stability

An attempt to analyse the global stability by using projected geometry as described in section 4.8, is implemented as the last components in this thesis. The idea is that by using two-dimensional geometry, the computational time will be reduced due to fewer degrees of freedom compared to a full global analysis. Hence the possibility for real time feedback is maintained.

In the analysis for global stability, two components were developed. Both components takes in the 3D geometry of the structure and the calculated forces, and simplify the buckling problem to 2D as shown in Figure 4.19. This is done by moving all the points in the geometry to z = 0 which results in a planar geometry in the xy-plane. Only in-plane buckling modes are considered. The FE-model is described in 2.5.2. The *Buckling 2D Bar* component, Figure 4.20, is based on the FE-bar theory; the second component, *Buckling 2D Beam* Figure 4.21, is based on the FE-beam theory.

The results from both components are the buckling-shapes and all the eigenvalues. The lowest eigenvalue will represent the critical buckling load factor. If the lowest eigenvalue should happen to be lower than 1, then the structure will subsequently be unstable for the applied loading.

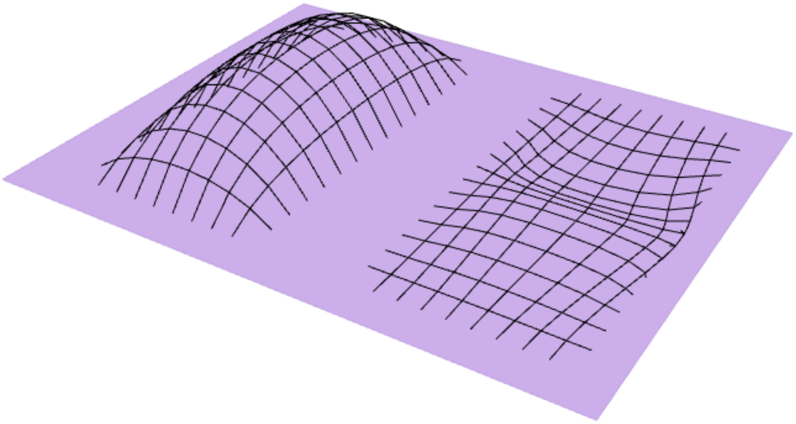


Figure 4.19: Going from 3D shape to 2D projection and illustrating the buckling shape using the *Buckling 2D Beam* component.

### Buckling 2D Bar

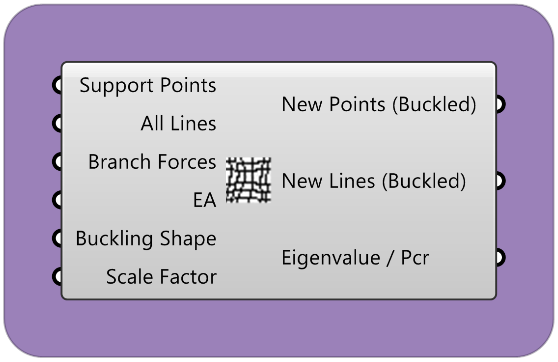


Figure 4.20: Camel component *Buckling 2D Bar*.

**Input**

- Support Points: List of support points.
- All Lines: All the structure lines from *TNA 3D Solver*.
- Branch forces: List of branch forces from *TNA 3D Solver*.
- EA: Double, E-modulus [N/mm$^2$] x Cross sectional Area[mm$^2$].
- Buckling Shape: Integer, the buckling shape to be displayed.
- Scale Factor: Double, scales the displacement of the buckling shape.

**Output**

- New Points: List of points displaying the buckled shape.
- New Lines: List of lines displaying the buckled shape.
- Eigenvalues / $P_{cr}$: List of all the values.

### Buckling 2D Beam

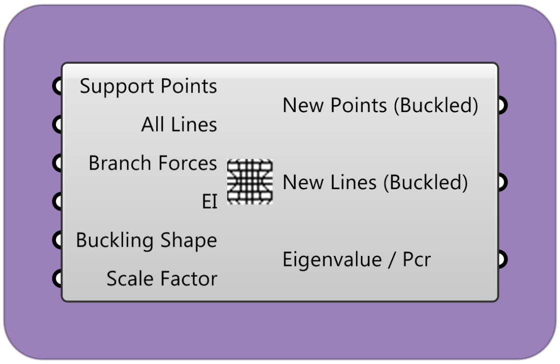


Figure 4.21: Camel component *Buckling 2D Beam*.

**Input**

- Support Points: List of support points.
- All lines: All the structure lines from *TNA 3D Solver*.
- Branch forces: List of branch forces, from *TNA 3D Solver*.
- EI: Double, E-modulus [N/mm$^2$] x Moment of inertia [mm$^4$].
- Buckling Shape: Integer, the buckling shape to be displayed.
- Eigenvalues / P$_{cr}$: List of all the eigenvalues.

**Output**

- New Points: List of points displaying the buckled shape.
- New Lines: List of lines displaying the buckled shape.
- Eigenvalues / P$_{cr}$: List of all the values.

# 5 Case Studies

In this chapter the components introduced in chapter 4 are tested and used in several case studies. In the first two cases, the purpose is to evaluate the results from the *TNA 3D Solver* by comparing them to an FEA. Before a final case study is shown to demonstrate the flexibility of the TNA, there will first be provided two cases where the components are used in a design process.

## 5.1 Case study I: Quadratic dome

This case study investigates the discrepancy between the *TNA3D Solver* and an FEA. A quadratic base area is created and from this a dome is designed with the TNA tools. First, the performance of different meshes from section 4.1 are tested on a 3.5 metre high dome. Afterwards, the effect of different height to span ratios are investigated in order to understand how the TNA better.

### 5.1.1 Performance of meshes

The meshes in Figure 5.1 are transferring the load in different directions depending on their geometry. Mesh 6 is a four-way load transfer system, meaning that the load is transferred in four different directions, Pourebrahimi et al. (2015). Mesh 3, 4, and 5 are three-way transfers, whereas mesh 1 and 2 are two-way transfers. The only difference between mesh 4 and 5 is the direction of the bracing. For a symmetric geometry and load, mesh 4 and 5 will give the same results. Because of this, the results of these meshes are merged together in tables and plots.

A plane surface of 7 m x 7 m, which is divided into a symmetric grid of 225 nodes, using the *Mesh Generator* with u- and v-division equal to 15, is used as a basis for this study. The different meshes are analysed with the *TNA3D Solver* to get a symmetrical funicular vault scaled to a height of 3.5 metres. The structures and associated grid meshes are all shown in Figure 5.1. The material used is wood with a self weight of 380 kg/m$^3$ and an elastic modulus of 14200 N/mm$^2$. The cross section is 35 mm x 35 mm for all branches. The external loading is set to 2 kN/m$^2$ and is, as a simplification, evenly distributed to all nodes independent of the nodes distance to each other. The self weight is small and represents approximately 1 % of the total load for all grid types in question. In practice, the total load is unaffected by the self weight, making the results of different grids more comparable.

*Karamba* is used to validate the results. As a simplification, the self weight is also here evenly lumped to all nodes, to make the load transfer more equal to the *TNA3D Solver*. The analysis in *Karamba* is performed for pinned and rigid joints.

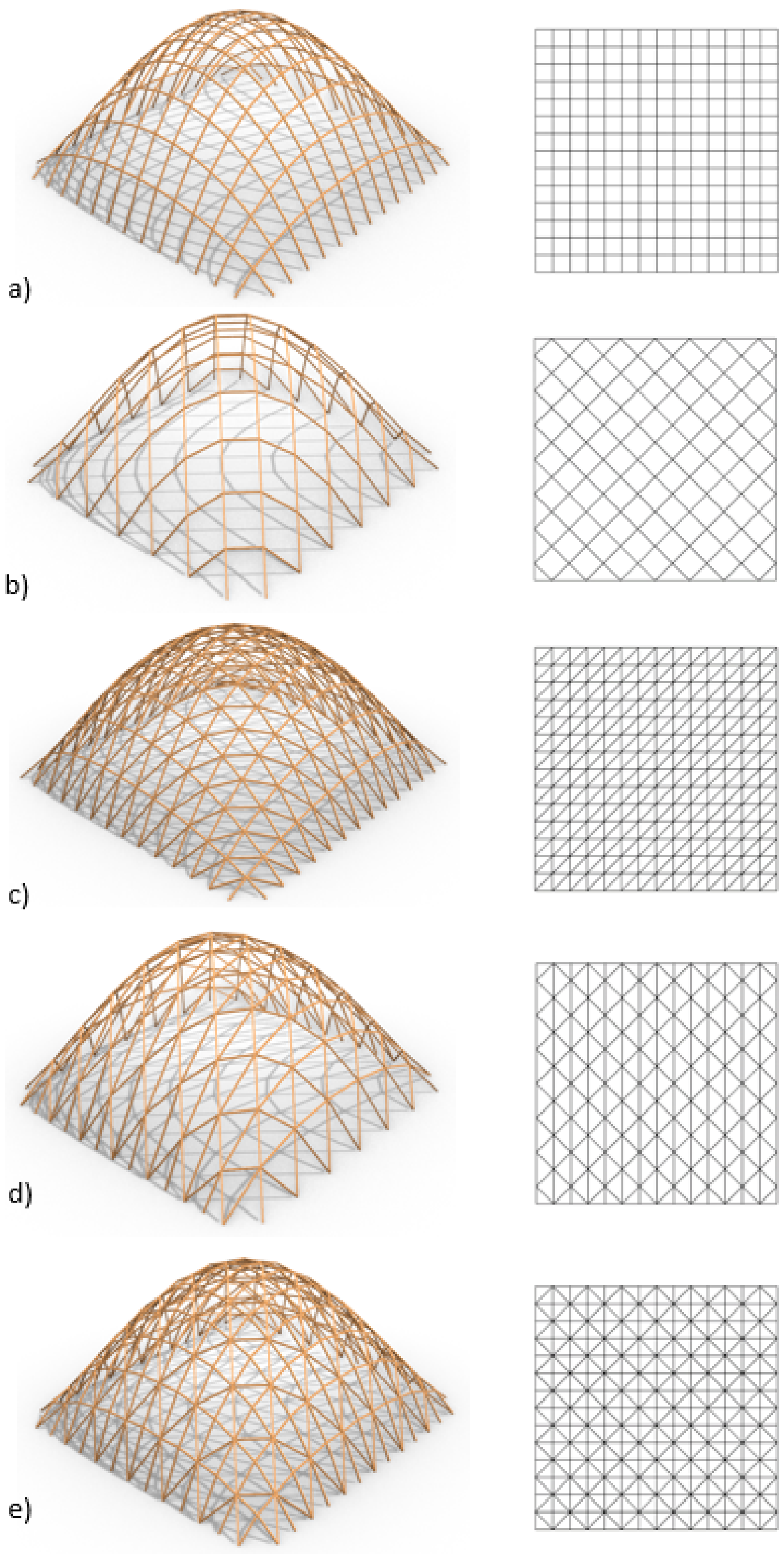


Figure 5.1: All structures and associated meshes used in the comparison study. a) Mesh 1. b) Mesh 2. c) Mesh 3. d) Mesh 4/5. e) Mesh 6.

### Results

The analyses with *Karamba* confirm, as expected, that for all grid types, the elements acts solely in compression, which is the goal of the *TNA3D Solver*. The maximum and minimum axial forces from the analysis are plotted in Figure 5.2; the biggest difference of maximum axial forces between the TNA and FEA is 32 %, and applies to mesh 6 with rigid joints; with a difference of 15 % between pinned/rigid joints, this is also the mesh with the biggest inequality between the two. For the two rectangular meshes, 1 and 2, the *TNA3D Solver* is giving very similar maximum forces as the FEA for both pinned and rigid joints.

Considering the minimum forces on the other hand, there is a bigger discrepancy between the *TNA3D Solver* and *Karamba* for some of the meshes. Especially for mesh 3 and 4/5, as Figure 5.2 shows. For mesh 3 and 5, the *TNA3D Solver* gets 0.58 kN and 0.67 kN respectively. *Karamba* gets 0.05 kN and 0.17 kN, and 0.09 kN and 0.17 for pinned and rigid nodes respectively. For all the analyses, the *TNA3D Solver* gives equal, or more conservative, results than the FEA. The large differences in minimum forces are not considered a problem in the initial design process.

Through these tests, it have been found that pinned joints in *Karamba* are giving higher axial forces than rigid joints. There is a bigger difference between the TNA and rigid joints, than with the TNA and pinned joints. This is a natural result of the pinned joints not being able to transfer moments between elements; which is closest to the *TNA3D Solver*'s calculation of a funicular structure. With moments and shear forces having a magnitude of less than 1 % of the axial forces, their impact in the structures performance are neglected, and thus not further discussed in this section.

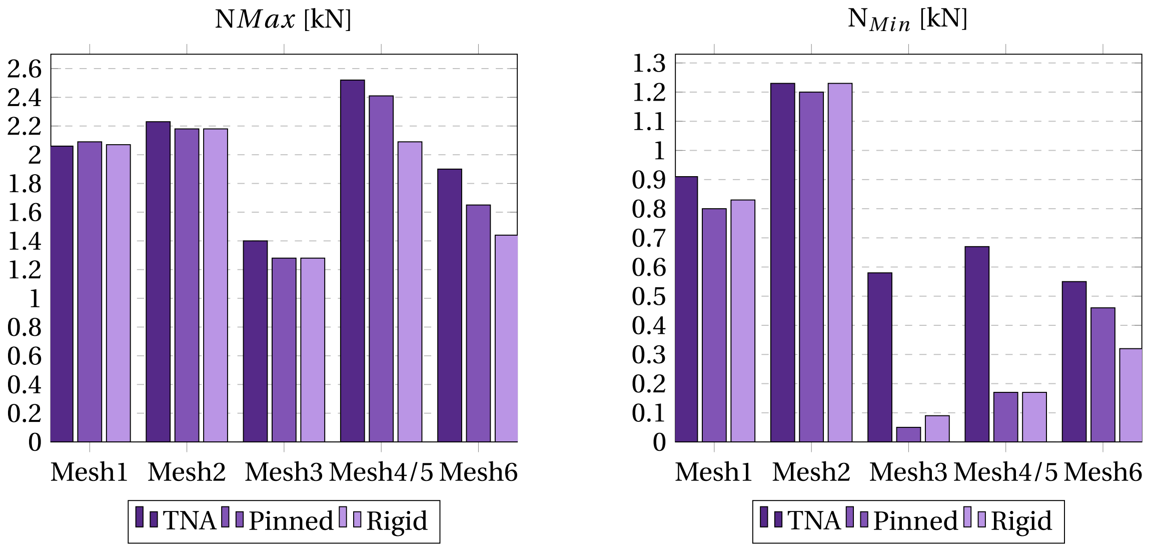


Figure 5.2: Diagrams of maximum and minimum axial forces for all meshes using *TNA3D Solver* and *Karamba* with pinned and rigid joints.

A more complete picture of how the *TNA3D Solver* distributes the forces compared to the *Karamba* results is shown in Figure 5.3 - 5.7. It shows that there is a correlation between the *TNA3D Solver* and *Karamba* when it comes to how the forces are distributed into the branches. The colours of the lines in the figures indicate if the forces are relatively high or relatively low compared to the internal average forces in the branches. From the given figures, it seems like *Karamba* is distributing the forces quite similar for pinned and rigid joints.

For both mesh 1 and 2, the colours in Figure 5.3 and 5.4 are identical for the *TNA3D Solver* and *Karamba* elements. This, combined with the results shown in Figure 5.2, indicates that the *TNA3D Solver* calculates satisfyingly accurate results, assuming that the FEA of *Karamba* is close to the exact solution. Common for these two meshes are that they both have a square mesh geometry.

For the triangular meshes, mesh 3 is quite similar for the *TNA3D Solver* and *Karamba*, see Figure 5.5. However, there are some small differences that indicate how the *TNA3D Solver* distributes the forces compared to *Karamba*. Focusing on the longest diagonal in Figure 5.5, going from the bottom left corner to the upper right corner, it can be seen that the *TNA3D Solver* is mainly carrying the forces through the diagonal, making all the branches connected to diagonal green. *Karamba*, on the other hand, seems to distribute some of the forces into the branches connected to the diagonal, making many of the connected branches yellow. The branches at the start and end of the described diagonal are green in *Karamba*, while they are red in the *TNA3D Solver*.

For the two other triangular meshes, i.e. mesh 4, 5, and 6, it seems from Figure 5.3 and 5.6 that the biggest differences between the *TNA3D Solver* and *Karamba* are located near the supported edges. Here *Karamba* gives higher forces in the diagonal branches than the *TNA3D Solver*.

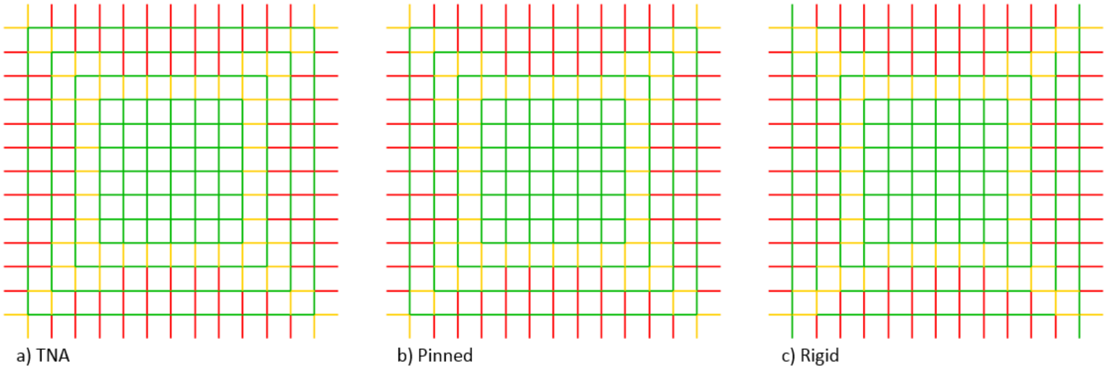


Figure 5.3: Mesh 1. High forces are shown with red colour and low forces with green, for a) TNA, b) Pinned joints and c) Rigid joints.

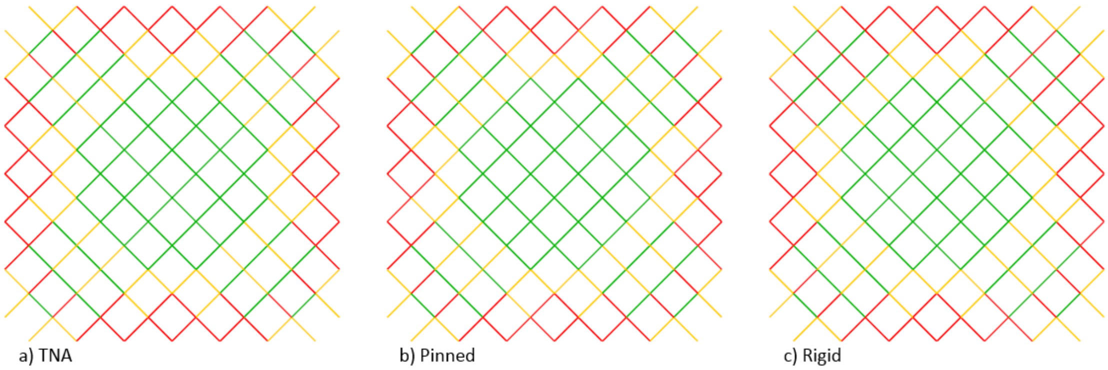

a) TNA b) Pinned c) Rigid

Figure 5.4: Mesh 2. High forces are shown with red colour and low forces with green, for a) TNA, b) Pinned joints and c) Rigid joints.

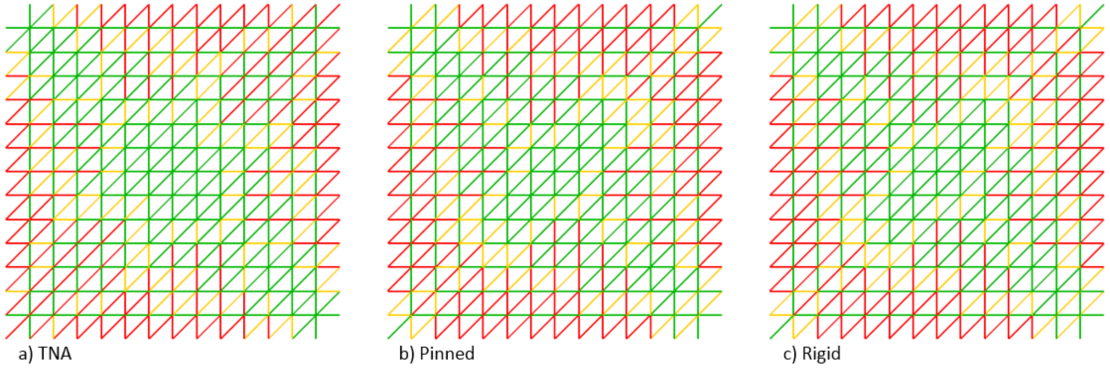

a) TNA b) Pinned c) Rigid

Figure 5.5: Mesh 3. High forces are shown with red colour and low forces with green, for a) TNA, b) Pinned joints and c) Rigid joints.

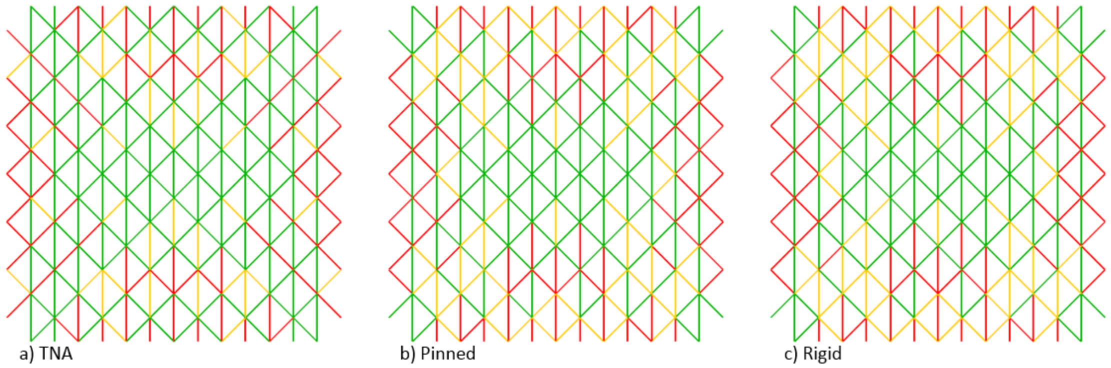

a) TNA b) Pinned c) Rigid

Figure 5.6: Mesh 4/5. High forces are shown with red colour and low forces with green, for a) TNA, b) Pinned joints and c) Rigid joints.

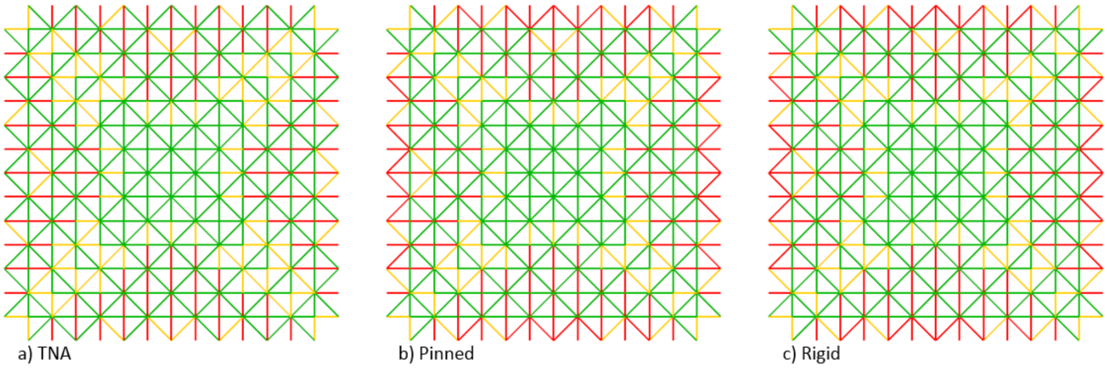


Figure 5.7: Mesh 6. High forces are shown with red colour and low forces with green, for a) TNA, b) Pinned joints and c) Rigid joints.

A disadvantage of the *TNA3D Solver* is that it does not say anything about displacement. Displacement is, therefore, investigated with *Karamba*. All meshes get a maximum displacement of around 1 mm, see Table 5.1. For the two square meshes, i.e. mesh 1 and 2, *Karamba* warns about multiple rigid body modes when using pinned joints. This means that part of the structure can move freely without causing any deformation, which, in turn, explains the very high displacement for mesh 1 and 2. Because of the rigid body modes, the results are not valid. These are caused by the pinned connections in the square meshes without bracing. Mesh 1 and 2 needs additional bracing or semi-rigid/rigid joints to be stable. is that, because of the fewer branches involved, they do not require as much material to be produced. Stability properties of the meshes will be further investigated in section 5.2.

Using rigid joints, the meshes give a more reasonable displacement. The greater quantity of elements, the lower displacements of the elements. The rigid joints yield a smaller displacement for for all the meshes but mesh 3, where the pinned and rigid joints have the same displacement.

Table 5.1: Maximum displacement for pinned and rigid joints. Note that with pinned joints, mesh 1 and 2 experience rigid body modes.

| Mesh | $w_{\text{Pinned}}$ [mm] | $w_{\text{Rigid}}$ [mm] |
|---|---|---|
| 1 | $1.17 \times 10^{12}$ | 0.51 |
| 2 | $2.03 \times 10^{12}$ | 0.82 |
| 3 | 0.33 | 0.33 |
| 4/5 | 1.00 | 0.66 |
| 6 | 0.69 | 0.34 |

This comparison study has shown that the results of the *TNA3D Solver* is quite accurate independent of mesh type. For the square meshes, the results are almost identical to the FEA results from *Karamba*. Overall, the results seem satisfactory considering that it is an initial design phase. Looking at the maximum and minimum forces, the *TNA3D Solver* is conservative compared to *Karamba*. This is not necessarily the case when looking at the location of forces. If the cross section is chosen based on the maximum force, the TNA results will be conservative, but not exaggerated.

### 5.1.2 Curvature

From the 2D study in section 3.6, it was concluded that the height to span ratio of the structure had a large impact on whether a funicular form was achieved or not. This effect is, therefore, further investigated here.

The same basis structure is used here as in the first part of the study, and the results from the *TNA3D Solver* is compared to *Karamba*. Although in this case the structure height is adjusted in order to alter the curvature. The height of the structure is varying from 0.5 metres at its lowest to 7.5 metres at its highest, Figure 5.8. Both rigid and pinned joints are used in *Karamba*, while the shape is obtained with the *TNA3D Solver*.

A ratio factor is introduced in order to make the study relevant for structures with varied dimensions; this is defined as the structure's height divided by half of the shortest span length:

$$r_h = \frac{H}{0.5 * L_{min}} \tag{5.1}$$

Low values of $r_h$ indicate a low curvature, while high values of $r_h$ indicate a high curvature. The structure from the basic case with a height of 3.5 metres has a factor of $r_h = 1.0$.

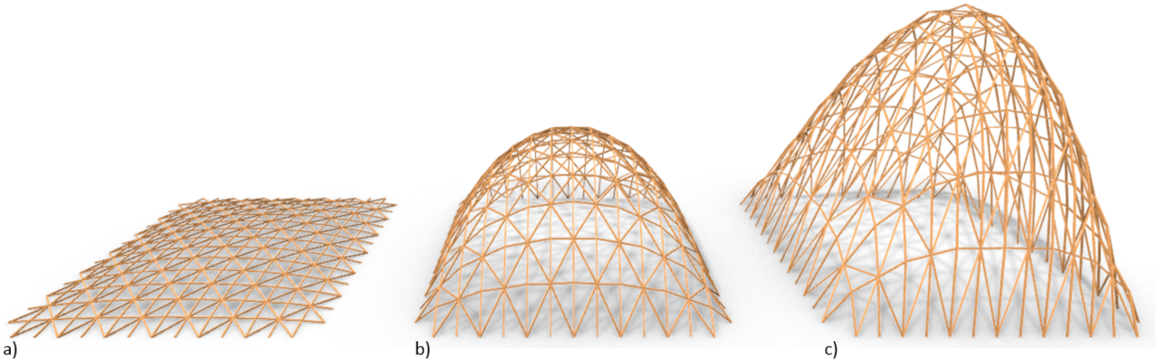


Figure 5.8: Different curvatures illustrated with mesh 6. a) Structure height 0.5 m, $r_h = 0.14$. b) Structure height 3.5 m, $r_h = 1.0$. c) Structure height 7.5 m, $r_h = 2.14$.

### Results

Maximum axial forces are investigated. The ratio of the axial forces obtained with the *TNA3D Solver* divided by the forces obtained with *Karamba* is defined as $r_N = \frac{N_{\text{TNA}}}{N_{\text{Karamba}}}$. $\mathrm{r_N}$ is plotted against $\mathrm{r_h}$ in Figure 5.9 and 5.10, for pinned and rigid joints respectively. Moment and shear forces were less than 1 % of the axial forces, and are therefore neglected when evaluating the results.

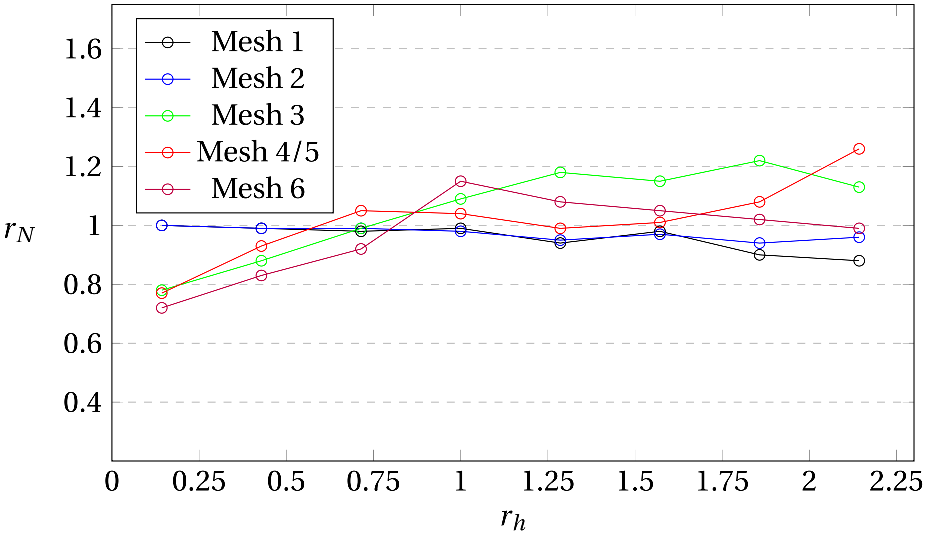


Figure 5.9: FEA with pinned joints. Axial force ratio $r_N$ plotted against curvature ratio $r_h$.

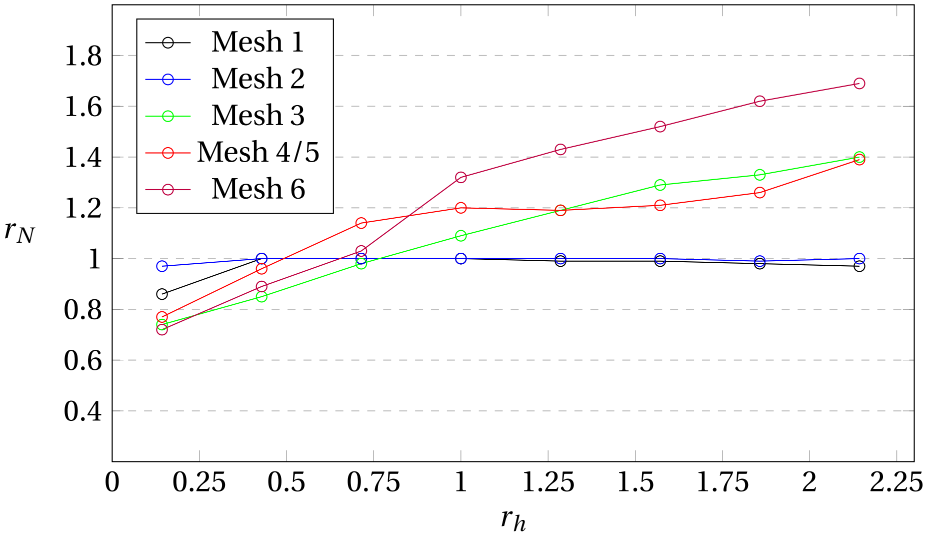


Figure 5.10: FEA with rigid joints. Axial force ratio $r_N$ plotted against curvature ratio $r_h$.

Starting with pinned joints; here the *TNA3D Solver* is most accurate for mesh 1 and 2, which produces precise results for almost all curvatures. The other meshes yield the most even results when $\mathrm{r_h}$ is in the range of 0.75 to 1.60.

Then looking at rigid nodes; the *TNA3D Solver* seems to be the most accurate compared to the FEA when $\mathrm{r_h}$ is between 0.5 and 0.8 for all meshes. While mesh 1 and 2 give almost identical results as the FEA independent of the curvature, it seems to be an almost linear relation between $\mathrm{r_h}$ and $\mathrm{r_N}$ for the other meshes.

As in the first part of the study, the TNA results for mesh 1 and 2 are the most precise compared to *Karamba* independent of curvature and connection type. For both the *TNA3D Solver* and *Karamba*, with pinned joints, analysis the axial forces are increasing for high curvatures.

In general, for triangular meshes, the *TNA3D Solver* has a tendency of predicting too low axial force for low curvatures, and opposite for high curvatures; while the accuracy of quadratic meshes are less influenced by the curvature.

### 5.1.3 Summary

The objective of this initial case study was to investigate the accuracy of the *TNA3D Solver* compared to FEA for a simple quadratic dome; the effect of curvature was also investigated. The case study discovered the following:

- The *TNA3D Solver* found a pure compression equilibrium for all meshes.
- The internal forces from the *TNA3D Solver* correlated well with FEA, independent of mesh type. However, the maximum axial forces were more accurate than the minimum forces. Also, the force distribution was at a satisfying level.
- Using pinned joints for the FEA gave more similar results as the TNA than by using rigid joints.
- Rectangular meshes gave the most accurate results, but the pinned joints need bracing in order to be stable. The accuracy for neither pinned joints nor rigid joints were affected by the change of curvature.
- For pinned joints, the *TNA3D Solver* was most accurate for a curvature ratio $r_h$ between 0.75 and 1.6; for rigid joints, the accuracy was at its highest for $r_h$ between 0.5 and 0.8.

## 5.2 Case study II: Buckling of quadratic dome

In this case study of buckling, several different buckling aspects will be tested to highlight what is important to consider when evaluating a system's stability. In addition to the simplified buckling analysis as described in chapter 4.8, the following studies will be tested:

- The effect of choosing pinned or rigid joints, with varying cross sections
- The curvature of the structure
- Horizontal loading
- A simplified 2D buckling analysis of a 3D problem

The case study will create a solid foundation for the understanding of how gridshells buckle, and which aspects one should consider when evaluating such systems.

### 5.2.1 Rigid vs pinned joints

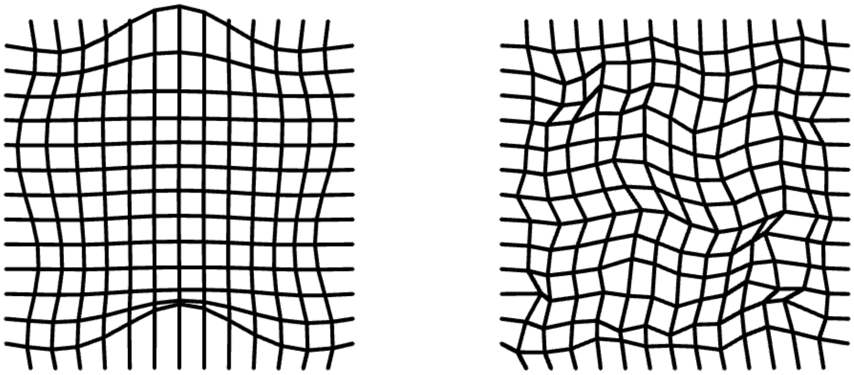


Figure 5.11: Buckling shapes for gridshell with rigid joints (left) and pinned joints (right).

In reality, joints will be semi-rigid and lie somewhere between rigid and pinned. The difference in joints will have a big impact on the structural behaviour, as shown in Figure 5.2.1. By analysing rigid and pinned joints, the two extremes will be tested. This section will, therefore, provide a guide as to what type of joint would be the most beneficial to choose (near pinned or near rigid) based on different mesh types.

The same meshes as in Case study I, section 5.1.1, are used for this case study as well. The base area is 7x7 metres, and the structure height 3.5 metres. The form is found with *TNA3D Solver*, and the buckling analysis is subsequently performed with *Karamba*. Each mesh is analysed with quadratic elements having a cross section width varying between 35-120 mm. The scaling factor *d* in the *TNA3D Solver* is adjusted for each case, so that the structure height remains constant. The difference between $P_{cr.rigid}$ and $P_{cr.pinned}$ for all cross sections are shown in Figure 5.13. The results from using a cross section of 35x35 mm is enlarged in Figure 5.12.

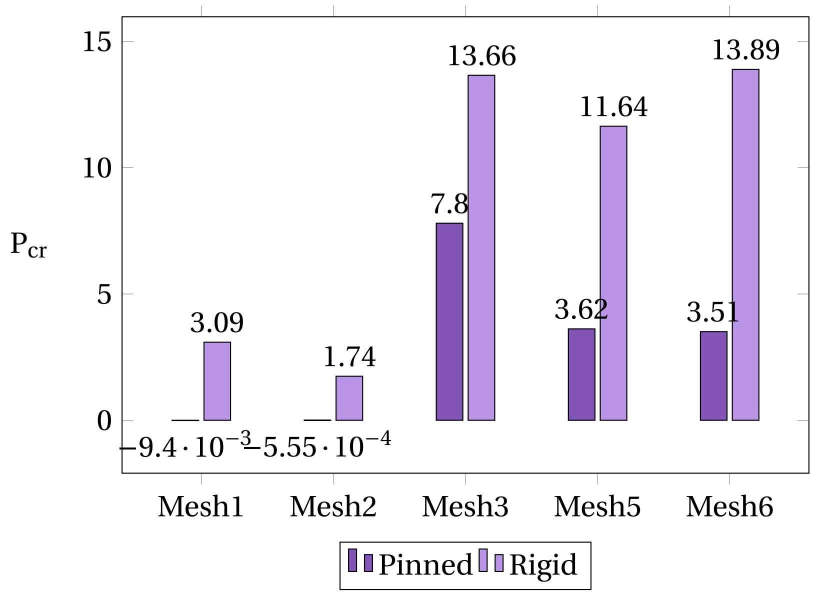


Figure 5.12: Critical buckling factor $P_{cr}$ for different meshes; these were analysed using rigid and pinned joints with *Karamba*.

Figure 5.12 shows a clear pattern. Mesh 1 and 2, which are the two rectangular meshes, have no resistance against buckling if the joints are pinned. They would need either bracing or joints that have rotational stiffness. The diamond-shaped mesh 2 also had a smaller resistance against buckling than mesh 1. Choosing rectangular grids can be more material effective, but the joints can be both expensive and hard to acquire when a rotational stiffness is needed.

All the triangular meshes display better results than the rectangular ones. Even with pinned joints, the triangular grids are stronger than pinned rectangular grids. This shows the benefit achieved when using triangular-shaped meshes in gridshells. Although the capacity of the triangular grid is doubled when choosing rigid joints, the results show that mesh 3 - 6 do not need much stiffness in the joints. This demonstrates what has already been discussed in section 1.4.

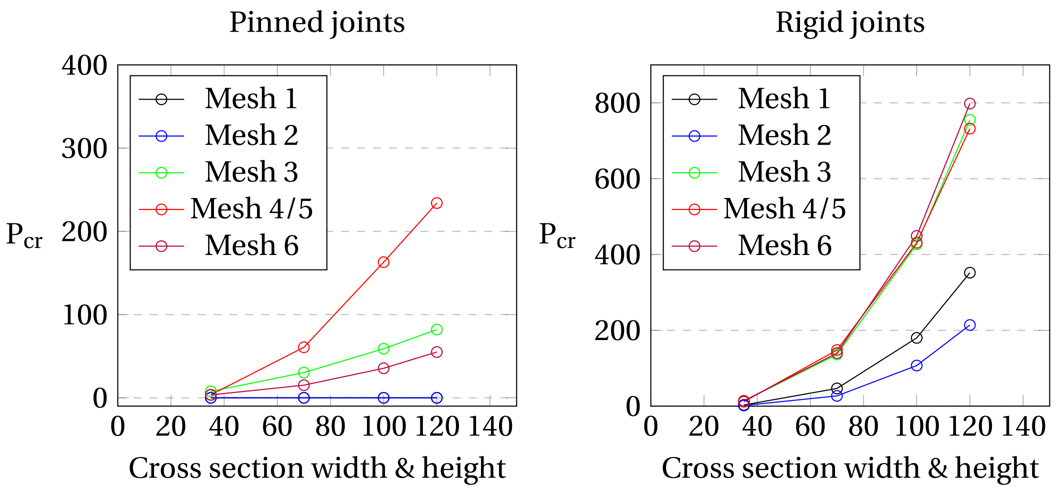


Figure 5.13: Critical buckling load $P_{cr}$ analysed for different cross sections evaluated with pinned and rigid joints.

The plots in Figure 5.13 show how the buckling strength increases as the cross section is enlarged; for the rigid joints, this effect is significantly greater than that of the pinned joints. Considering that the numbers are usually very small, it shows that only a small change of the cross section will have a large impact on the buckling capacity. All three triangular grids perform almost identical, meaning that one can choose the one which is the most material effective among them and as a result not influence the buckling capacity to any significant degree. When it comes to the rectangular grids, mesh 1 performed better than mesh 2.

For the pinned joints, the graph shows that no matter how large the cross section becomes, the rectangular meshes will buckle if the structure is not braced. Additionally, the improved stability achieved by choosing a larger cross section is lower than for rigid joints. What this in essence signifies is that for stronger joints, a smaller cross section can be chosen.

### 5.2.2 Curvature

In this section the effect of curvature on the stability of the structure is studied. Again, the same geometries as in section 5.1 apply: a plane of 7x7 metres as a base surface with element cross sections of 35x35 mm. The height of the structure varies from 0.5 m to 6.5 m, and the critical load factor obtained from *Karamba* is plotted against $r_h$ in Figure 5.9 and 5.10, for pinned and rigid joints respectively. Only the triangular meshes are evaluated with pinned joints, because of the instability problems of mesh 1 and 2 discussed in section 5.1 and 5.2.1. $r_h$ is the same as explained in 5.1.

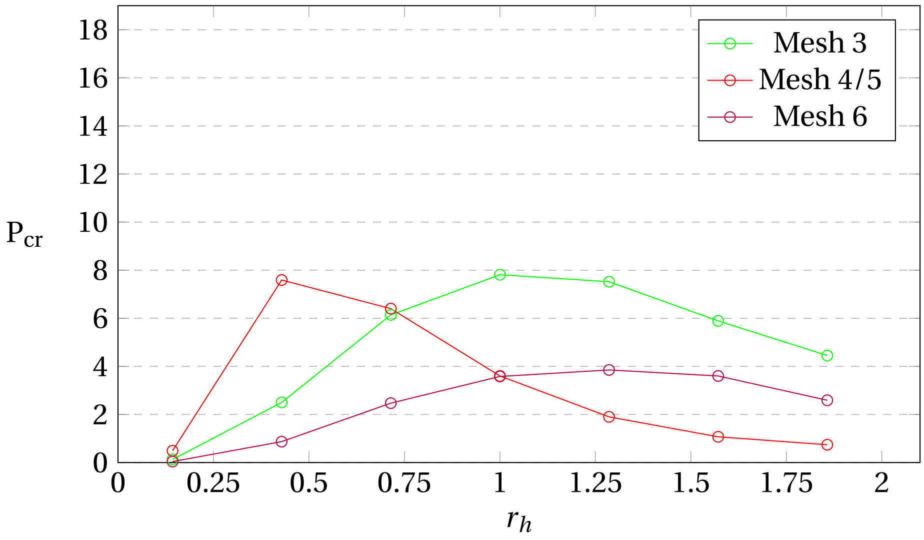


Figure 5.14: FEA using pinned joints: Critical load factor plotted against curvature ratio $r_h$.

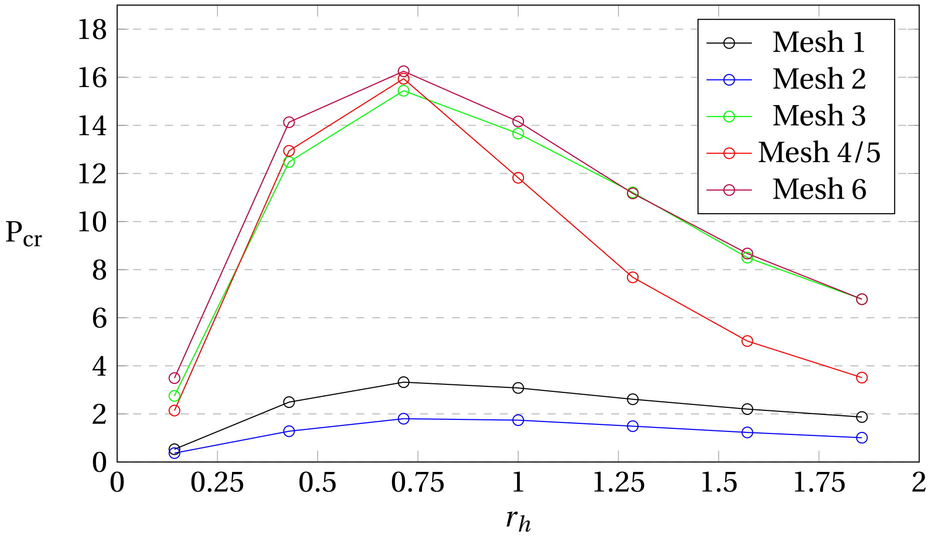


Figure 5.15: FEA using rigid joints: Critical load factor plotted against curvature ratio $r_h$.

Figure 5.14 shows the critical load factor for pinned joints. $P_{cr}$ is at its highest when $r_h$ ≈ 0.7 for mesh 4/5, $r_h$ ≈ 1.0 for mesh 3, and $r_h$ ≈ 1.25 for mesh 6. If pinned (or weak semi-rigid) joints are to be used in a structure, then a curvature between 0.7 and 1.25 is recommended, as it produces the strongest solution. This also shows how the structure benefits from the correct curvature.

For the rigid joints, all the meshes have its highest critical load factor when $r_h \approx 0.7$, as Figure 5.15 shows, meaning that all the meshes are most stable at this level of curvature. Both for pinned and rigid joints the triangular meshes are most affected by the curvature. The rectangular meshes will have a more stable value, showing that the chosen curvature is not as important for those grids. The value of $r_h$ is recommended to be 0.7 if the joint type is rigid or nearly rigid.

### 5.2.3 Horizontal loading

As shown in section 2.3.3, only vertical forces are implemented in the *TNA3D Solver*. Gridshells are, however, in most cases also subjected to horizontal loading, such as wind. This section will, therefore, analyse how horizontal loads affect the global stability.

A conservative value of $3\text{kN/m}^2$ is applied as lumped nodal loads to half of the structure. The load area is calculated as the plane side-area of the structure. All nodes are also subjected to a force in z-direction of $2\text{kN/m}^2$. All structures are calculated with *Karamba* using beams with rigid joints. The results below are shown with and without wind load.

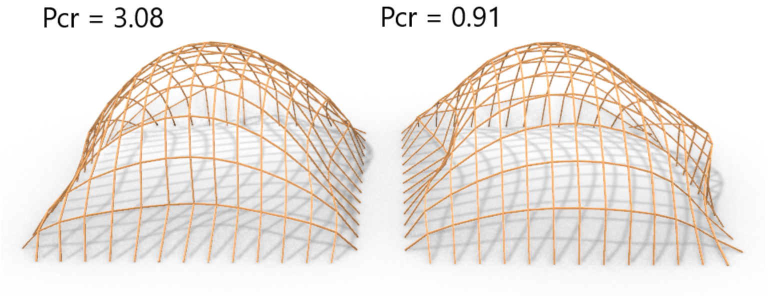


Figure 5.16: Mesh 1: Critical load factor. Without wind to the left, and with wind to the right. Buckling scale factor = 0.06.

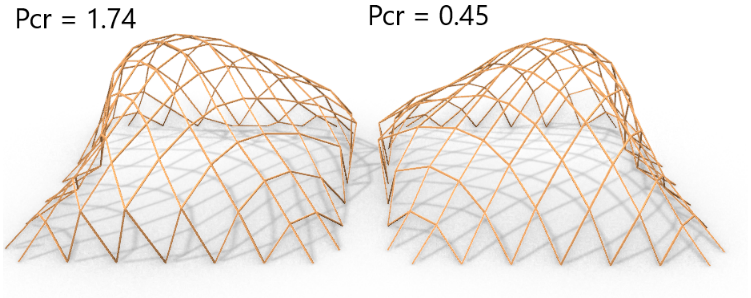


Figure 5.17: Mesh 2: Critical load factor. Without wind to the left, and with wind to the right. Buckling scale factor = 0.06.

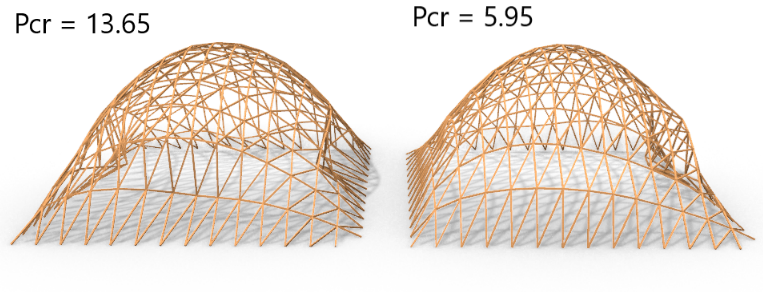


Figure 5.18: Mesh 3: Critical load factor. Without wind to the left, and with wind to the right. Buckling scale factor = 0.15.

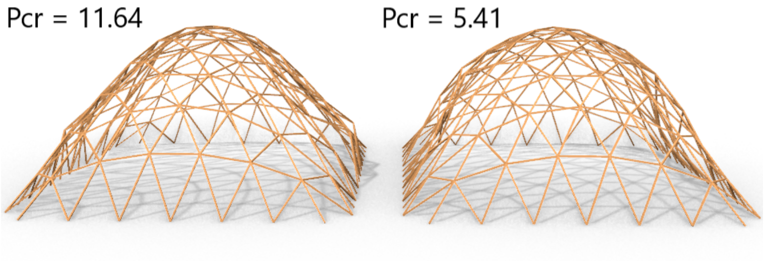


Figure 5.19: Mesh 4/5: Critical load factor. Without wind to the left, and with wind to the right. Buckling scale factor = 0.5.

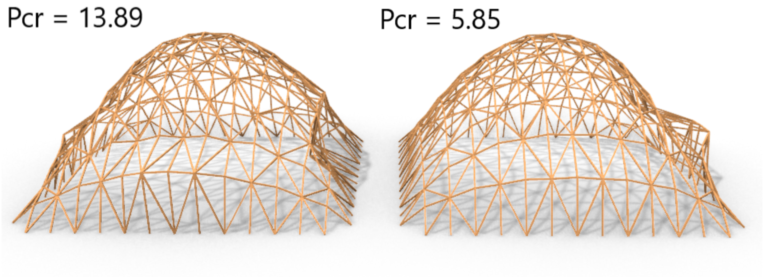


Figure 5.20: Mesh 6: Critical load factor. Without wind to the left, and with wind to the right. Buckling scale factor = 0.15.

The results show that the buckling factor is highly affected by the wind load. Just like the case with only vertical loads, all the triangular grids behave similar. The buckling factor was twice as large without the wind load. This illustrates how the global stability is highly dependent on the horizontal forces; especially for the rectangular gridshells, where the strength was 3-4 times greater without the wind load.

The developed TNA components do not take into consideration the horizontal loads, as a results, this effect is not implemented in the 2D buckling estimate. However, the effect has such an impact that it should be considered during the two dimensional buckling estimates. Based on the results from Figures 5.16-5.20, a reduction factor of 0.5 is recommended for the triangular grids, and 0.25 for the rectangular grids respectively.

### 5.2.4 2D simplification

This section aims to evaluate the 2D buckling components described in 4.8. First implemented with bar elements, then with beam elements. If a 2D simplification can be a viable option, the designer the designer can evaluate the global stability without need to establish an FEA first. This study will create the foundation for further development of the components presented in this thesis.

Again a 7x7 metre rectangular dome with height 3.5 metres, as in section 5.1.1, is evaluated. Meshes 1-6 will be tested. The results of each component are compared against the *Karamba* analysis using beam elements and rigid joints.

**Buckling 2D Bar**

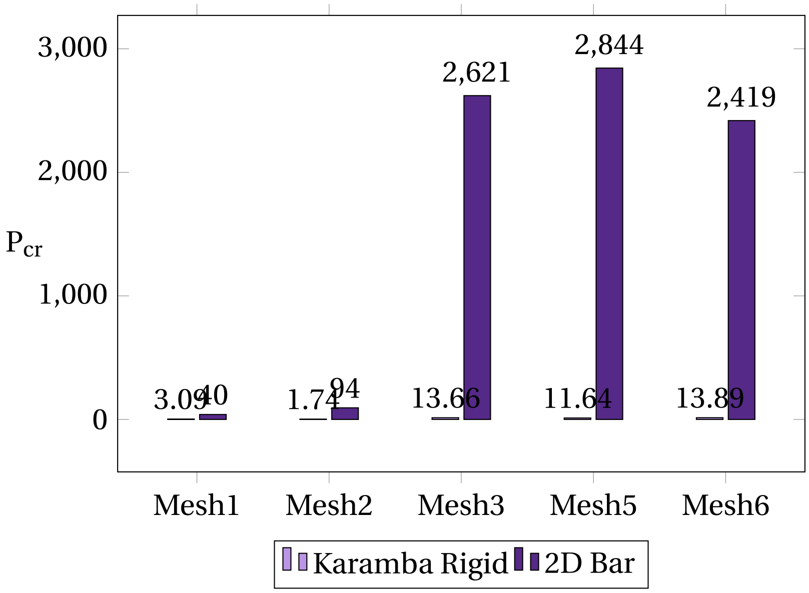


Figure 5.21: Critical buckling factor for different meshes; which is analysed with rigid beams in *Karamba* and *Buckling 2D Bar*.

The results using *Buckling 2D Bar* had unrealistically high values. The reason is probably due to the fact that the structure now acts as a truss, i.e. a very stiff construction. All displacements are in-plane, and with out-of-plane buckling being restrained, the projected structure simply becomes too stiff.

### Buckling 2D Beam

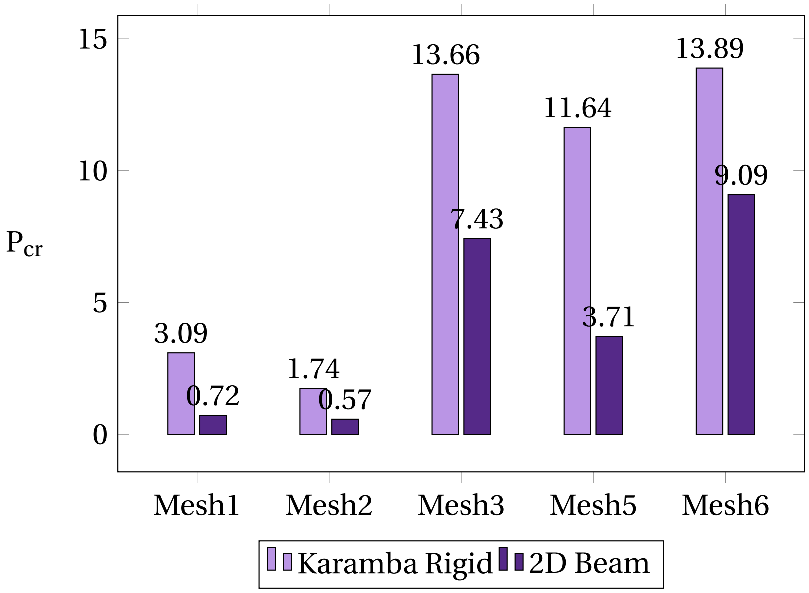


Figure 5.22: Critical buckling factor for different meshes; which is analysed with rigid beams with *Karamba* and *Buckling 2D Beam*.

Figure 5.22 shows that the results using *Buckling 2D Beam* have realistic values, in contrast to the *Buckling 2D Bar*. By projecting the beams down in the plane, the positive effects from the 3D curvature are not accounted for, rendering the solution weaker than the FEA result. The overall evaluation is that the component can be tested further.

Going further with the *Buckling 2D Beam* beam component, a series of height to span ratios are tested as in section 5.1.2. The cross section of the elements is 35x35 mm. A ratio factor is introduced as the critical buckling factor from the 2D estimate divided by the factor from the FEA:

$$r_{P.cr} = \frac{P_{cr.2D}}{P_{cr.3D}} \tag{5.2}$$

Where the critical buckling factor for the 3D structure is obtained from a second order analysis in *Karamba*.

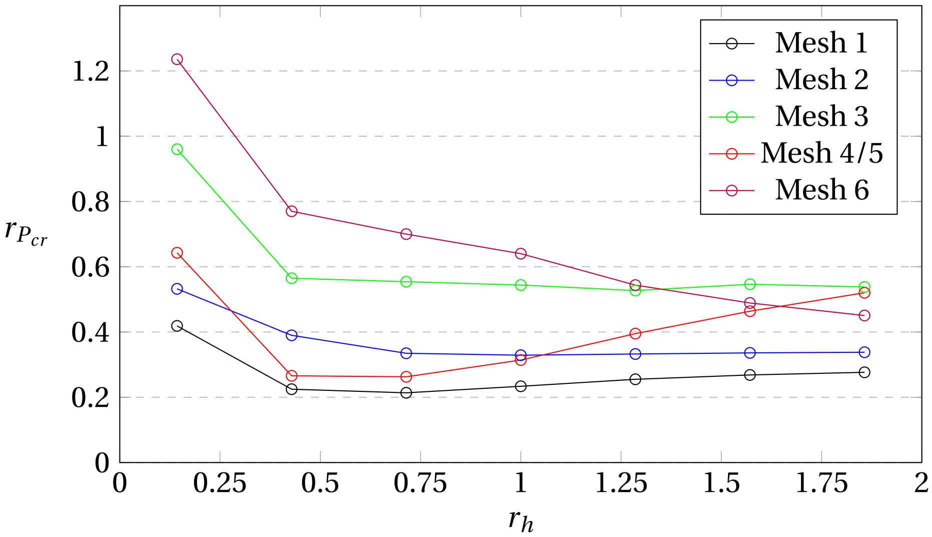


Figure 5.23: Ratio factor $\mathrm{r}_{\mathrm{P}_{\mathrm{cr}}}$ plotted against curvature ratio $r_h$. These results are obtained using the *Buckling 2D Beam* component.

Figure 5.23 shows promising results; for mesh 1, 2. and 3, $r_{Pcr}$ has a stable value when $r_h$ > 0.4; while mesh 4/5 and 6 had more varying values. All the results lies between 0.2 and 0.8 for $r_{Pcr}$.

As for the rectangular dome in this case study, the simplified 2D buckling analysis can give a good estimate of the $\mathrm{P}_{\mathrm{cr}}$ obtained from an FEA if the buckling factor is adjusted by the following equation:

$$Pcr_{estimate} = \frac{Pcr_{2D}}{r_{Pcr}} * r_{Wind} \tag{5.3}$$

where $r_{Pcr}$ = 0.22 for rectangular grids, since they have a $r_{Pcr}$ between 0.22 and 0.33. For mesh 3 the recommended value is $r_{Pcr}$ = 0.55. With regard to mesh 4/5 and 6, it is, on the other hand, recommended to choose a $r_{Pcr}$ based on the curvature. This can be integrated in the code of the component. From the horizontal load analysis, the value of $r_{Wind}$ is recommended to be 0.5 for triangular grids and 0.25 for rectangular grids. For all meshes, the curvature $r_h$ must be larger than 0.4 for the simplification to yield a good estimate.

### 5.2.5 Summary

The purpose of this study was to investigate how different factors influenced global buckling, before the accuracy of the 2D evaluation of global stability was tested. Varying meshes, cross sections, curvatures, and joints have been tested. This case study found the following:

- In agreement with previous studies mentioned in chapter 1.4, we found that pinned rectangular meshes will buckle if they are not braced; triangular meshes are more stable.
- Both triangular and rectangular meshes will benefit from having rigid joints, and the larger the cross section the higher the benefit; therefore, the results from the rigid joint and the pinned joints is most coinciding for the smaller cross sections.
- Triangular meshes perform very similarly both with and without the wind load. From a structural perspective, the one that is the most material effective should be chosen.
- The curvature of the structure will have a large impact on the global stability. With rigid joints it was found that the a $r_h = 0.7$ gives the strongest solution, whereas for the pinned joints the solution was strongest with a curvature ratio between 0.7 and 1.25. An interesting finding is that rectangular meshes are not as affected by the curvature as the triangular meshes are.
- When introducing wind loads the buckling capacity of the structure decreases significantly. Since the TNA does not consider horizontal forces, the designer should keep in mind that the solution will be non-conservative if there are large horizontal forces present.
- A 2D simplification of the 3D problem was investigated. When using pinned joints the solution had unrealistic values and showed little potential.
- The 2D beam simplification from *Buckling 2D Beam* displayed promising results, and it would be interesting to investigate it further for a larger variety of gridshell geometries.

## 5.3 Case study III: Public square roof

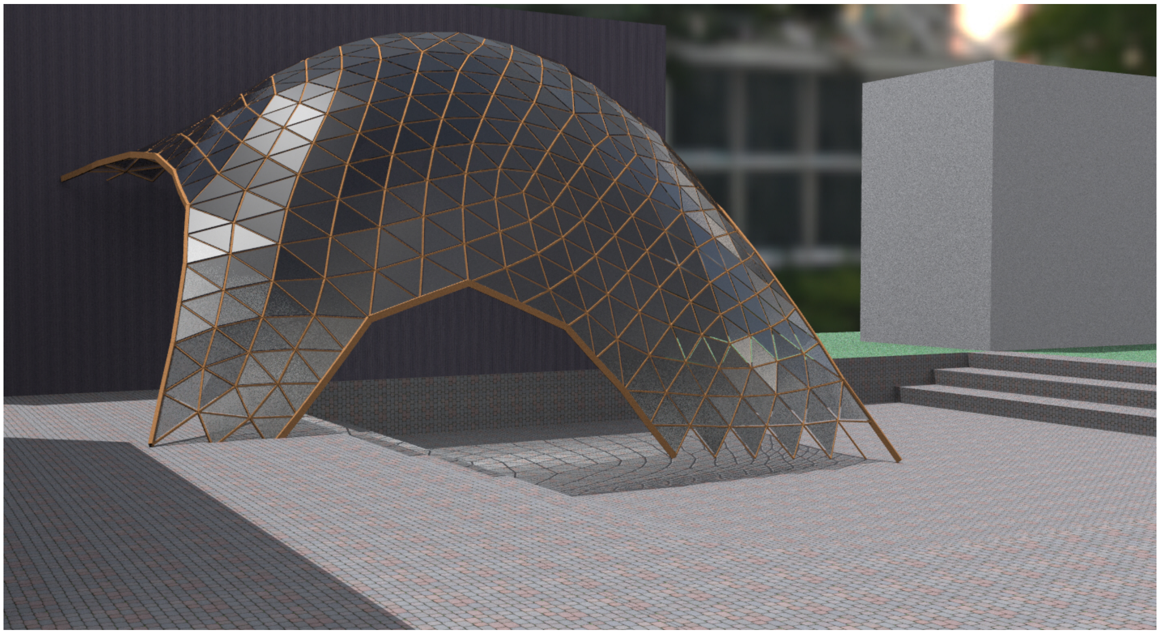


Figure 5.24: Illustration of the gridshell developed through the case study.

In this section a structure with both free edges and support points at multiple heights is created. Starting by creating and evaluating an initial shape with the components developed in chapter 4, the resulting geometry is then verified by Karamba. The initial results are then used in an optimisation process where the objective is to preserve the initial TNA-shape while at the same time keeping the branch lengths as equal as possible; this is done out of practical concerns regarding assembly and manufacturing.

**Objectives and specifications**

The objectives in this case study are to evaluate the developed tools' performance when subjected to arbitrary shapes and skewed geometries. Furthermore, the functionality of the *Camel* tools coupled with optimisation methods such as *Galapagos* is tested when going from conceptual design to optimisation of the structure.

A roof spanning over a public square shall be designed in such a way that it is supported both at a ground level, and towards a wall. The ground area should be approximately 15x25 metres, with a maximum height of 15 metres. In order to preserve the possibility to move freely beneath the roof, several openings need to be created. The list of specifications is given below:

- Maximum height: 15 metres.
- Ground surface: 15 x 25 metres.
- Loading: Self weight and a area load of 2kN/m$^2$ lumped onto the nodes.
- Material: Timber, C30, $\sigma_{\text{fc,0,k}}$= 24MPa.
- Initial cross section of internal branches: 35 mm x 35 mm.
- Three openings should be created.

### 5.3.1 Creating the roof

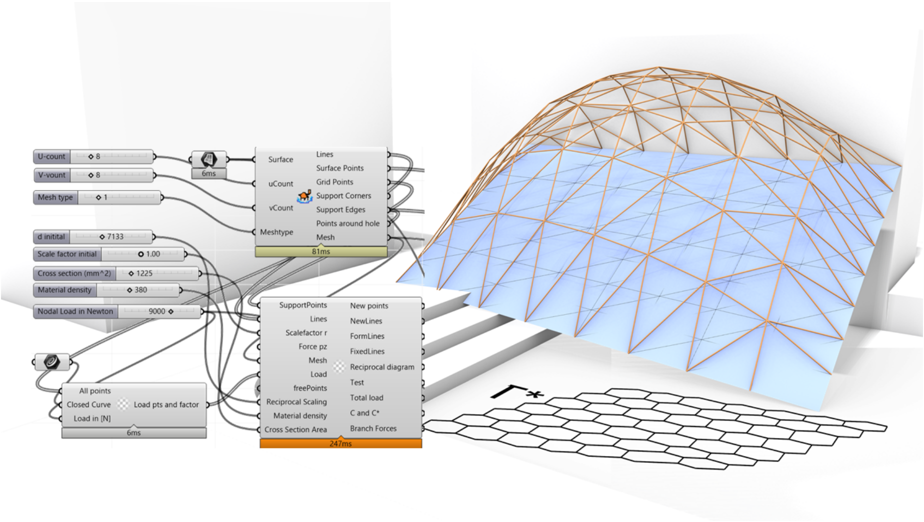


Figure 5.25: Initial results from the first TNA. The blue surface is the input surface modified in Rhino. All input variables are shown.

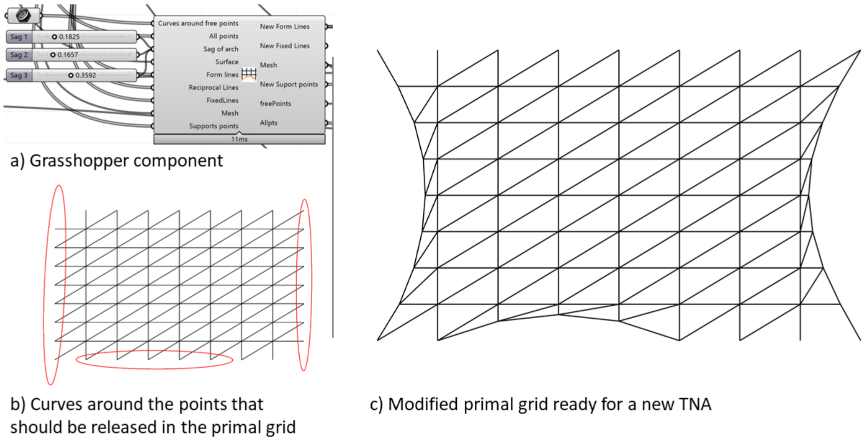
Starting with the area of which the roof should cover, a planar rectangular surface of 15x25 metres is drawn in the Rhino environment. The control points of the surface are then moved vertically to achieve different heights on the surface. This is the only geometry from Rhino used as input for the Grasshopper components, Figure 5.25. The *Mesh generator* component is set to mesh 3 with a u- and v-division of 8, resulting in a total of 174 branches in the initial primal grid. The course division for the mesh is chosen to obtain a solver time for the component that enables for the exploration of geometry with close to real-time feedback. The triangular mesh is selected based on the results from case studies in section 5.1 and 5.2. The evenly distributed vertical load results in external loads of approximately 9 kN per node.

The resulting geometry here is then used to run the first *TNA3D Solver*. As Figure 5.25 shows, all the edges are initially fixed to the surface edges. The highest point of the gridshell is approximately 15.2 metres.

From the case studies in 2D section 3.6, a height to span ratio above 0.5 was recommended in order to get a funicular solution. With a tilted base surface the distribution of forces will no longer be symmetric, and the lower edges will attract larger forces. Knowing that open edges will be present as well, and thus resulting in fewer support points with larger concentration of forces, a conservative height to span ratio of around 1.0 is used.

For the initial geometry, the solver is able to find a solution with the target height of 15 metres when the reciprocal scaling is in the region of 7000. The factor is used in the linear optimisation in order to create the reciprocal diagram. Eq. (2.38) uses $d$ as a constraint for the branch lengths lower bounds. Working with a mesh with branches ranging from around 1000 to 6000 mm, the reciprocal scaling needs to be above 1000 for the solver to find feasible solutions.

With the first shape established, the *FreeEdge* component from section 4.6 is used to release the edges. From the "top view" perspective in Rhino, curves are drawn around the points that are free to move. Together with the results from the first TNA, the component creates a new mesh and lines to be used in a new *TNA3D solver* as shown in Figure 5.26. Having created a new mesh the scale factor $r$ and reciprocal scaling $d$ have to be adjusted in order to obtain a desired shape. The final geometry of the initial gridshell with three openings are shown in Figure 5.27.



a) Grasshopper component

b) Curves around the points that should be released in the primal grid

c) Modified primal grid ready for a new TNA

Figure 5.26: Using the *FreeEdge* component, the primal grid is modified by drawing curves around the new free points.

### 5.3.2 Forces and stability from TNA

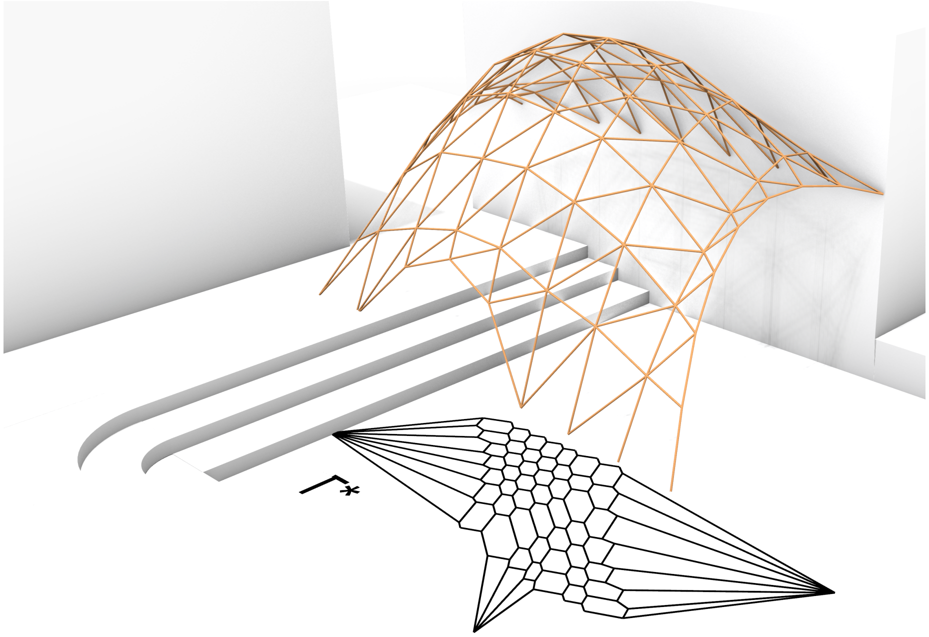


Figure 5.27: The obtained shape of the gridshell using the TNA tools.

When a satisfying geometry is achieved, the structural performance of the gridshell has to be evaluated. The capacity controls from the TNA are first used to get an indication of the gridshell's behaviour.

**Axial Forces**

The axial forces in each member, calculated from equation 2.28, are given as a list from the TNA solver. The resulting compression stresses vary from 5.17 MPa to 147.99 MPa. Intuitively, this seems like very large stresses, and looking at the utilisation of the selected cross sections in Table 5.2, the maximum ratio is far from acceptable. Examining the reciprocal diagram illustrated in Figure 5.27, the branches from the three free edges are significantly larger than the original branches. By splitting the results into arch branches and internal branches the results are more like the ones expected. The maximum utilisation for the internal branches are still above 1.0 but closer to the material capacity than at the arches. With the nodal loads being identical in each branch, the reciprocal diagram gives a clear visualisation of the total forces in each branch. Therefore it gives users without knowledge of statics an indication of potential problem areas in the design.

**Buckling**

With the members having a constant cross section of only 35 mm x 35 mm and branches up to 6 metres long, then it comes as no surprise that the *Capacity Check* component indicates Euler buckling for almost every single element, which means that the critical buckling load $P_{Cr}$ from equation 2.41 is lower than the material capacity.

Table 5.2: Maximum forces in the initial gridshell created with *TNA3D Solver*.

| | $\sigma_{\max}\ \frac{N}{mm^2}$ | $\sigma_{\min}\ \frac{N}{mm^2}$ | UR | Local buckling? |
|---|---|---|---|---|
| Internal Branches | 30.45 | 5.17 | 1.27 | Yes |
| Arch branches | 146.99 | 34 | 6.12 | Yes |

Despite knowing that the branches will fail, the global stability is still evaluated by the *Buckling 2D beam* for the comparison with *Karamba*'s stability analysis. As explained in section 4.8, the global geometry is projected down to the planar surface, which in this case is tilted, before the stiffness relations from equation (2.55) are established. The eigenvalues are then solved from (2.59).

The value of interest from the component is the lowest critical value. Being the factor for which the applied loads should be multiplied with for the system to buckle, a factor smaller than 1.0 indicates an unstable structure. The lowest $P_{cr}$ from the component is 0.0046, hence the structure is highly unstable. Both the distribution of stresses and buckling shapes generated by the *Camel* components seem plausible at first glance. From the second case study in section 5.2 the *Buckling 2D beam* could be expected to be around 30 % of the global stability analysis in *Karamba*, meaning that the structure would still be highly unstable. Furthermore, the cross sections of the elements are too small, especially in the arches creating the free edges.

### 5.3.3 Initial finite element analysis

A verification of the results from the above section is produced by an FEA. The geometry, support conditions, and nodal loads are identical to the TNA-analysis, while the self weight is applied as an evenly distributed load for simplicity's sake. With a total structure weight of only 2,9kN and total external loads of 594kN this approach is assumed to make an insignificant difference compared to the lumping of self weight in the TNA.

**Forces and utilisation**

Initially the system is analysed with the nodal connections between branches set to completely rigid. This is the standard connection between elements in most computational programs; making it the easiest to use as well. Since the elements then are able to transfer moments, the axial forces are assumed to be lower than the ones suggested by the *TNA3D Solver*.

An analysis with pinned joints is, therefore, also introduced to evaluate the effect of rigid joints. From the analysis in section 3.4 and 3.5, the arches become statically under determined if hinged; consequently, the connections in the free edges have to remain rigid in order to avoid rigid body modes in the system.

Table 5.3: Results from first order analysis. Compression forces in *Karamba* are displayed with negative sign in front. The axial tension in the tables includes the contributions from the resultant moments.

| | Rigid Joints | Pinned Joints |
|---|---|---|
| $N_{\text{Arch.max}}$ [kN] | −41.06 | −42.07 |
| $N_{\text{Arch.min}}$ [kN] | −164.41 | −176.25 |
| $N_{\text{Int.max}}$ [kN] | 1.86 | 1.23 |
| $N_{\text{Int.min}}$ [kN] | −53.34 | −46.87 |
| $M_{\text{Res.max}}$ [kNm] | 0.69 | 0.41 |
| $V_{\text{Res.max}}$ [kN] | 0.50 | 0.3 |
| $M_{\text{t.max}}$ [kNm] | 0.3 | 0.22 |
| $\sigma_{\text{x.Arch}}$ $[\frac{N}{mm^2}]$ | -174.0 | -195.0 |
| $\sigma_{\text{x.Int}}$ $[\frac{N}{mm^2}]$ | -107.0 | -38.3 |
| $UR_{\text{Arch}}$ | 7.25 | 8.13 |
| $UR_{\text{Int}}$ | 4.56 | 1.61 |

One of the advantages with the *Camel*-tools is the sorting of the lines when free edges are created; with the free edges appended to the end of the list, a single list split at the index equal to the original number of branches renders the two list necessary to apply the wanted nodal connections. Apart from the internal hinges, the procedure is the same for the two approaches; the results illustrated in Table 5.3.

By examining the results from *Karamba*, the magnitude of axial forces for both rigid and pinned joints correspond remarkably well with the TNA results. Starting with the pinned joints, which, as expected, is closest to the *TNA3D Solver*, the discrepancy in maximum axial forces in the arches is only 2 %, Table 5.4; whereas for the rigid joints the FEA yields a maximum compression force 9 % lower than TNA.

Upon investigation of the internal elements the discrepancy between the TNA and the FEA is as high as 43 % for the rigid joints and 26% for the pinned ones. Upon further investigation, it is found that anomaly is caused by a single element connected to a support node attracting larger forces for both pinned and rigid branches. Disregarding this element, the forces in the internal elements from *Karamba* varies from the TNA with a size similar to the arches.

Table 5.4 summarises the differences between results produced by the two methods. The normal forces are, as expected, relatively similar, which the previous case studies both in 2D and 3D also showed. Beside minor tensile forces present in two of the 174 branches with magnitudes smaller than 2 kN, the obtained shape is close to funicular. The 7.5 times higher solution from *Karamba* compared to the *TNA3D Solver* for the axial stresses are caused by the small moments and local singularities near the supports. With the applied cross sections even a small moment of 0.7 kNm will add additional normal stresses with a magnitude of 98 MPa, which, in turn, again causes the large deviance between the TNA and the FEA.

Table 5.4: Relative solution value of FEA compared to the TNA.

| | TNA | Pinned FEA | Rigid FEA |
|---|---|---|---|
| $\epsilon_{\text{N.max.Arch}}$ | 1.00 | 0.98 | 0.91 |
| $\epsilon_{\text{N.max.Int}}$ | 1.00 | 1.26 | 1.43 |
| $\epsilon_{\sigma\text{.x.Arch}}$ | 1.00 | 1.32 | 1.18 |
| $\epsilon_{\sigma\text{.x.Int}}$ | 1.0 | 7.5 | 3.14 |

**Global stability with Karamba**

Results from 5.3.2 indicated element buckling for almost every branch in the gridshell as well as a $P_{cr}$ of 0.0046; thus the FEA is also expected to generate an unstable structure.

In order to get the critical buckling loads, and the corresponding buckling shapes, a second order analysis with small deflections is performed. The resulting critical factor is 0.0074 with a buckling shape corresponding to failures in the arches where the stresses are highest. This is very much the same indications given by the *Buckling 2D beam* component in section 5.3.2. The critical force from the projected 2D beam solution is 62 % of the FEA. From Table 5.23 in Case study 2, Mesh 3 gave values around 55 % of the FEA. With both a tilted surface and three openings the prediction from the 2D solution in this case study is not that far apart. Due to the extreme instability of the structure, no other conclusions could be drawn from this.

**Summary of initial analysis**

After analysing the shape it becomes clear that the number of branches as well as the cross sections are far from ideal; this in regard to both material capacity and global stability. The created shape, however, shows potential as a funicular and material effective solution. The next section investigates how the initial form and topology obtained by TNA may be used in a sizing optimisation of the elements.

### 5.3.4 Sizing optimisation of the gridshell

**Remeshing the surface**

The obtained mesh from section 5.3.1 is first resized using plug-ins from *Kangaroo*. Starting with a simple re-mesh for the purpose of getting even face-sizes, after which these are then subdivided once to approach the wanted edge length of 1 metre. This new mesh is then smoothed using a dynamic solver to achieve smaller diversity in line lengths. The resulting mesh is displayed in Figure 5.28 together with the original one.

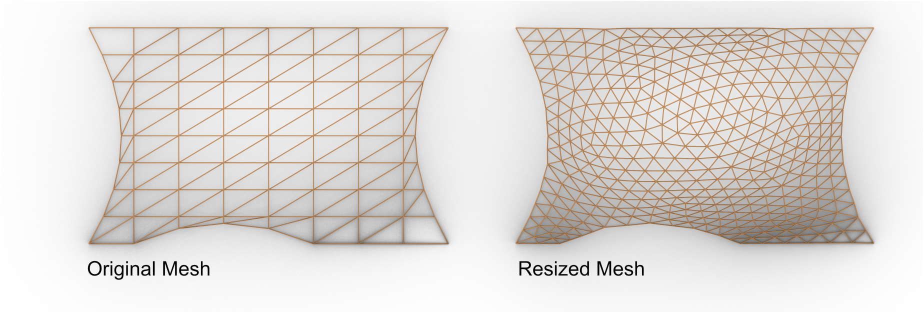


Figure 5.28: Top view of the mesh obtained from initial TNA consisting of 174 branches compared to the size-optimised mesh with 1003 branches.

### Analysing the new mesh

Before further optimisation of the gridshell a quick analysis is executed to verify the geometry after the re-meshing. Assuming that the magnitude of stresses in the arches remains the same after the re-mesh, new cross sections of 80 mm x 200 mm are subsequently selected for external arches. The internal elements initially get a cross section of 50 mm x 50 mm. The area load remains the same, but is this time divided into a greater number of nodes. The number of boundary edges also gets additional support nodes due to the increased amount of branches.

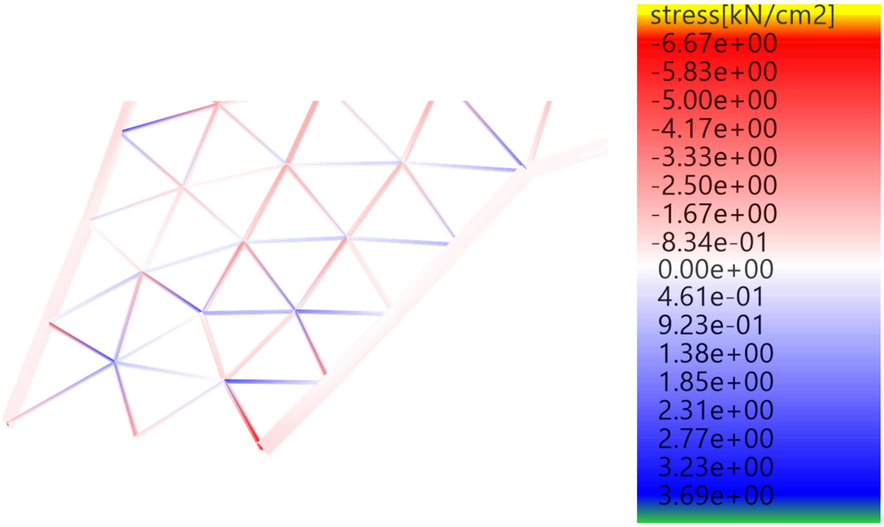


Figure 5.29: Detail of the gridshell where the stresses are highest. Blue colour indicates tensile stresses while red indicates compression. The location coincides well with the early phase indications from the reciprocal diagram in Figure 5.27 .

Table 5.5: Results from first order analysis of mesh in Figure 5.28 with *Karamba*. Utilisation ratios (UR) do not include reduction in capacity due to element buckling.

| | Rigid Joints | Pinned Joints |
|---|---|---|
| $N_{\text{Arch.max}}$ [kN] | −11.3 | −0.3 |
| $N_{\text{Arch.min}}$ [kN] | −178.5 | −143.7 |
| $N_{\text{Int.max}}$ [kN] | 32.66 | 84.9 |
| $N_{\text{Int.min}}$ [kN] | −69.9 | −164.4 |
| $M_{\text{Res.max}}$ [kNm] | 3.6 | 13.4 |
| $V_{\text{Res.max}}$ [kN] | 2.9 | 21.0 |
| $\sigma_{\text{x.Arch.min}}$ $[\frac{N}{mm^2}]$ | −16.2 | −70.0 |
| $\sigma_{\text{x.Arch.max}}$ $[\frac{N}{mm^2}]$ | 7.7 | 62.2 |
| $\sigma_{\text{x.Int.min}}$ $[\frac{N}{mm^2}]$ | −65.9 | −65.7 |
| $\sigma_{\text{x.Int.max}}$ $[\frac{N}{mm^2}]$ | 40.5 | 58.8 |
| $UR_{\text{Arch}}$ | 0.68 | 2.92 |
| $UR_{\text{Int}}$ | 2.74 | 2.73 |

Table 5.5 displays the results from a first order analysis similar to the one performed in section 5.3.3. With an increased number of structural elements to transfer the loads, utilisation of elements also decreases overall. The force patterns remain the same with the highest stresses in similar areas. Generally, the shape still remains funicular with moments primarily located at the edge arches and with a small magnitude compared to the axial forces. This is also the case when analysed with rigid internal nodes; thus substantiating the plausibility of the shape created with the TNA.

With the new cross sections, the arches now has a utilisation below 1.0 when submitted to similar stresses as in section 5.3.2. For the internal branches the utilisation is mainly below 1.0 as well, but some local concentrations of high stresses are present around the smallest arch. Here the forces have to travel around the opening in order to get to the supports; which increases the magnitude of forces in the area. Around the arch tensile elements are also present. This is emphasised in Figure 5.29. Going back to the reciprocal diagram for the initial shape in Figure 5.27, the polygons between the arches one the left size also have the largest areas; this exemplifies how the reciprocal diagram may be used to read the distribution of forces in an early stage design.

Overall, the subdivision of the mesh and the increase in cross sections have lowered the maximum utilisation of the structure; still being too high in some areas, the constructed gridshell is highly plausible and worth taking a closer look on with a final optimisation targeting the cross section and global stability.

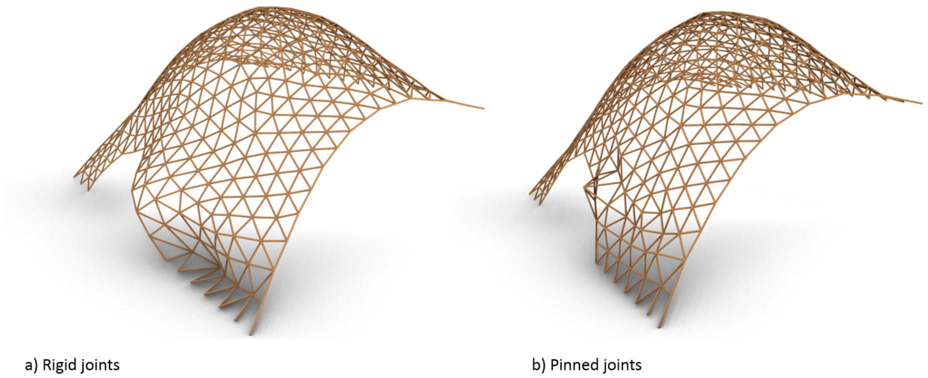


Figure 5.30: a) Global buckling with rigid joints. In Figure b) the joints are pinned, and the structure collapses under its own weight with local instability of members causing the collapse.

An observation regarding the global stability is mentioned before moving on to the final optimisation part. Unrealistic as it might be in timber gridshells, the structural analysis is first done with rigid joints to get an indication of the global behaviour of the structure. A second order analysis for this indicates global instability in the area around the front arch where the forces are highest. The corresponding critical buckling factor for the first buckling mode is 0.87. When using pinned joints, the critical coefficient plummets down to 0.0078, and the structure is no longer stable even for its self weight. Without the ability to transfer the load between elements through some moments in the nodes, the single layer gridshell becomes unstable. This is illustrated in Figure 5.30; Figure a) shows the global instability for the rigid joints, while Figure b) shows the element instability when using pinned joints.

As mentioned in section 1.4, the joint stiffness in timber gridshells is dependent on multiple factors such as element lengths, node geometry, number of fasteners etc. According to Larsson (2018), rotational values of a gridshell connection could range between 200-40.000 kNm/rad. With detailed analysis being beyond the scope of this example, a conservative rotational stiffness ($C_r$) of 100 kNm/rad is applied in $r_y$ and $r_z$ to demonstrate the effect on global stiffness. The resulting critical buckling factor then climbs to 0.20 after a new calculation. Even a small rotational stiffness in the joints greatly increases the stability of single layer gridshells by allowing the transport of moments through the nodes, and consequently making the system work as a uniform shell.

Summing up the results after modifying the mesh, the initial geometry still carries the load mainly by axial forces. Having obtained the new element lengths about 1 metre, the utilisation ratios are reduced to values more consistent with the selected cross section, with the exception of some areas. In addition, the importance of rotational stiffness is commented and handled in order to achieve a single layer gridshell, which, when based on the performed analysis, has a plausible geometry.

### 5.3.5 Structural optimisation

Using the new mesh obtained in the previous section, a final structural optimisation is performed in order to finally verify the plausibility of the obtained shape.

**Objective**

Before an optimisation is performed, a reasonable objective function needs to be formulated. From the analysis in section 5.3.4, the high concentrations of stresses near the smallest arch caused problems to both element utilisation and global stability. In other areas the stresses were low with little utilisation of the material. By using the elements' cross section area as parameters, the weight of the structure could be minimised by assigning cross sections which are proportional to the forces. This will result in a material effective structure which is also cheaper to produce.

*Galapagos* is used for this optimisation. The edge arches are set to a fixed cross section of 150 mm x 200 mm and has as a result a constant size. The input parameters will then be the cross sectional width/height of all the quadratic internal branches. These can be scaled individually by *Galapagos* in the range between 20-100 mm. By adjusting these, the fitness parameter, i.e. the total weight, is to be minimised. To ensure that the weight optimisation does not affect the structural stability of the gridshell, a penalty weight of 30,000 kg is added to the fitness function if the utilisation ratio of one of the branches is above 0.95.

The utilisation ratio is taken from the *Karamba* component checking the elements according to Eurocode 3 for steel. This accounts for capacity reduction due to local buckling. *Karamba* has yet to release an equivalent component for timber, but since it accounts for both lateral torsional buckling and element buckling it is considered sufficient for this purpose.

Nodal stiffness in the internal joints of 250 kN/rad is assumed in order to avoid the local instabilities illustrated in Figure 5.30 b) as discussed in section 5.3.4 while still being a conservative estimate.

With the fitness function and parameters defined, the *Galapagos* solver finds the structure weight and utilisation ratio needed by running a first order *Karamba* analysis for each iteration. As 50 unique solutions are required for each generation, this is a time-consuming process even for this single objective optimisation process; thus the advantage of already having an optimised shape when starting this process becomes clear.

Table 5.6: Summary of optimisation. The utilisation ratio in this Table is only taking into account the axial stresses and not the reduction factors in order for it to be comparable to the previous results.

| Total mass | $N_{min/max}$ | $M_{res.max}$ | $UR_{max/min}$ | $\sigma_{x.max/min}$ | $P_{cr}$ |
|---|---|---|---|---|---|
| 2387.3 kg | 45.6 kN / -161.6 kN | 2.58 kNm | 0.88 | 21.2 / -20.0 N/mm$^2$ | 1.98 |

### Results from Galapagos

Table 5.6 summarises the results from the weight optimisation. With a total weight of 2387.3 kg, the gridshell now has a utilisation ratio of only 0.88 compared to before the optimisation when it was 2.92, and this means that the structure now has efficient capacity to transport the loads to the ground. Furthermore, the global stability has increased with a critical buckling factor of 1.98.

Figure 5.31 presents the cross sections after the optimisation. The thickest elements are located where the forces from the previous analysis in section 5.3.4 were largest; i.e. the front arch, while elements in areas with lower forces have become smaller after the optimisation.

In addition, the elements where tensile stresses are present are marked in red. Apart from around the front arch these are in the order of 0-3 N/mm$^2$, and thus not of any particular interest. The tensile forces around the front arch, on the other hand, has a maximum value of 20 N/mm$^2$. This is close to the capacity of the material and does not correlate with the hypothesis of the structure being in a state of pure compression.

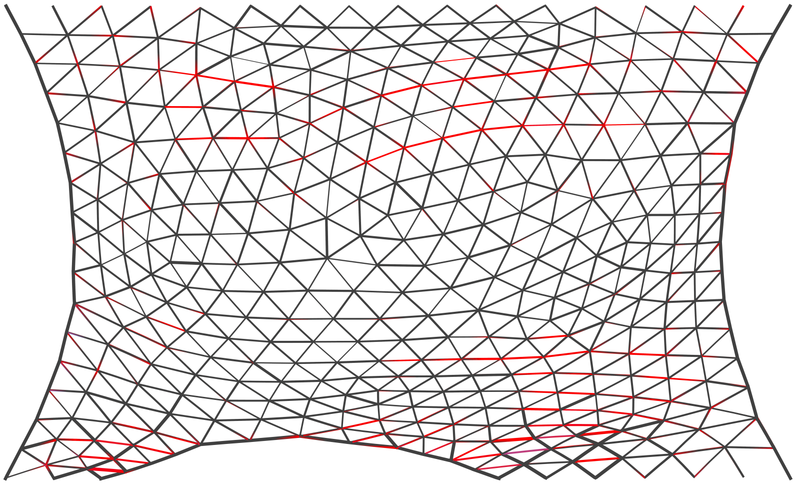


Figure 5.31: Top view of the results from the cross section optimisation. The elements in red are in tension, while the black ones are subjected to compression forces. The width of the cross sections corresponds to the line thickness of each element.

By closer examination, the main cause for this seems to be the compression forces pushing the front arch down. The horizontal-oriented elements restrain this motion and thus experience axial tension. This also causes some torsional moments in the same region, which again causes additional stresses. With timber having greater capacity in tension than compression, this is not regarded as a problem.

**Resistance to horizontal loading and eccentricities**

As a final verification, the structure's stability when subjected to horizontal load and eccentricities is tested. Section 5.2.3 revealed how the single layer gridshell's stability decreased when subjected to horizontal loading. Furthermore, section 1.4 describes how the stability should be evaluated by applying eccentricities based on the scaled eigenvalues, Bulenda & Knippers (2001).

The scaled eigenvalues is first used to test the strucure's stability. When moving the nodes according to the first eigenvalue mode, the structure is forced into its most unstable position. This makes for an efficient method to evaluate how prone the structure is to eccentricities which in a real structure is inevitable. Assuming the maximum allowed deflection in the nodes to be L/100, a sufficient deviation from the ideal condition is expected. With element lengths of about 1 metre, this results in a maximum nodal deviation of 10 mm.

When running a second order analysis with *Karamba*, the critical buckling factor is 1.98 while the maximum displacement is 41.7 mm. The *Nodal displacement* component is then used to get the displacement vectors for each node corresponding to the first eigenmode. By normalising these according to the maximum nodal deviation of 10 mm the transformation vectors are created. The nodal positions in the original gridshell are then moved according to these vectors before a new analysis is performed.

In addition, eccentricities caused by random translations in each node with a maximum length of 10 mm are applied as well. To do this, a new optimisation problem is carried out; this time using *Silvereye*. By creating vectors for each free node where the x-, y-, and z-displacements are used as parameters for the optimisation, the *Silvereye* optimiser seeks to minimise the critical buckling load. The results are presented in Table 5.7.

Both of the methods used to apply eccentricities decreased the stability of the structure. The critical buckling load from the scaled eigenvalues decreased by 2.5 % compared to 4.5 % for the random eccentricities. Even though the minimisation of the critical buckling factor yielded a more conservative result, the difference does not necessarily justify the extra computational time. The scaled eigenvalue method only takes a couple of minutes to set up, while the optimisation problem in this case took 45 minutes. Furthermore, the eigenvalue scaling has a maximum eccentricity in one node of 10 mm, whereas all the other will have a smaller value. The approach with random eccentricities could, in theory, yield a 10 mm displacement in every node, which is extremely conservative given the already exaggerated allowed displacement. Regardless of this, the structure seems to handle eccentricities in the nodes under vertical loading.

Table 5.7: Stability of the gridshell from a second order analysis. LC0 is the original load case where only vertical loads are present. LC1 is the load case with horizontal loads. Note: In the last row only 5 mm eccentricities are applied compared to 10 mm for the rows.

| | $w_{Max}$ [mm] | $P_{cr}$ | Decrease $P_{cr}$ [%] |
|---|---|---|---|
| LC0 (250 kNm/rad) - No eccentricities | 41.7 | 1.98 | 0.0 |
| LC0 (250 kNm/rad) - Scaled eigenvalues | 46.8 | 1.93 | 2.5 |
| LC0 (250 kNm/rad) - Random eccentricities | 45.6 | 1.89 | 4.5 |
| LC1 (250 kNm/rad) - No eccentricities | N.A | N.A | N.A |
| LC1 (rigid joint) - No eccentricities | 117.6 | 1.46 | 26.3 |
| LC1 (2500 kNm/rad) - No eccentricities | 127.6 | 1.35 | 31.8 |
| LC1 (2500 kNm/rad) - Scaled eigenvalues | 145.6 | 1.27 | 35.6 |

The global stability when the gridshell is subjected to a horizontal load is also tested. The magnitude of this load is assumed to be 1.5 kN/m$^2$ evenly distributed into every free node. This is a coarse estimate, but regarded as sufficient to assess the effect it has on the stability in a conceptual phase. The vertical loading is now only the dead load from the cladding, 1 kN/m$^2$, and the structure's self weight.

As for section 5.2.3, this single layer gridshell also seems to be highly affected by horizontal loads. When using semi-rigid joints with a rotational stiffness of 250 kNm/rad the second order analysis never converges, and no stable solution is found. Consequently, the system is recalculated with rigid joints, resulting in a critical buckling factor of 1.46. This is a reduction of 26.3 % compared to the initial buckling factor. The importance of a certain amount of nodal stiffness when considering the effect of horizontal loads on the global stability is clearly shown here.

To summarise the findings of this section a final configuration is tested. A more realistic rotational stiffness in the joints of 2500 kNm/rad is used, and the stability is again tested with and without applying eccentricities from the scaled eigenvalues. Compared to the initial system, the critical buckling load is now reduced by 31.8 % to 1.35 without applying eccentricities. This is substantially lower, but the structure is still considered stable. Even more interesting is the effect of applying the scaled eigenvalues to this system.

For the maximum allowed nodal translation of 10 mm the solution never converges, and a stable system is, as a result, not found. By reducing the allowed eccentricities to 5 mm, the system finds a stable solution with a $P_{cr}$ of 1.27. The effect of eccentricities on the horizontal load situations, therefore, seems to have a bigger effect on the stability than for the vertical loading situations.

### 5.3.6 Summary

The purpose of this case study was to investigate how the form finding components developed in this thesis could be used when developing a conceptual gridshell. Aiming for a gridshell primarily carrying loads through membrane forces, the initial form was generated by the components presented in this thesis. Although the obtained form was mainly subjected to axial forces, the limited number of branches yielded gridshell elements of unrealistic sizes. Further optimisation was subsequently performed based on the original shape, before the plausibility of the gridshell was verified by the use of an FEA.

A brief summary of the case study is listed below:

- The *TNA3D Solver* creates global shapes primarily carrying the loads through membrane forces.
- Due to computational time the number of available branches is kept below 200 in order to discover a shape with real time feedback. This causes the results from the *Capacity Check* and *Buckling 2D beam* components to have little value as the branches have unrealistic lengths.
- The workflow in this section, where the global shape is first found with a fast tool such as the TNA, before being refined with more time consuming methods, proves to be an efficient way of developing a plausible gridshell, which, in the end, was the main objective of this case study.
- When evaluating the global stiffness of a single layer gridshell, the effect of the joints' rotational stiffness was investigated. While vertical loading situations only required a small rotational stiffness in order to work as a uniform shell; the stability of the gridshell when subjected to horizontal loads was highly dependent on the joints' stiffness.

## 5.4 Case study IV: Compost hall

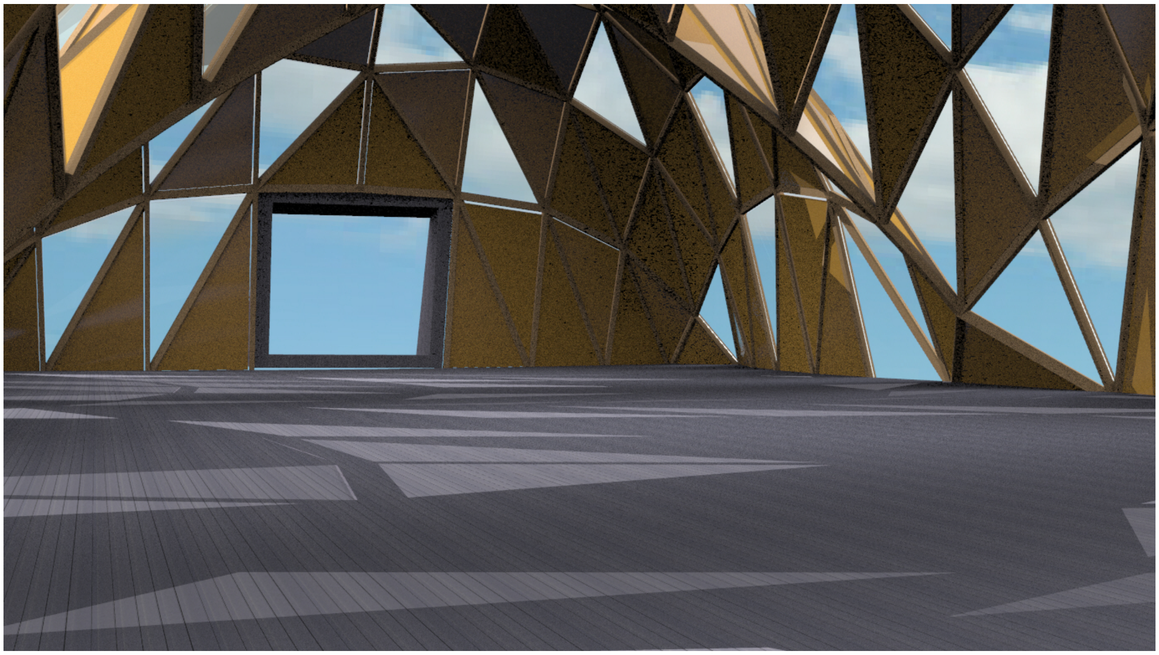


Figure 5.32: Illustration of the compost hall.

On the western coast of Norway, atn the island of Stord, an industrial hall for compost storage has been proposed built; the original design has a roof consisting of a series of steel trusses. In this case study, two alternative design propositions presented by PhD candidate Steinar Hillersøy Dyvik is recreated and evaluated by the TNA components. These are then compared to the original design in order to evaluate if they are feasible alternatives to the original proposition.

Some design specifications have to be specified to ensure that the original functionality of the building is maintained. The design propositions must have the same ground area as the original hall. The steel structure has a maximum height of 10 metres, but for the purpose of enabling funicular structures to be created by the TNA, the allowed height is increased to 13 metres. Additional specifications are listed below:

- Maximum height: 13 m.
- Base area: 30 m x 60 m.
- Material: Timber GL30c, the properties are shown in Table 5.8.
- Three openings with a size of 5 m x 6 m.
- Self weight from the branches (load factor 1.2).
- Roof load of 1 kN/m$^2$ lumped onto nodes (load factor 1.2).
- Snow load of 2 kN/m$^2$ lumped onto nodes (load factor 1.5).

- Vertical wind load of 1 kN/m$^2$ lumped onto nodes (load factor 1.05).
- Horizontal wind load of 2 kN/m$^2$ lumped onto half of the nodes for a given direction (load factor 1.05).

Table 5.8: Properties of GL30c is obtained from NS-EN 14080, Timber Structures (2016), and reduced according to NS-EN 1995-1-1, Eurocode 5 (2010).

| Properties | Value |
|---|---|
| Compression strength [N/mm$^2$] | 24.5 |
| E-modulus [N/mm$^2$] | 13 000 |
| Shear-modulus [N/mm$^2$] | 650 |
| Density [kg/m$^3$] | 430 |

In the previous sections, the nodal connections of the gridshell elements have been discussed. With rigid joints being difficult to achieve, this study uses semi-rigid joints similar to the one in Figure 5.33. The values for rotational- and translational stiffness are given in Table 5.9 and are based on the research by Larsson (2018).

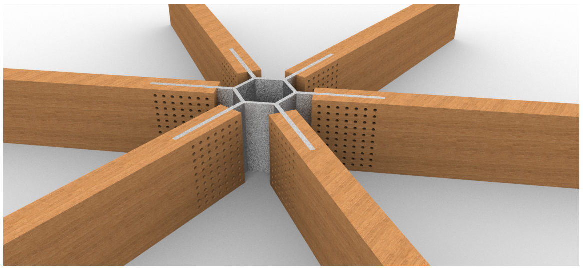


Figure 5.33: The nodal connection used in this case study.

Table 5.9: Stiffness values for steel joint used in Case study IV.

| $C_{t.x}$ [kN/m] | $C_{t.y}$ [kN/m] | $C_{t.z}$ [kN/m] | $C_{r.x}$ [kNm/rad] | $C_{r.y}$ [kNm/rad] | $C_{r.z}$ [kNm/rad] |
|---|---|---|---|---|---|
| 237431 | 722329 | 237431 | 20688 | 11090 | 12657 |

### 5.4.1 Design Concept 1

#### Creating the roof

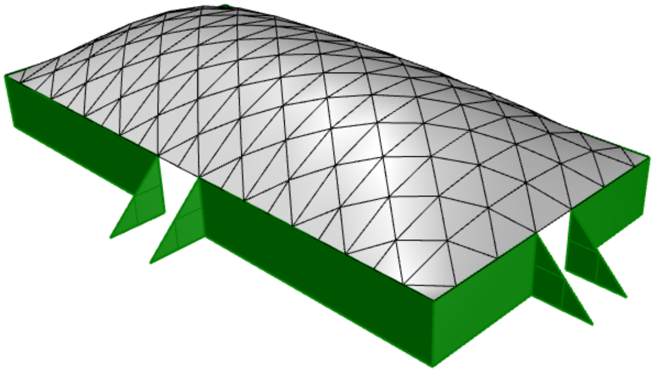


Figure 5.34: Design Concept 1: Illustration from the architect. Concrete walls and a timber gridshell roof.

In the first proposition from the architect four concrete walls supports a rectangular timber gridshell roof, Figure 5.34. This is created directly with the *Surface To Grid* component. Based on the stability analysis from section 5.2, mesh 5 is selected with a u- x v-division of 20 x 20, resulting in 595 branches and 238 nodes. Given the large span of the roof, an initial cross section of 150 mm x 250 mm is selected.

Having established the geometry, the *TNA 3D Solver* is then used to find the initial form. With the concrete walls having a height of 6.5 metres, the maximum allowed height for the roof is 6.5 metres as well. The solution is scaled to this height based on the results from section 5.1.2. The obtained shape is pictured in Figure 5.35 along with its primal and dual grid.

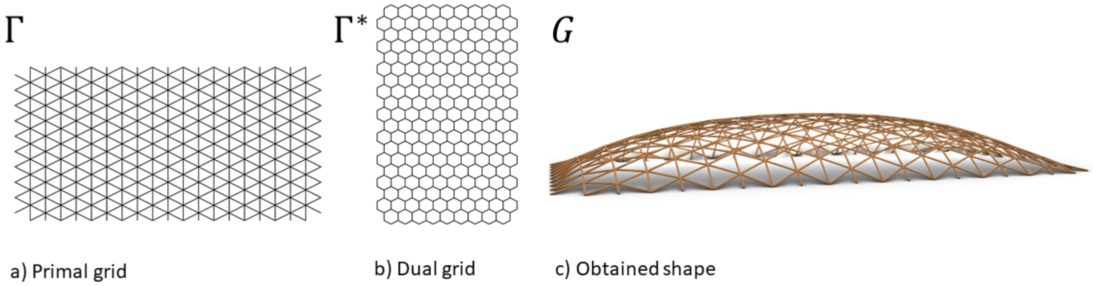


Figure 5.35: Proposed shape from the *TNA3D Solver* along with the corresponding primal and dual grid.

Table 5.10: Results from the TNA.

| $N_{Max}$ [kN] | $N_{Min}$[kN] | UR [%] | $P_{cr}$ | $N_{Max.Estimate}$ [kN] | $P_{cr.Estimate}$ |
|---|---|---|---|---|---|
| -188.5 | -142.2 | 28 | 2.0 | (-235.6 to -269.3) | 6.7 |

**Forces and stability from the TNA**

Table 5.10 shows the results generated from the TNA analysis. A material utilisation of 28 % is acceptable at this stage knowing that other loading situations from the horizontal loads could create a more unfavourable situation when verified in *Karamba*. Case study I and II showed that a low height to span ratio will influence the accuracy; as a consequence, an adjustment to the values are attempted based on the results from these studies.

Starting with the axial forces in Figure 5.9 and 5.10; the height to span ratio of this roof becomes $r_h = 0.4$ which places it in the area where the accuracy of the TNA decreases. Both for pinned and rigid connections mesh 5 produced a solution approximately 10 % lower than the FEA for the quadratic dome. With a rectangular base area, where the length is two times larger than the width, the FEA results are expected to behave more like a barrel vault than the TNA. Based on this, the results in Table 5.10 are expected to produce results 20-30 % lower than the FEA.

Then an estimation of the global stability is attempted. Looking at Figure 5.23, a value $r_h = 0.43$ indicates that the 2D estimate is about 60 % lower than the result from the TNA. When applying this to the estimate, the critical buckling factor from *Karamba* should be around 6.7. If both of these estimates are precise, the obtained shape is highly plausible and could be a good alternative to the original steel hall.

**Verification with FEA**

To verify the results and estimates from the TNA, *Karamba* is used to analyse the obtained roof geometry. All edges are assumed simply supported and the nodal connections are modelled according to Table 5.9. All loads apart from the self weight of the grid are lumped onto the nodes. Two calculations, both with and without the horizontal wind load, are subsequently performed. The wind load is applied either to the long side or the short side of the roof. The worst case of these two are presented for each column in Table 5.12 together with the results from the vertical loads.

An FEA was presented in the last section based on the TNA results from the previous case studies. For the maximum axial forces, these seem to apply when including the change in base geometry. The maximum compression force from the FEA with rigid joints is -252.0 kN which falls in the range of the predicted estimate. Apart from just 4 corner elements, the entire structure is in compression with the forces being transferred over the shortest span like a barrel vault as predicted. The utilisation ratios are also acceptable and below 50 % for all load cases.

Table 5.11: Results from the FEA. Rigid joints are added for both load cases as a reference.

| | Vertical loads | | w/Horizontal wind | |
|---|---|---|---|---|
| | Semi-rigid | Rigid | Semi-rigid | Rigid |
| $N_{Max}$ [kN] | 32.3 | 25.9 | 49.1 | 41.8 |
| $N_{Min}$ [kN] | -273.0 | -252.0 | -284.7 | -277.6 |
| $V_{Res.max}$ [kN] | 2.2 | 2.2 | 2.4 | 2.7 |
| $M_{Res.max}$ [kNm] | 5.5 | 4.5 | 6.4 | 5.4 |
| $\sigma_x$ [N/mm$^2$] | -9.5 | -8.2 | -10.1 | -9.1 |
| UR [%] | 40.7 | 35 | 43.2 | 38.7 |
| $P_{cr}$ | 2.4 | 3.4 | 2.3 | 3.2 |
| $w$ [mm] | 51 | 23 | 53 | 24 |

The estimated critical buckling factor for the vertical loading with rigid connections was 6.7. A second order analysis from *Karamba* produced a value of 3.4. Thus the estimate was close to twice as high compared to *Karamba*, while the actual critical buckling factor from the TNA was 2.0, which is 40 % lower than the FEA result. This could be caused by the differences in the base geometry, loading conditions, element lengths, or cross sectional areas and is not investigated further in this case.

**Concept summary**

Overall, the first proposition from the architect seems highly plausible and easy to produce. The simple shape was uncomplicatedly obtained with the TNA, and the estimated results corresponded well with the ones from *Karamba* when considering the axial forces and utilisation. Working almost funicular when subjected to vertical loading, the structure also has sufficient capacity when exposed to different load situations. The global stability analysis was inaccurate and, consequently, needs further research in order to become useful for arbitrary geometries.

With a total weight of 32.0 tons and a maximum utilisation below 50 %, the weight can be further reduced by optimising the design at another time. A final comparison with the other design propositions is performed in section 5.4.6.

### 5.4.2 Design Concept 2

In the second design concept the entire hall is built as a timber gridshell; the architect proposed a hall with three dome-shaped sections, Figure 5.36. The highest point of the structure is originally 13.6 metres, but the proposition presented here is limited to 13 metres in order to comply to the design specifications. The internal arches have a maximum height of 9 metres.

The objective is to use the *TNA3D Solver* to create a shape as close as possible to the architect's vision. First, an initial shape is established and analysed before the *Internal Arch* component is used to create internal arches; the openings will not be designed here. Specifications, material properties, and load cases are the same as used for Concept 1 and are given in the introduction of section 5.4.

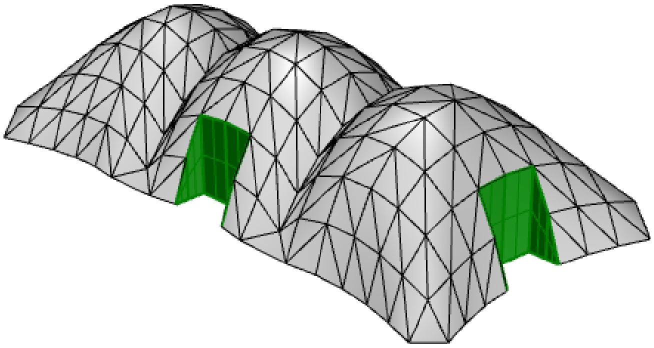


Figure 5.36: Design concept 2: Illustration from the architect. The timber gridshell with port-openings.

### 5.4.3 Creating the initial shape

Before the initial shape is created the primal grid has to be established. The global geometry from the architect is projected down to the xy-plane as illustrated in Figure 5.37 a). The "cuts" in the mesh where the gate openings are located are then drawn manually to create an even mesh (Figure 5.37 b)). The final primal grid has a total of 614 branches and 171 nodes which is used by the *TNA3D Solver* to create the initial geometry in Figure 5.37 c).

With the initial geometry established, the *Capacity Control* is used to verify the magnitude of forces. The largest forces in the structure have a magnitude of 357 kN, giving an utilisation of 29 % for the highest utilised elements. None of the grid elements are prone to local buckling, and the critical buckling factor for the system is 5.4. The corresponding buckling shape is illustrated in Figure 5.38.

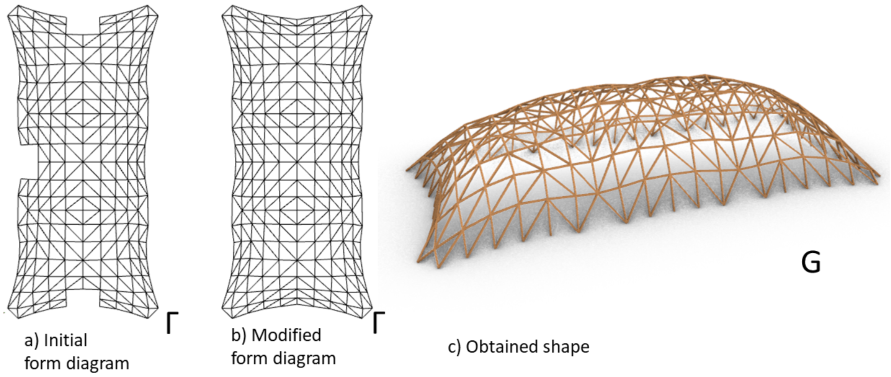


Figure 5.37: Form finding process of the initial shape.

As for the previous design proposition, an estimation of the FEA results is presented based on the studies in previous cases. This particular mesh configuration has not been tested earlier in the thesis, which makes it interesting when evaluating the generality of the estimates.

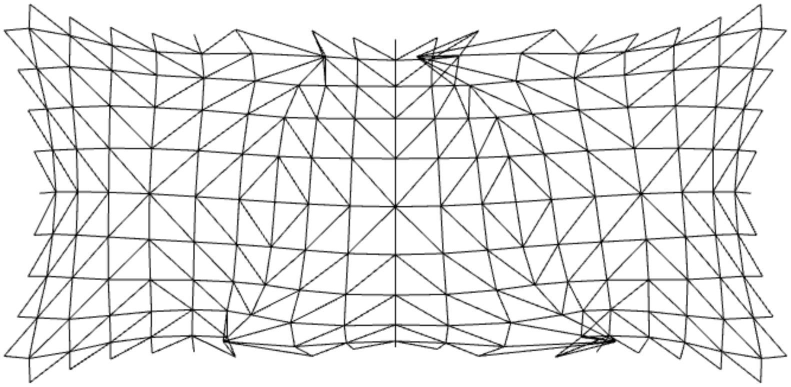


Figure 5.38: The buckling shape obtained with *Buckling 2D Beam* component.

Given that $r_h = 0.85$, Figures 5.9 and 5.10 indicate that triangular meshes overestimate the solution by a factor of 1.1-1.2. Assuming that the rectangular base area makes the structure behave more like a barrel vault, similarly to the first concept, this would result in the estimate being approximately 10 % lower than presented in the figures.

Table 5.12: Results from the FEA. Rigid joints are added for both load cases as a reference to the TNA estimates. The results in the last four columns are obtained with *Karamba*. The two critical buckling factors are from the *Buckling 2D Beam* component, with the estimated one being inside the parentheses. Maximum utilisation ratios use axial forces from both bending moments and normal forces.

| | | Vertical loads | | w/Horizontal wind | |
|---|---|---|---|---|---|
| | TNA | Semi-rigid | Rigid | Semi-rigid | Rigid |
| $N_{Max}$ [kN] | -40.0 | 20.1 | 12.7 | 72.2 | - |
| $N_{Min}$ [kN] | -354.7 | -273.5 | -262.5 | -382.8 | -368.6 |
| $V_{Res.max}$ [kN] | - | 4.9 | 4.4 | 6.6 | 5.1 |
| $M_{Res.max}$ [kNm] | - | 7.3 | 6.5 | 11.5 | 8.8 |
| $\sigma_x$ [N/mm$^2$] | -7.1 | -7.5 | -6.8 | -11.4 | -10.3 |
| UR [%] | 29.0 | 31.9 | 29.0 | 48.5 | 44.2 |
| $P_{cr}$ | 5.4 (*9.8*) | 3.7 | 5.1 | 3.6 | 5.0 |
| $w$ [mm] | - | 28.8 | 14.1 | 35.8 | 15.9 |

Equation (5.3) from section 5.2.4 is used to estimate the critical buckling factor of the structure. With a triangular grid the estimate for a vertical loading situation becomes:

$$P_{cr.Estimate} = \frac{Pcr_{2D}}{r_{Pcr}} = \frac{5.4}{0.55} = 9.8 \tag{5.4}$$

Results from the initial TNA are given in Table 5.12 together with the FEA results.

### Karamba verification of initial design

When comparing the TNA to the FEA from *Karamba*, the results with rigid joints are used as reference. The semi-rigid joints are used for the structural verification.

Starting with the axial forces, the TNA result is 35 % higher than the FEA forces. This is higher than both the indications from Case I as well as the assumption of the load transfer made in the previous section; this makes the TNA result more conservative than expected. The reason for this may be how the mesh configuration transfers forces in an unrealistic and inefficient way compared to the FEA. Apart from this discrepancy in forces, the presence of tension elements is moderate and the bending moments are small.

The results for the global stability differs significantly compared to the previous sections. The 2D simplification gives a critical buckling factor 5.9 % higher than the one from the FEA. When modifying this according to equation 5.3 the value becomes unrealistically high. In the previous cases the 2D result was always conservative, which is clearly not the case here. The meshes, therefore, seem to largely influence the accuracy of the estimates. This should be further investigated before the component's results should be trusted.

Using the semi-rigid joints, the structure has a sufficient capacity with a maximum utilisation of 48.5 % when subjected to horizontal loads as well. The global stability, like for Concept 1, seems to be almost unaffected by the wind load and has a satisfactory critical buckling load. Due to the overall low utilisation the cross section of the elements is reduced to 150x200 mm before the shape is adjusted.

### 5.4.4 Adjusting the shape

To achieve the shape proposed by the architect, the force diagram is modified as described in section 4.5. By scaling the branches in the dual grid where the internal arches are to be located, the horizontal forces in these branches increase. The dual grid before and after this modification is pictured in Figure 5.40.

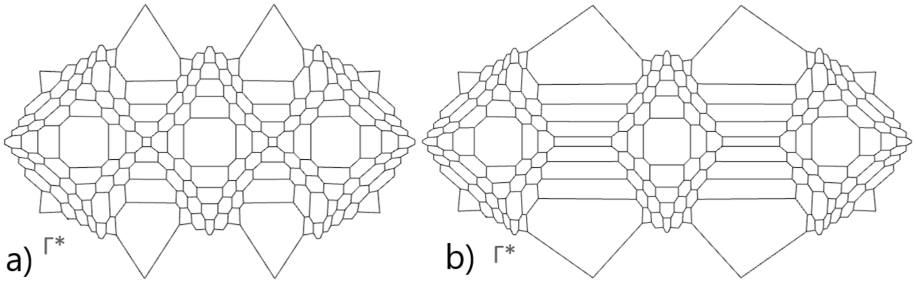


Figure 5.39: Altering of force diagram. a) Initial force diagram. b) Modified force diagram.

After changing the forces in the dual grid the shape shown in Figure 5.40 is obtained; the altering of the forces creates two internal arches. The new shape is 12.7 metres at the highest point, and the internal arches are 8.6 metres high. While still fulfilling all requirements, the new shape now resembles the proposed one. The main difference is that all domes have the same height, whereas the middle one was tallest in the sketches. The branches have lengths varying from 2.7 m to 6.8 m.

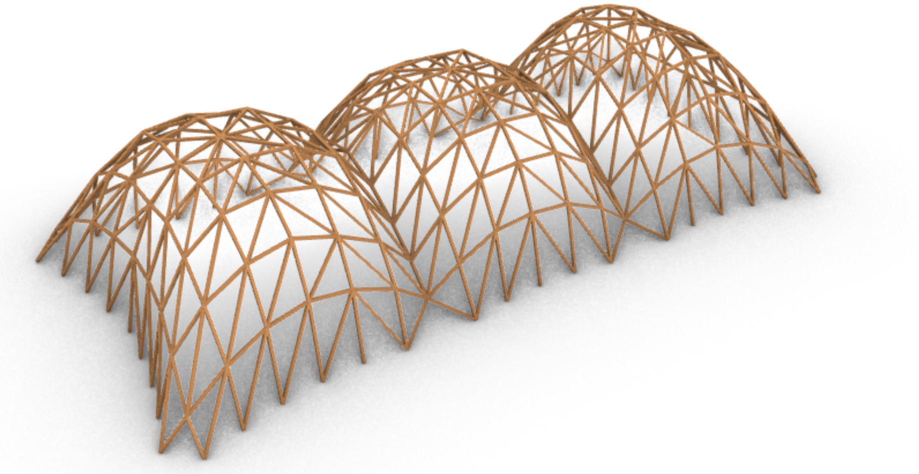


Figure 5.40: The final shape obtained after altering the force diagram.

As a consequence of the modified force diagram, the distribution of forces in the gridshell needs to be re-evaluated. As the new dual grid illustrates, the internal arches have increased maximum compression forces. These acts almost as an internal support for the other branches, consequently attracting larger horizontal forces. Table 5.13 shows the results from a new TNA analysis together with the FEA from *Karamba*.

As for the initial design, this one is also evaluated using the *Capacity Control* and *Buckling 2D Beam*. Figure 5.41 pictures the utilisation of the structure, which apart from at the internal arches' end, has sufficient capacity. Aside from the branches in red and yellow where the utilisation is between 75-115 %, the utilisation is approximately 13%.

The results from the *Buckling 2D Beam* shows how the structure buckles around the end of the arches where the forces are largest, Figure 5.42. The following critical buckling factor is now 1.1. Trying to estimate the global factor again from equation (5.3), without reduction from horizontal loads, gives:

$$Pcr_{estimate} = \frac{Pcr_{2D}}{r_{Pcr}} = \frac{1.1}{0.55} = 2.0 \tag{5.5}$$

Results from the TNA verification are presented in Figure 5.13.

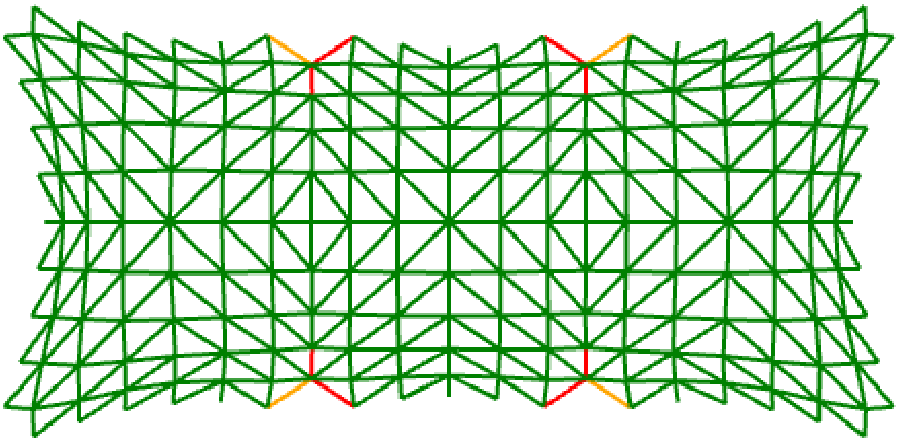


Figure 5.41: The utilisation of elements calculated and visualised with the *Capacity Control*. Red elements indicate too high utilisation.

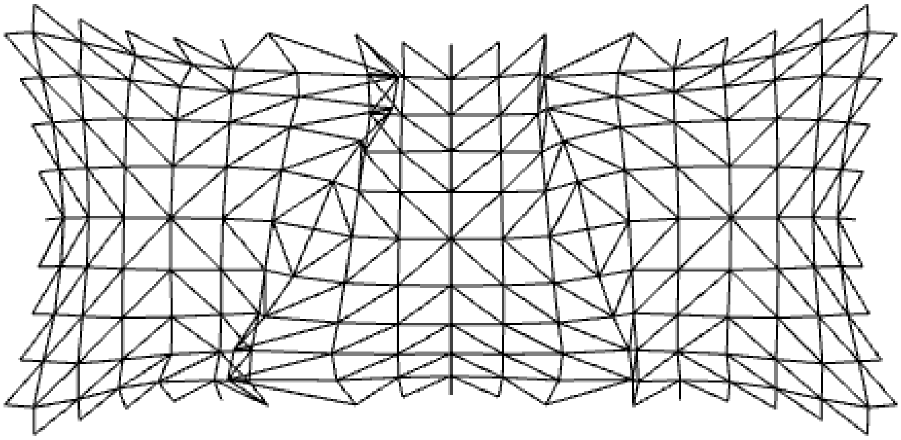


Figure 5.42: The buckling shape obtained with the *Buckling 2D Beam* component.

### 5.4.5 Verification with Karamba

As stated, the structure is analysed in *Karamba* both with and without wind load. The most unfavourable result for each loading situation is shown together with the ones from the TNA in Table 5.13.

Starting with the global stability, the estimated critical buckling load is now 50 % lower than the one obtained with semi-rigid joints (which, again, is smaller than for rigid joints), showing once again that further research must be undertaken before the tool becomes useful.

When comparing the axial forces from the TNA and the FEA, the difference in loading distribution causes a large deviance. Figure 5.43 illustrates the normal force diagram of the building. Here, tensile forces above 100 kN are present in the branches spanning in the longest direction towards the internal arches. The system is modelled with semi-rigid joints also used in the internal arches. While the TNA uses these arches to absorb normal forces as a "support" for the other branches, the FEA transfers forces more like a barrel vault. All branches spanning over the shortest side are in compression with FEA, and the tension forces are present around the internal arches.

Although the structure is stable, it is irrational to build a vault with several kinks without supports beneath them; therefore, an additional FEA analysis is undertaken. This time a modification similar to the one in section 5.3 is made to the model; the two internal branches get a cross section of 200x250 mm and rigid connections between them in order for them to behave like actual supportive arches.

Table 5.13: Results from the FEA. Rigid joints are added for both load cases as a reference to the TNA estimates. The results in the last four columns are obtained with *Karamba*. The two critical buckling factors are from the *Buckling 2D Beam*component, with the estimated one being inside the parentheses. Maximum utilisation ratios use axial forces from both bending moments and normal forces.

| | TNA | FEA No Wind | FEA Wind |
|---|---|---|---|
| $N_{\text{Max.arch}}$ [kN] | -20.6 | -105.6 | -88.8 |
| $N_{\text{Min.arch}}$ [kN] | -846.7 | -301.0 | -418.3 |
| $N_{\text{Max.internal}}$ [kN] | -20.6 | 125.0 | 147.2 |
| $N_{\text{Min.internal}}$ [kN] | -94.7 | -276.0 | -315.2 |
| $UR_{\text{Max arch}}$ [%] | 115 | 39.2 | 1.48 |
| $U_{\text{R.max internal}}$ [%] | 13 | 53.4 | 58.4 |
| $M_{\text{Max}}$ [kNm] | - | 5.7 | 7.7 |
| $V_{\text{Max}}$ [kN] | - | 2.5 | 3.3 |
| $w_{\text{max}}$ [mm] | - | 18.1 | 26.4 |
| $P_{\text{cr}}$ | 1.1(*2.0*) | 5.1 | 4.0 |

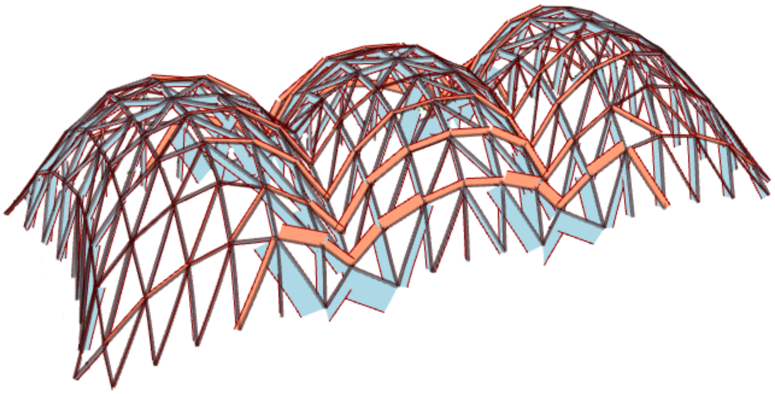


Figure 5.43: Axial forces caused by vertical loading. Tensile forces are in red and compression forces are in blue.

### Modified Karamba analysis

Table 5.14 shows the results after the modification of the internal arches. The internal arches now attract more forces, but still not close to the magnitude from the TNA. The distribution resembles the one from Figure 5.43, but with a smaller presence of tensile forces. The utilisation remains approximately the same; however, the global stability has now increased as a consequence of both an increased cross section in the internal arches and greater stiffness.

Even though the TNA results in the internal branches are greatly overestimated, the resulting shape still transfers the load mostly through axial forces. When subjected to horizontal wind load, the structure still has sufficient capacity and stability.

Table 5.14: Results from the FEA. All vertical loads are applied. *Karamba* results are shown with and without wind load. Cross section 150 mm x200 mm. GL30c. Height 12.8 m. The negative sign indicates compression. Two values for critical buckling factor are shown in the table; one from *Buckling Beam 2D* and the estimated one in parentheses. The maximum utilisation ratio uses axial forces both from bending moments and normal forces.

| | TNA | FEA No Wind | FEA Wind |
|---|---|---|---|
| $N_{Max.arch}$ [kN] | -20.6 | -243.3 | -332.6 |
| $N_{Min.arch}$ [kN] | -846.7 | -353.4 | -480.6 |
| $N_{Max.internal}$ [kN] | -20.6 | 107.2 | 138,7 |
| $N_{Min.internal}$ [kN] | -94.7 | -246.9 | -295.7 |
| $UR_{Max.arch}$ [%] | 115 | 44.2 | 60.4 |
| $UR_{Max.internal}$ [%] | 13 | 47.2 | 56.2 |
| $M_{Max}$ [kNm] | - | 7.7 | 8.7 |
| $V_{Max}$ [kN] | - | 3.5 | 4.1 |
| $w_{Max}$ [mm] | - | 17.2 | 25.3 |
| $P_{cr}$ | 1.1(*2.0*) | 5.7 | 4.6 |

**Concept summary**

Overall, the second proposition has a satisfying structural performance with two internal arches giving the structure an increased stability compared to the first proposition. For this specific mesh the TNA results were inaccurate. For the initial shape, the critical buckling factor was too high and non conservative; after the modifications of the dual grid, the buckling estimate was lower than the FEA results. This deviance rendered it useless before further research on the effect of different shapes, loading conditions etc. has been performed. For the final design the forces in the internal arch were vastly greater than the TNA forces; as a result, the *Capacity Control* and *Buckling 2D Beam components* yield invalid results for structures with internal arches.

Apart from this, the final shape is highly plausible to construct and will undergo a final comparison with two the others in the next section.

Table 5.15: Price estimates for the different designs. Note: Assembly costs are not included in design 1 and design 2.

| | Original | Design 1 | Design 2 |
|---|---|---|---|
| Joints [kr] | - | 1.071.000 | 770.000 |
| Timber [kr] | - | 595.000 | 590.000 |
| Concrete [kr] | - | 300.000 | 32.000 |
| Total | 4.000.000 | 1.966.000 | 1.392.000 |

### 5.4.6 Comparison of designs

The original design has a roof supported by 11 steel trusses with a total weight of 31.9 tons. The price of the planned steel building is 4 MNOK, 2200 NOK/m$^2$, including the assembly, but without the groundwork.

Design Concept 1 has a total of 238 steel joints and 32.0 tons of timber in the roof structure. In addition, concrete walls have to be cast. Assuming a thickness of 200 mm, the required volume then becomes 150 m$^3$. The final price is given in Table 5.15, but can probably be reduced if cross section is optimised.

Design Concept 2 is a pure gridshell without the need for supportive concrete walls. The internal branches have a cross section of 150 mm x 200 mm, while the arches have a size of 200 mm x 250 mm. The total weight of timber is 31.8 tons. 171 steel joints are needed to assemble the gridshell. In addition, three gate openings with 200 mm thick walls have a volume of 16 m$^3$.

Both of the concepts investigated in this case have the possibility to be optimised in order to achieve a more effective and economic structure. Regarding the assembly process, the first alternative is likely to be the easiest to construct. With all the beam elements having the same cross section and more even lengths than the second option, this is most likely the cheapest to construct. The second concept has the advantage of needing less concrete, making it a more viable, environmental-friendly choice..

Table 5.15 uses the following assumptions when estimating the price: Concrete has a cost of approximately 2000 NOK/m$^3$, glue laminated timber elements in Norway have a price of around 18,6 NOK/kg for beams up to 250 mm wide, and each steel joint has an estimated price of 4500 NOK.

Based on the results in this section, Concept 2 is recommended as an alternative to the steel hall. Both the timber gridshells have a material price below 2 MNOK, and should thus be price competitive with the original hall. With greater stability, a more interesting design and less use of concrete, the second concept is preferred to the first one.

### 5.4.7 Summary

This case study investigated if a timber gridshell could be a feasible alternative to a planned steel truss hall on Stord. Two design concepts were developed using the *TNA 3D Solver* and validated with *Karamba*. The main findings were as follows:

- Both of the proposed designs could be created using the developed components in the *Camel* package, thus proving how versatile the components are.
- Both of the design propositions are good alternatives to the steel truss hall. The prices are competitive, the total weights are competitive, and the designs are more visually appealing.
- Concept 1 is plausible, and it is an easy structure to produce. The axial force and utilisation from the TNA corresponded well with the *Karamba* analysis. The global stability estimate from the *Buckling 2D Beam* was inaccurate.
- The results from Concept 2 showed that for the internal arches in the structure, the TNA and *Karamba* had different distribution of forces. This causes the forces from the TNA to be vastly larger in the arches than the FEA forces; therefore, the capacity components produced inadequate results. Apart from this, the obtained shape transferred the forces mostly through axial forces, resulting in a material effective structure.
- When considering price, material use, and stability both design concepts were found to be feasible alternatives to the original steel hall.
- For the *Buckling 2D Beam* component, the numerical approximation from the quadratic dome research does not seem to apply in this case. When the base surface was rectangular and the mesh irregular, the results became invalid; therefore, a larger variety of reference shapes should be tested to further improve the component.
- The estimated buckling shapes were useful as they gave indications of where the weak points of the structure were in terms of the global stability.

## 5.5 Case study V: Flexibility

In this final case study, the versatility of the *TNA3D Solver* is presented. Eight different shapes that differ from are obtained with the developed solver. No structural analysis and verification of the shapes are performed.

### 5.5.1 British Museum: Great Court roof

Starting with the Great Court roof at the British museum, it spans an area of approximately 95 m x 74 m which is covered by almost 5000 steel elements, Sischka (2000). To recreate this, a base surface equal to the original is first created. This is then subdivided into four smaller surfaces which are meshed individually by the *Mesh Generator* component. The u- and v-division is set to 8 x 12 and mesh 4 is chosen based on its resemblance to the original grid. These are then patched together before the *TNA3D Solver* is used to create the shape presented in Figure 5.44 a).

With a total of 600 elements the computational time is acceptable while the solver obtains a satisfying global shape. To achieve a smoother grid, the output geometry is then subdivided into 2500 branches as presented in Figure 5.44 b).

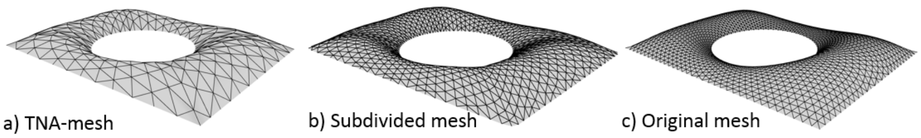


Figure 5.44: a) The coarse mesh and shape obtained with *TNA3D Solver*. b) The same shape as in a), but subdivided into 2500 branches. c) Original mesh and shape of the Great Court roof.

In Figure 5.45 the grid obtained from the TNA analysis is placed on top of the original one. The TNA grid has a more even height than the original with the highest points coinciding with the original shape. The two narrow spans are noticeably lower in the original design; one reason for this could be the support edges of the internal circle being 1.2 metre higher than the external edges in the original design, while the TNA grid has all supports in the same plane. Overall, the *TNA3D Solver* creates a gridshell resembling the original one even with only 12 % of the original elements used.

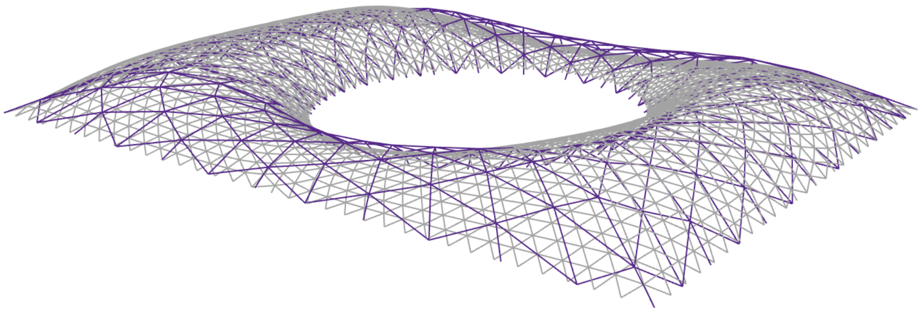


Figure 5.45: Illustrating the different shapes from the TNA (purple) and the original (grey).

### 5.5.2 Arbitrary meshes

Finally, a series of arbitrary shapes are created to illustrate different ways of exploring form with the TNA tools. Using meshes created from the *Mesh generator*, *Lunchbox* components, and random Delaunay meshes as input, the *TNA3D Solver* proves itself as a versatile tool.

**Circular vault**
The first shape is created by using the *Mesh Generator* to mesh a circular surface. Mesh 2 is used to create a rectangular primal grid. With four branches meeting at 90 degrees in the nodes of the primal grid, the dual grid will also be quadratic, as shown in Figure 5.46.

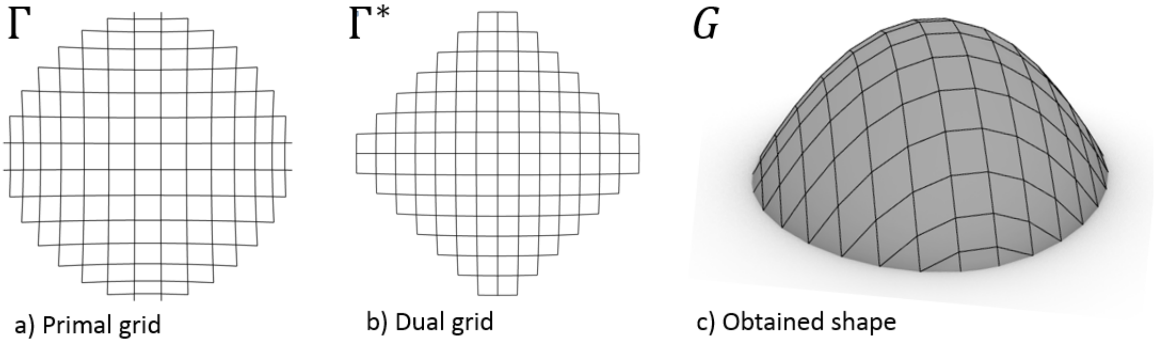


Figure 5.46: Form finding of a circular plane surface.

**Star-shaped vault**
A star-shaped vault is obtained by creating the base surface as four equal parts. Each of these are then meshed with the *Mesh Generator* before they are patched together in order to achieve the primal grid in Figure 5.47.

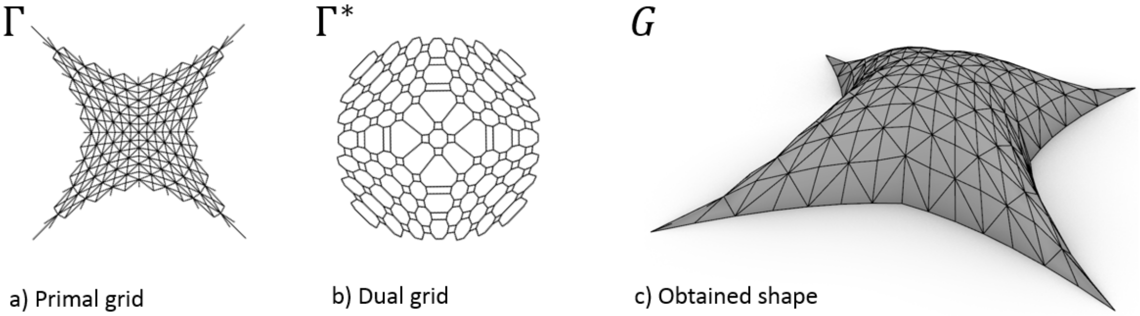


Figure 5.47: Form finding of a star-shaped plane surface.

**Random shape**

An arbitrary shaped planar surface is meshed using the *Mesh Surface* component from Grasshopper. The lines from this can then be fed directly to the *TNA3D Solver*; the results are illustrated in Figure 5.48.

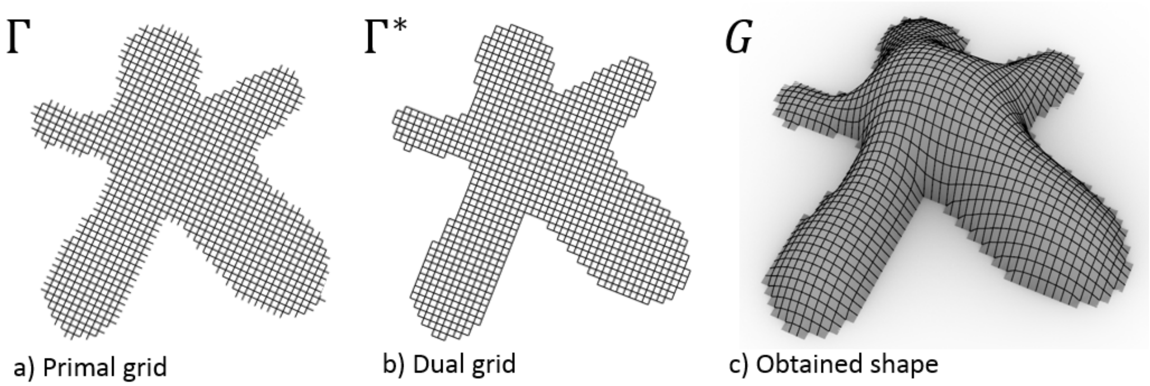


Figure 5.48: Form finding of a random plane surface.

**Lunchbox grid**

From three rectangular surfaces the *Lunchbox* panel component is used to create the primal grid. Some nodes in the primal grid have eight branches, while others have four. As a result the dual grid has both small diamond-shaped faces and larger quadratic-shaped faces with trimmed edges, so that the faces has eight belonging edges.

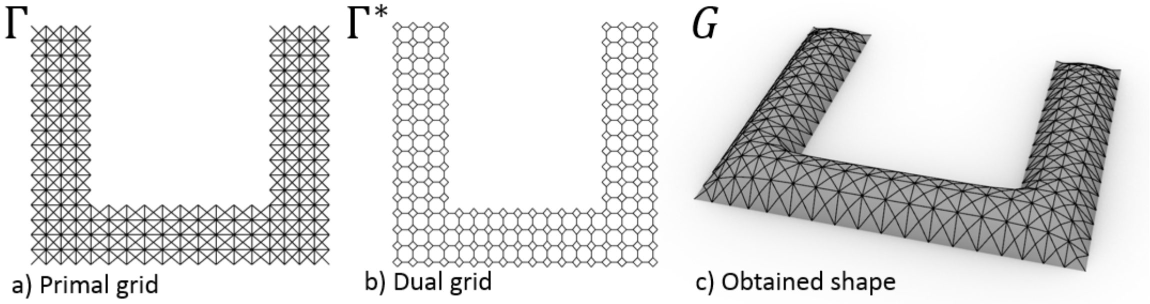


Figure 5.49: Form finding of grid created with *Lunchbox*.

**Delaunay mesh**

A pentagon-shaped surface is randomly populated by a series of points. By creating a Delaunay mesh from these points, the solver finds a dual grid and the belonging geometry. Although it is an inefficient structure with a highly random mesh, the solver finds a solution regardless.

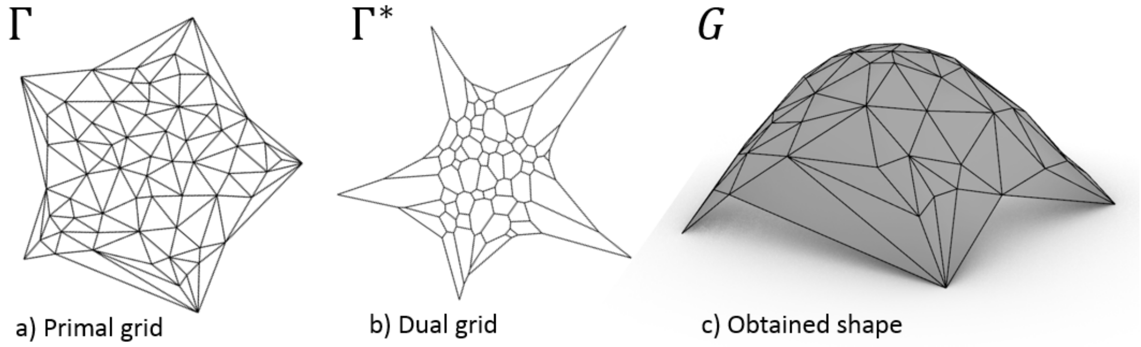


Figure 5.50: Form finding of Delaunay mesh, from random points.

**Openings**

This shape is created by drawing four closed polylines within a rectangle to obtain one grid element in Rhino. Using an array, the primal grid is constructed before it is imported into Grasshopper. Here it is transformed into a mesh with *Weaverbird* components. By deleting polylines in Rhino, the openings in the mesh are created.

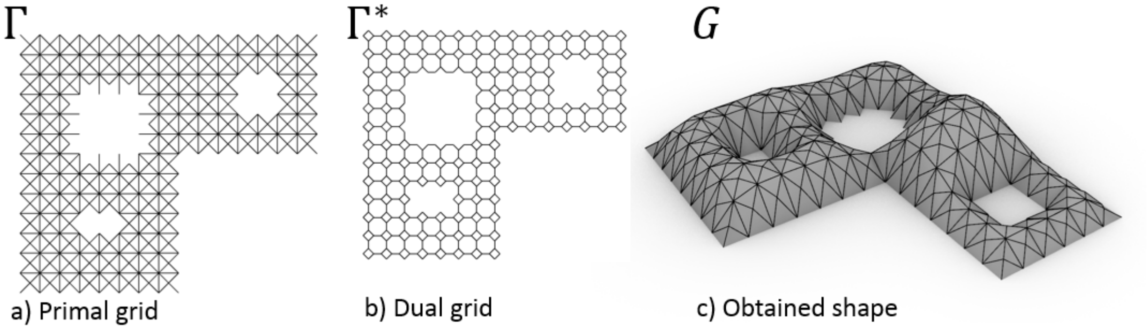


Figure 5.51: Form finding of mesh with several openings.

### Tunnel

Finally, a rectangular surface is modified with multiple *Camel* components. First the *Free Edge* is used to release two edges before the dual grid is modified with the *Internal Arch*. The modified dual grid is then used together with the *Force To Form* component in order to obtain the final geometry.

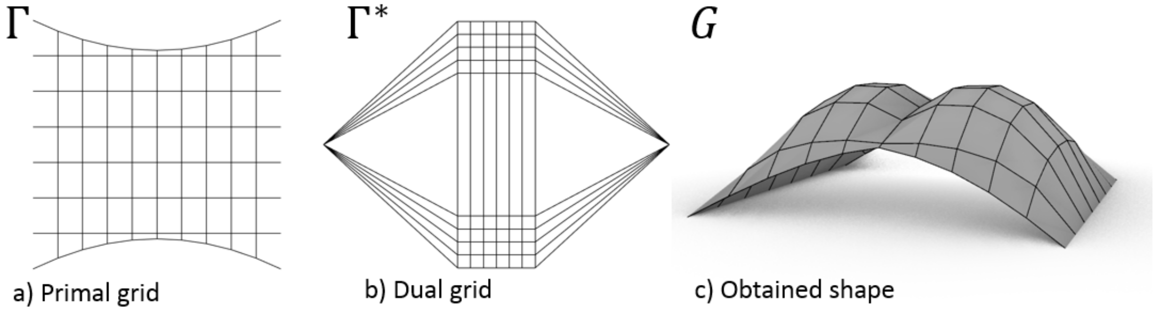


Figure 5.52: Form finding of a tunnel shape, combining *TNA3D Solver*, *Free Edge*, *Internal Arch*, and *Force To Form*.

# 6 Concluding Remarks

Throughout this thesis, an implementation of the form finding method Thrust Network Analysis in a parametric environment has been presented, namely two- and three-dimensional problems. Through a series of case studies research on their performance, as well as their functionality has been undertaken. Below, a final summary of the thesis is provided:

- When implementing the form finding method TNA in Grasshopper, the inclusion of visual feedback of the structural performance enables the architect to evaluate the plausibility of the design simultaneously with the exploration of form.
- The *Camel* components are compatible with a parametric FEA in *Karamba* as well as several optimisation algorithms. This enables the engineer to process complex design proposals delivered from the architect directly in Grasshopper without needing to export the model to other programs.
- Existing research related to the evaluation of the global stability of discrete gridshells was found to be inadequate. The assumption of rigid connections between elements is very common; but as the case studies in this thesis prove; the decisions taken when modelling the joints highly influence the global stability of the gridshell.
- When using TNA to find a structure in global equilibrium, the resulting shape will be optimal only for the applied loading situation. When subjected to other loading conditions, both bending moments and tensile forces may increase.
- For simple geometries the TNA force distribution and utilisation of members corresponded well with the *Karamba* results, but as the structure gradually became more complex, the results consequently differed more from each other.
- By introducing a simplified buckling estimation, which is compatible with the TNA solver, the user can evaluate the global stability of the TNA geometry without having to establish a complete FEA. A correlation between the simplified estimates made with the TNA and FEA result was presented. For more complex geometries further investigation is needed.
- With the use of parametric modelling and form finding methods, it is possible to preserve the interests of both architects and engineers by creating meaningful structures that are also structurally effective.

## 6.1 Further work

During the form finding process only vertical loads are considered. With the TNA originally developed to evaluate funicular masonry structures where the dead load normally governs the loading situation, the effects of horizontal loading is modest. When using the method on gridshells, however, horizontal loads will play a bigger part in the structure's behaviour. Therefore, the inclusion of these would be an useful addition.

While the TNA is a great tool to use when starting from a planar surface and creating a funicular shell, the possibility to use an existing shell as input for a form optimisation such as Best-fit TNA (Van Mele et al. (2014)) could prove to be a useful addition to the current components.

As Case study IV demonstrated, the computational time of the first linear optimisation problem, which establishes the dual grid, quickly becomes a bottleneck in the process, hence limiting the mesh size available for a real-time solution. In his thesis, Block (2009) proposes a mesh division for faster computational time. Incorporating this in the *TNA3D Solver* is an exciting way of increasing the method's functionality.

The introduced 2D verification of discrete gridshells' global stability was evaluated on simple geometries, and an estimate was presented based on the results. When the same estimate was used on the more complex geometries in case studies III and IV, the presented estimate was invalid. Further research on more complex geometries and an incorporation of the results into the components are both interesting topics worth devoting more work to.

The assumptions made with regard to the gridshell's joints highly influenced the global stability of the structure. Further research on how to model these in the conceptual phase is needed. Both for discrete gridshells in general, as well as for implementation in the *Camel* components.

Finally, further development of the components introduced in this thesis and the introduction of new ones are interesting topics for future studies. Apart from the TNA solver, modifications of reciprocal diagrams, free edges, and mesh generation would benefit from further work. Both regarding functionality and user-friendliness.

# References

Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C. (2014), *Shell Structures For Architecture*, Routledge, London and New York.

ArchDaily (2018), 'Padre pio pilgrimage church', `https://www.archdaily.com/900757/padre-pio-pilgrimage-church-renzo-piano-building-workshop`. Accessed: 2020-02-27.

Barnes, M. R. (1999), 'Form finding and analysis of tension structures by dynamic relaxation'.

Bell, K. (2018), *Matrisestatikk*, Fagbokforlaget, Kanalveien 51, Bergen.

Block, P. (2009), Thrust Network Analysis: Exploring Three-dimensional Equilibrium, PhD thesis.

Block, P., DeJong, M. & Ochsendorf, J. (2006), 'As hangs the flexible line: Equilibrium of masonry arches', *Nexus Network Journal* **8**(2), 13–24.

Brown, T. (2008), 'Olympic stadium, frei otto, 1972', `https://www.flickr.com/photos/atelier_flir/2751390103/in/photolist-5c8Aqp-rKZQKy-5ccRM3-rL7JtF-5ccQsd-s3vzPg-s3roNC-rKYFKE-s1h6jy-rKZNem-r6`
Accessed: 2020-03-09.

Bulenda, T. & Knippers, J. (2001), 'Stability of grid shells', *Computers  Structures* **79**, 1161–1174.

Calladine, C. R. (1989), *Theory of shell structures*, Cambridge university press.

Carlson, S. C. (2015), 'Catenary', `https://www.britannica.com/science/catenary`. Accessed: 2020-03-20.

Cichocka, J. M., Migalska, A., Browne, W. N. & Rodriguez, E. (2017), Silvereye–the implementation of particle swarm optimization algorithm in a design optimization tool, *in* 'International Conference on Computer-Aided Architectural Design Futures', Springer, pp. 151–169.

Day, A. (1965), 'An introduction to dynamic relaxation(dynamic relaxation method for structural analysis, using computer to calculate internal forces following development from initially unloaded state)', *The engineer* **219**, 218–221.

Douthe, C., Baverel, O. & Caron, J.-F. (2006), 'Form-finding of a grid shell in composite materials', *Journal of the International Association for Shell and Spatial structures* **47**(1), 53–62.

Eurocode 5 (2010), 'Design of timber structures - part 1-1: General - common rules and rules for buildings'.

*Galapagos* (2020), https://www.grasshopper3d.com/group/galapagos. Accessed: 2020-05-10.

Giel, I. (2010), 'Die multihalle ist eine halle mit einem mehrfach gekrümmten gitter aus holzleisten', https://web.archive.org/web/20161014215546/http://www.panoramio.com/photo/36599877. Accessed: 2020-05-18.

*Google.OR-Tools* (n.d.), https://developers.google.com/optimization.

*Grasshopper* (2020), https://www.grasshopper3d.com/. Accessed: 2020-03-09.

Happold, E. & Liddell, W. I. (1975), 'Timber lattice roof for the mannheim bundesgartenschau', *The structural engineer* **53**(3), 99–135.

Harris, R., Romer, J., Kelly, O. & Johnson, S. (2003), 'Design and construction of the downland gridshell', *Building Research & Information* **31**(6), 427–454.

Hibbeler, R. C. (2014), *Mechanics of materials*, Pearson.

Hwang, K., Knippers, J. & PARK, S.-W. (2009), 'Influence of various types of node conectors on the buckling loads of grid shells'.

Jones, R. M. (2006), *Buckling of Bars, Plates, and Shells*, Bull Ridge publishing.

*Kangaroo* (2020), https://www.grasshopper3d.com/group/kangaroo. Accessed: 2020-05-10.

*Karamba3D* (2020), https://manual.karamba3d.com/. Accessed: 2020-05-10.

Kilian, A. (2004), 'Linking digital hanging chain models to fabrication'.

Kilian, A. & Ochsendorf, J. (2005), 'Particle-spring systems for structural form finding', *Journal of the international association for shell and spatial structures* **46**(2), 77–84.

Kindem, S. (1979), *Mekanikk*, Fabritius.

Kirsch, D. (2012), 'Palazzo dello sport', https://www.grasshopper3d.com/. Accessed: 2020-03-09.

Larsson, S. (2018), 'Design implications of rigid timber gridshells'.

Liddell, I. (2015), 'Hanging chain model multihalle', https://www.researchgate.net/figure/Final-Hanging-chain-model-for-Mannheim_fig2_283164806. Accessed: 2020-05-18.

*Lunchbox* (2020), https://www.grasshopper3d.com/group/lunchbox. Accessed: 2020-05-10.

Malek, S., Wierzbicki, T. & Ochsendorf, J. (2014), 'Buckling of spherical cap gridshells: A numerical and analytical study revisiting the concept of the equivalent continuum', *Engineering Structures* **75**, 288–298.

MatWeb (2020), 'Limestone', `http://www.matweb.com/search/datasheet.aspx?matguid=87597d62662c46a7a308b11e16c563c6&ckck=`. Accessed: 2020-03-23.

Maxwell, J. C. (1864), 'Xlv. on reciprocal figures and diagrams of forces', *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **27**(182), 250–261.

Mesnil, R., Douthe, C., Baverel, O. & Léger, B. (2017), 'Linear buckling of quadrangular and kagome gridshells: A comparative assessment', *Engineering Structures* **132**, 337–348.

Nouri-Baranger, T. (2004), 'Computational methods for tension-loaded structures'.

Orton, A. (2013), *The way we build now: form, scale and technique*, Taylor & Francis.

Peteinarelis, A. (2016), 'Frei otto's contribution - legacy to parametric design and material computation'.

*Plankton* (2017), `https://github.com/meshmash/Plankton/releases`. Accessed: 2020-05-01.

Pourebrahimi, M., Fallah, M. H. & Maghareh, M. R. (2015), 'Gridshells, downland gridshells: A creative mechanism in setting up structure', *Bulletin of Environment, Pharmacology and Life Sciences* .

*Rhinoceros* (2020), `https://www.rhino3d.com/`. Accessed: 2020-05-10.

Robin-Angelo (2012), 'Chiesa san pio da pietrelacina', `https://www.flickr.com/photos/vorenus/8057966402/`. Accessed: 2020-03-09.

Sánchez, J., Serna, M. Á. & Morer, P. (2007), 'A multi-step force–density method and surface-fitting approach for the preliminary shape design of tensile structures', *Engineering Structures* **29**(8), 1966–1976.

Schek, H.-J. (1974), 'The force density method for form finding and computation of general networks', *Computer methods in applied mechanics and engineering* **3**(1), 115–134.

Shiveta Singh, T. E. o. E. B. (2005), 'Arch', `https://www.britannica.com/technology/arch-architecture`. Accessed: 2020-04-30.

Sischka, J. (2000), 'Engineering the construction of the great court roof for the british museum', *Widespan roof structures* pp. 199 –207.

Timber Structures (2016), 'Glued laminated timber and glued solid timber - requirements'.

Van Mele, T., Panozzo, D., Sorkine-Hornung, O. & Block, P. (2014), 'Best-fit thrust network analysis', *Shell Structures for Architecture-Form Finding and Optimization* pp. 157–170.

Wang, K. (2011), 'L'oceanogràfic', `https://www.flickr.com/photos/kentwang/7367669036/`. Accessed: 2020-03-09.

*Weaverbird* (2020), https://www.grasshopper3d.com/group/weaverbird. Accessed: 2020-05-10.

# Appendix A

**Files submitted with the thesis:**

The project files created as a part of this thesis are delivered and available at Github and Zenodo upon request.

# Appendix B

**Videos:**

Three introduction videos explaining how the developed components are working is also uploaded as part of the thesis; these can be found at the NTNU's CSDG YouTube channel.

- *Simple TNA Demo*: This video illustrates how the developed components interact with each each other in order to establish a simple TNA. The video also illustrate the simple capacity control developed.
- *From TNA to Karamba*: Here, a shape created with TNA is fed directly into *Karamba*.
- *Demo Compost Hall*: A simplified version of Case Study IV is presented here.

# Appendix C

### Conference Paper

As a part of the final submission is a conference paper submitted to the *International Conference on Spatial Structures* 2020.

Esjeholm, Haakonsen, Instanes

Design & Evaluation of Discrete Gridshells with Thrust Network Analysis

NTNU
Norwegian University of
Science and Technology