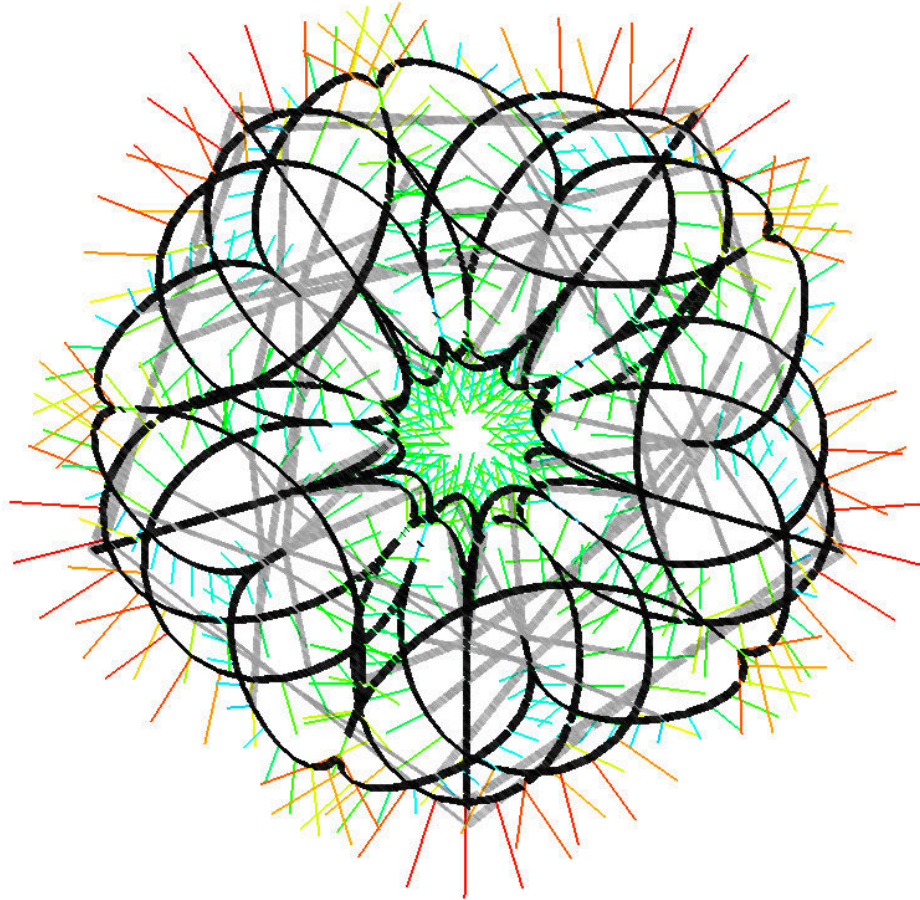


Calibrated Modelling of Form-active Structures



by

Cecilie Søs Brandt-Olsen

Master's thesis in Architectural Engineering

The Technical University of Denmark

June 2016

CALIBRATED MODELLING OF FORM-ACTIVE STRUCTURES

Master of Science in Architectural Engineering

Department of Civil Engineering

The Technical University of Denmark

AUTHOR

Cecilie Brandt-Olsen, s103539

c.brandtolsen@gmail.com

SUPERVISORS - DTU

Henrik Almegaard

ha@henrikalmegaard.dk

Vasehøjvej 10

2920 Charlottenlund

Associate Professor Christian Rønne

chrir@byg.dtu.dk

Brovej 118

2800 Kgs. Lyngby

BUSINESS PARTNER - FORMAT ENGINEERS

Stephen Melville

sm@formatengineers.com

James Solly

js@formatengineers.com

8a Bartlett Street, Studio 3

Bath BA1 2QZ

United Kingdom

PROJECT

Period: 04.01.16 - 04.06.16

The project counts for 30 ECTS

Signature of Author Cecilie Søs Brandt-Olsen

Abstract

Form-active structures are a more rare building type but very eye-catching due to their slenderness, double curvature and overall elegance. From a structural perspective, they are particularly interesting because their shapes cannot be predefined but necessitate a form-finding step before the actual structural analysis. Additionally, their flexibility means that they undergo large deformations when subjected to external loads. This makes them difficult to analyse as finite element software is based on the assumption of small displacements. Some packages have the capability of analysing this more advanced behaviour by storing the stress state from a form-finding step as a separate load case, which are subsequently imposed during the analysis. This modelling set-up is not easily defined when the system becomes sufficiently complex; it is also very time consuming, making an exploration of multiple configurations infeasible. The correlation between the form-finding and analysis requires that both steps are performed within the same software to avoid losing important information, which furthermore has the effect of isolating the architect from the design process.

This research offers the ability to analyse form-active structures in a much more interactive environment, which encourages an informed exploration in the early design stage where both architect and engineer are involved in the process. This is facilitated by the improved stability of the physics constraint solver “Kangaroo2” developed by Daniel Piker, which makes it possible to input real material properties and thereby simulate accurate structural behaviour with meaningful output values. The underlying dynamic relaxation solving technique inherently deals with the large deformations associated with form-active structures. The behaviour of cables, bars, beams (with rotational symmetric cross section) and simple supports have been implemented as a plug-in that builds on top of Kangaroo2. Relevant case studies provided by Format Engineers combined with a collaboration centered around a Smart Geometry workshop have pushed the developments forward and enhanced the applicability of the design tool in practice.

Acknowledgement

Thank you to my DTU supervisor Henrik Almegaard for being an inspiration and support throughout all of my projects in the last six years.

Thank you to my other DTU supervisor Christian Rønne for his endless support to make this specialisation in computational design possible.

Thank you to my two supervisors from Format Engineers, Stephen Melville and James Solly, for providing me with real projects to apply my research to and their enthusiasm about the outcome.

Thank you to Anders Holden Deleuran, Gregory Quinn, Daniel Piker, Will Pearson, Harri Lewis and Dragos Naicu for the many discussions related to this thesis topic, a successful Smart Geometry cluster and being a big motivation throughout this project.

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 1 |
| 1.1 | Definition | 1 |
| 1.2 | Analysis of form-active structures | 3 |
| 1.3 | Parametric and performance based design | 4 |
| 1.4 | Aim | 4 |
| 1.5 | Thesis structure | 5 |
| 2 | Literature & software review | 7 |
| 2.1 | The OnGreening Pavilion | 7 |
| 2.1.1 | Shape generation | 8 |
| 2.1.2 | Structural analysis | 9 |
| 2.1.3 | Modelling and analysis of elastica curves with Karamba | 9 |
| 2.1.4 | Remarks | 14 |
| 2.2 | Atmeture | 14 |
| 2.2.1 | Shape generation | 15 |
| 2.2.2 | Structural analysis | 16 |
| 2.2.3 | Modelling of prestress with Karamba | 17 |
| 2.2.4 | Remarks | 18 |
| 2.3 | Hybrid Tower | 19 |
| 2.3.1 | Hybrid concept | 20 |
| 2.3.2 | Design analysis workflow | 20 |
| 2.3.3 | Structural analysis | 21 |
| 2.3.4 | Improvements | 21 |
| 2.3.5 | Remarks | 22 |
| 2.4 | Research direction | 22 |

| | | |
|----------|---|-----------|
| 3 | Kangaroo2 | 25 |
| 3.1 | Goals | 26 |
| 3.1.1 | Scripting custom goals | 27 |
| 3.1.2 | Example: perpendicularity constraint | 29 |
| 3.2 | Workflow | 32 |
| 3.3 | Solver | 33 |
| 3.4 | Convergence speed | 34 |
| 4 | K2Engineering | 37 |
| 4.1 | Supports | 37 |
| 4.1.1 | Implementation | 38 |
| 4.2 | Bar | 40 |
| 4.2.1 | Implementation | 40 |
| 4.3 | Cable | 43 |
| 4.3.1 | Implementation | 43 |
| 4.4 | Rod | 45 |
| 4.4.1 | Moments and shear forces | 46 |
| 4.4.2 | Actively bent or fabricated curved elements | 47 |
| 4.4.3 | Implementation | 48 |
| 4.4.4 | Beam element | 49 |
| 4.4.5 | Example: Elastica | 51 |
| 4.5 | Goal limitations and further developments | 53 |
| 4.5.1 | Biaxial bending and torsion | 53 |
| 4.5.2 | Constant strain triangle | 53 |
| 4.6 | Convergence | 54 |
| 4.7 | Stress summation | 54 |
| 4.7.1 | Implementation | 55 |
| 4.8 | Non-linear buckling | 55 |
| 4.8.1 | Implementation | 57 |
| 4.8.2 | Example: Elastica buckling with and without prestress | 58 |
| 5 | Smart Geometry 2016 | 63 |
| 5.1 | Workshop proposal | 63 |
| 5.2 | Projection board | 64 |
| 5.3 | Modelling pipeline | 66 |
| 5.4 | Tower challenge | 68 |

| | | |
|----------|---|------------|
| 6 | GFRP gridshell design | 71 |
| 6.1 | Brief | 71 |
| 6.2 | Shape generation | 72 |
| 6.3 | Material | 73 |
| 6.4 | Design exploration | 74 |
| 6.4.1 | Design 1 | 76 |
| 6.4.2 | Design 2 | 77 |
| 6.4.3 | Design 3 | 78 |
| 6.4.4 | Design 4 | 79 |
| 6.4.5 | Design 5 | 80 |
| 6.5 | Design selection and refinement | 81 |
| 6.5.1 | Analysis results | 82 |
| 6.5.2 | Buckling | 86 |
| 6.5.3 | Connections and supports | 89 |
| 6.6 | Final design | 91 |
| 7 | Conclusions | 93 |
| 7.1 | Summary | 93 |
| 7.2 | Discussion and future work | 94 |
| 7.3 | Perspective | 95 |
| 7.3.1 | Clamshell Tent | 95 |
| 7.3.2 | UWE Pavilion | 97 |
| A | Structural validation | 103 |
| A.1 | Axial behaviour | 103 |
| A.1.1 | Tension | 103 |
| A.1.2 | Pretension | 104 |
| A.1.3 | Catenary | 104 |
| A.1.4 | Catenary with prestress | 105 |
| A.1.5 | Truss | 106 |
| A.2 | Bending behaviour | 107 |
| A.2.1 | Simple supported beam | 107 |
| A.2.2 | Cantilever beam | 109 |
| A.2.3 | Frame | 111 |
| A.2.4 | Arch | 113 |
| A.3 | Buckling | 115 |
| A.3.1 | Column | 115 |
| B | K2Engineering overview | 117 |

List of Figures

| | |
|--|----|
| 1.1.1 The Olympic stadium in Munich | 2 |
| 1.1.2 The ICD/ITKE Research Pavilion 2010 | 3 |
| 2.1.1 The OnGreening Pavilion | 7 |
| 2.1.2 OnGreening form-finding process | 8 |
| 2.1.3 Elastica form-finding set-up | 10 |
| 2.1.4 Results from elastica curve study | 10 |
| 2.1.5 Elastica curves generated by Karamba | 11 |
| 2.1.6 Bending moments from curvature | 12 |
| 2.1.7 Superposition of bending moments for elastica curve | 13 |
| 2.2.1 Atmeture, Letchworth 2014 | 14 |
| 2.2.2 ArchiLace shape generation | 15 |
| 2.2.3 Atmeture reinforcement pattern | 16 |
| 2.2.4 Simulating prestress of an elastica curve with Karamba | 17 |
| 2.2.5 Moment plot of elastica curve with prestress (Karamba) | 18 |
| 2.3.1 Interior of the hybrid Tower at CITA | 19 |
| 2.3.2 Hybrid Tower modelling workflow | 20 |
| 3.0.1 Force-based approach | 26 |
| 3.1.1 Position-based approach | 27 |
| 3.1.2 Perpendicular goal behaviour | 29 |
| 3.1.3 Result from perpendicular goal | 31 |
| 3.1.4 Perpendicular goal for Thrust Network Analysis | 32 |
| 3.2.1 Kangaroo2 workflow | 33 |
| 4.2.1 Axial behaviour of a bar element | 40 |
| 4.4.1 Modelling of bending in a 3 DoF system | 45 |
| 4.4.2 Bending moments for a spline | 46 |

| | |
|---|----|
| 4.4.3 Shear forces for a spline | 47 |
| 4.4.4 Elastica curves generated by K2Engineering | 51 |
| 4.4.5 Elastica behaviour with and without prestress | 52 |
| 4.7.1 Summation of axial and bending stresses | 55 |
| 4.8.1 Goats balancing on a flexible steel ribbon | 56 |
| 4.8.2 Different configurations for the buckling analysis of elastica curves | 59 |
| 4.8.3 Buckling of elastica curves subjected to a UDL | 59 |
| 4.8.4 Buckling of elastica curves subjected to a symmetric point load | 60 |
| 4.8.5 Buckling of elastica curves subjected to an asymmetric point load | 60 |
| 5.2.1 Projection board kit | 65 |
| 5.2.2 Projection board setup | 65 |
| 5.3.1 Exploration of hybrid concepts | 67 |
| 5.3.2 Digital recreation of hybrid concept | 68 |
| 5.4.1 Digital model of the tower design | 69 |
| 5.4.2 Final hybrid tower | 70 |
| 6.1.1 Gridshell location at Torvehallerne | 71 |
| 6.2.1 Gridshell shape generation | 73 |
| 6.4.1 The simplified wind load acting on the gridshell | 75 |
| 6.4.2 Gridshell design 1: Geometry and structural performance | 76 |
| 6.4.3 Gridshell design 2: Geometry and structural performance | 77 |
| 6.4.4 Gridshell design 3: Geometry and structural performance | 78 |
| 6.4.5 Gridshell design 4: Geometry and structural performance | 79 |
| 6.4.6 Gridshell design 5: Geometry and structural performance | 80 |
| 6.5.1 Structural performance for load case 1 | 82 |
| 6.5.2 Structural performance for load case 2 | 83 |
| 6.5.3 Structural performance for load case 3 | 84 |
| 6.5.4 Selected deformations of the gridshell during a buckling analysis (part I) | 87 |
| 6.5.5 Selected deformations of the gridshell during a buckling analysis (part II) | 88 |
| 6.5.6 Different options for the GFRP gridshell connection detail | 89 |
| 6.5.7 Support option 1 | 90 |
| 6.5.8 Support option 2 | 90 |
| 6.5.9 Support option 3 | 90 |
| 6.6.1 The final gridshell design | 91 |
| 7.3.1 Clamshell Tent analysis with K2Engineering | 96 |

| | |
|--|-----|
| 7.3.2 Timber gridshell for the graduate year show 2016 at UWE | 98 |
| 7.3.3 Buckling analysis of the UWE gridshell with K2Engineering | 98 |
| | |
| A.1.1 K2Engineering tension example | 104 |
| A.1.2 K2Engineering pretension example | 104 |
| A.1.3 Axial- and reaction forces for a catenary | 105 |
| A.1.4 Analytical solution for the catenary using graphic statics | 105 |
| A.1.5 Axial forces for a prestressed catenary (K2Engineering) | 106 |
| A.1.6 Axial forces for a prestressed catenary (Karamba) | 106 |
| A.1.7 K2Engineering truss example | 107 |
| A.1.8 Karamba truss example | 107 |
| A.2.1 Simple supported beam with moment plot from K2Engineering | 108 |
| A.2.2 Simple supported beam with shear plot from K2Engineering | 108 |
| A.2.3 Cantilever beam with moment plot from K2Engineering | 110 |
| A.2.4 Cantilever beam with shear plot from K2Engineering | 110 |
| A.2.5 Frame with moment plot from K2Engineering | 111 |
| A.2.6 Frame with shear plot from K2Engineering | 112 |
| A.2.7 Frame with moment plot from Karamba | 112 |
| A.2.8 Frame with shear plot from Karamba | 113 |
| A.2.9 Arch with moment plot from K2Engineering | 114 |
| A.2.10 Arch with shear plot from K2Engineering | 114 |
| A.2.11 Arch with moment plot from Karamba | 115 |
| A.2.12 Arch with shear plot from Karamba | 115 |
| A.3.1 Column buckling behaviour from K2Engineering | 116 |
| | |
| B.0.1 K2Engineering components | 117 |

Chapter 1

Introduction

This chapter clarifies the meaning of form-active structures and highlights their qualities as well as the difficulties related to the design and analysis of these structures. From this context, the overall aim for this research is identified and a structure of the thesis is outlined.

1.1 Definition

The term “form-active” originates from the categorisation of building structures according to Engel (1997) and is defined as a flexible system, where the shape of the structure in the ideal case coincides precisely with the flow of forces. As a consequence, the architectural form and space of these structures are purely defined from the support and loading conditions and thus cannot be subject to arbitrary free-form design. At the same time this makes them very interesting and noticeable structures. According to Engel’s definition, form-active structures include cable nets, tents, pneumatic and arch structures. Since form follows force, these structures are highly efficient and have the ability to span large distances. However, their flexibility often necessitates that the system is prestressed in order to reduce the deflections and make sure that the structure does not become slack under various load scenarios. Frei Otto was one of the pioneers in relation to form-active structures and the Olympic stadium in Munich from 1972 (Kroll, 2011) as seen in Figure 1.1.1 is a good example of his incredible work in this context.

Inspired by Engel’s categorisation, the term “bending-active” has more recently been introduced to describe curved beams or surfaces, which derive their shape from elastic deformation (bending) of initially straight or planar elements (Lienhard et al., 2013). The ICD/ITKE Research Pavilion from 2010 (Menges, 2010) is a good example of this type of structure as shown in Figure 1.1.2. Like for form-active structures, the shape is dependant on the internal force equilibrium but in this case primarily from bending action rather than tension/compression (as the name indicates). Bending-active structures are generally characterised by the use of thin cross sections of a material with low stiffness in order to achieve sufficient curvature without exceeding the material



Figure 1.1.1 – The Olympic stadium in Munich from 1972 designed by Frei Otto. Copyright Jorge Royan

strength. Even though the initial bending of the elements introduces prestress to the structure, it is motivated by the simplicity of creating curved elements and it makes transportation easier as well because the straight elements are more easily packed. Whilst the bending-active term is relatively new, the technique can be traced much further back in the context of timber gridshells such as the Multihalle Mannheim from 1974 by Frei Otto (Naicu et al., 2014). Here timber laths in a 0.5 m grid arrangement were elastically deformed into the intended three dimensional shape.

The differentiation between the terms “form-active” and “bending-active” is useful to make a distinction between how the system primarily resists the applied loads i.e. via axial or bending action. However, the two systems have many shared properties such as being light-weight, flexible, undergo large deformations when subjected to various loads and the fact that their shape cannot be drawn beforehand as it is a result of the support conditions, geometrical connectivity of members, loads and materials. Thus, the design of these structures necessitates a form-finding step. These common characteristics suggest that they both fit into the same category of “form-active” structures, which is what the title of this thesis refers to.

More recent structures also combine form-active and bending-active elements into one system and are referred to as “hybrids”. The use of membranes to restrain bending active elements is an example of such a system. This type of structure still has much potential that awaits to be explored.



Figure 1.1.2 – The ICD/ITKE Research Pavilion 2010 made from thin, elastically bent plywood sheets

1.2 Analysis of form-active structures

Due to the flexibility of these structures, they are more difficult to analyse because most functionality in finite element programs are based on the assumption of small displacements. Several of the more advanced FEA packages have an option to perform a form-finding step though and the stress state from this step can be stored as a load case and subsequently imposed in a non-linear analysis. The finite element software SOFiSTiK (2016) is especially known for handling structures with large deformations well. The two step process implies that there is an interdependency between the form-finding and analysis, which forces the two steps to be performed within the same software in order to transfer the necessary information. However, this gives the architect little opportunity to be part of a design process.

The rigorous finite element environment generates very accurate and detailed results but at the cost of being very time consuming to set up a model. As a consequence, it is only feasible to analyse and refine one or two options whereas it would be more beneficial in the conceptual design stage to explore a variety of shapes with less details but enough to make informed design decisions.

1.3 Parametric and performance based design

During the last ten years parametric modelling software has become very popular in both architecture and engineering. The parametric approach makes repetitive tasks easier and provides a flexible environment to investigate the influence of the different parameters that define the design space without having to model everything from scratch. Especially the 3D modelling software Rhinoceros (McNeel, 2016b) with its parametric modelling plug-in Grasshopper (McNeel, 2016a) has gained much popularity in this context. A combination of a very active community, that has encouraged the continuous development of additional Grasshopper plug-ins, and the ever increasing requirements to energy and material efficiency in the building industry, has pushed a performance based design approach forwards. This approach aims to establish a live feedback loop between the geometry and certain aspects of its performance such that any changes to the shape can be directly evaluated and used to make informed design decisions. This is achieved by the integration of analysis software in the Grasshopper environment to avoid data exchange issues and increase the speed.

From a structural perspective, the two most interesting plug-ins are the finite element software Karamba3D (2016) and the physics engine Kangaroo (Piker, 2016b). The latter has many different applications, one of them being structural form-finding. This allows both architects and engineers to explore a variety of shapes emerging from different boundary conditions in a very playful environment. However, once a desirable shape is found, the geometry has to be regenerated inside a finite element program in order to transfer the information about the stress state to the analysis. So whilst this approach helps to engage the architect in the design process of form-active structures, the workflow is still interrupted once a satisfactory result from an architectural perspective has been obtained, which eventually means that the structural behaviour of only one or two options are analysed.

1.4 Aim

The aim of this research is to improve the current workflow related to the design and analysis of form-active structures within the Rhino/Grasshopper environment.

For this purpose, a tool has been developed, which encourages an exploration of form-active structures in this more spontaneous modelling environment and balances interactive speed with a level of analysis detail that is sufficient to make informed design decisions in the conceptual stage. Hence, this tool does not try to replace existing finite element software but is rather targeted to help the development of a more intuitive understanding of the form-active structures and explore a variety of design options before one is selected for detailed design.

1.5 Thesis structure

Chapter 2: reviews three recent projects in order to identify current design approaches and their limitations and thereby specify this research path.

Chapter 3: describes the software framework, which the developments in this research build upon.

Chapter 4: outlines relevant theory and its implementation along with small examples to validate the behaviour and create a progress history.

Chapter 5: demonstrates the integration of the developed tool in a modelling pipeline for a Smart Geometry workshop.

Chapter 6: contains a gridshell case study, which shows the applicability and advantages of the developed design tool.

Chapter 7: concludes this thesis with a summary of what this research has delivered and discusses the limitations, future work and design potential in practice.

Chapter 2

Literature & software review

The purpose of this chapter is to provide an overview of three recent projects in order to highlight current design challenges and thereby more specifically identify the direction of this research. The three projects are all related to form-active structures and the design and analysis methods revolve around the Rhino/Grasshopper environment. Relevant aspects of the projects are described and accompanied with small studies using similar software to better understand the concepts and their limitations.

2.1 The OnGreening Pavilion

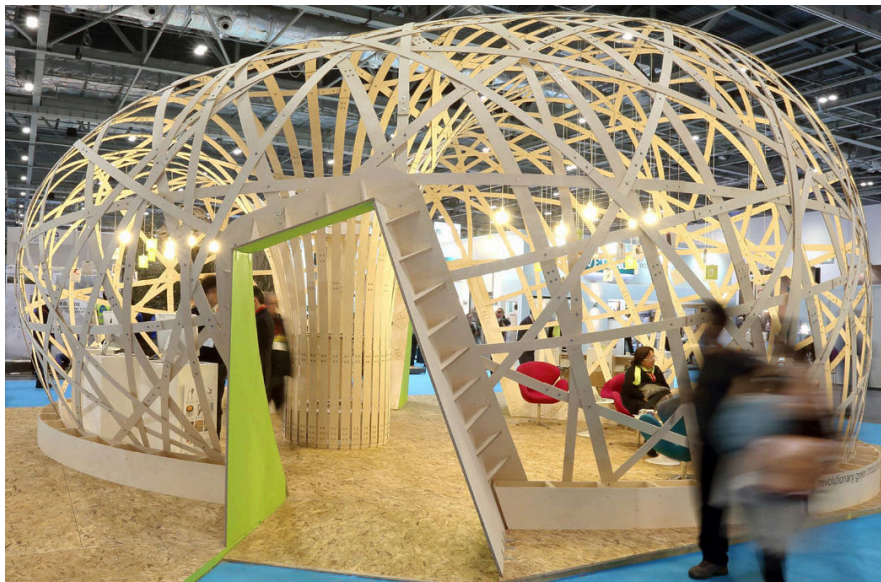


Figure 2.1.1 – The OnGreening Pavilion

The OnGreening Pavilion was a timber gridshell designed in a collaboration between OnGreening Ltd and Ramboll Computational Design and erected at Ecobuild 2014 in London. The pavilion had to fit within a footprint of 10 x 8 m and used for an indoor public exhibition on sustainable buildings (Harding et al., 2014).

2.1.1 Shape generation

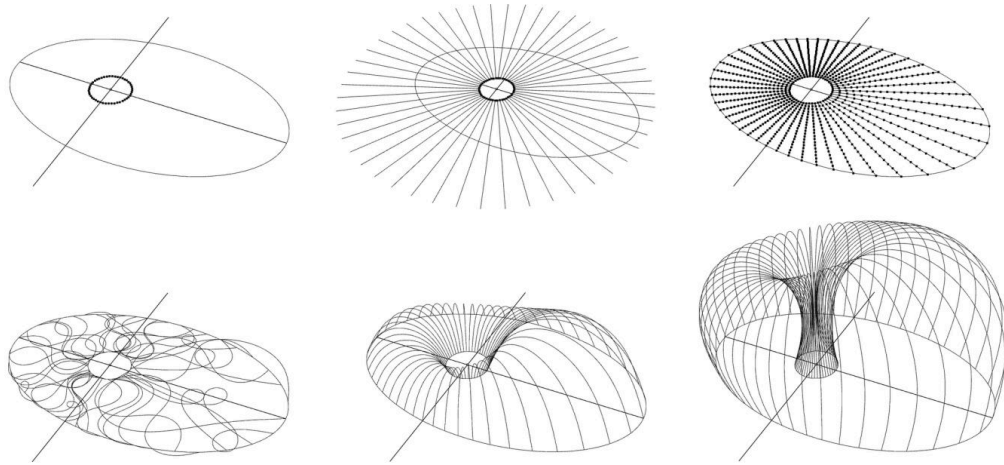


Figure 2.1.2 – OnGreening form-finding process (Harding et al., 2014)

The overall concept for the pavilion was decided to be a bending-active timber gridshell. Due to the size restrictions, a funnel scheme was adopted to maximise the usable volume. The gridshell was realised through a sequential assembly process, where each lath was bent individually rather than the more commonly used approach of predefining a flat grid topology and pushing it into shape. This assembly method was more appropriate for a funnel scheme and also allowed a larger design space to be explored in order to meet client requests.

The spatial shape of the pavilion emerged through a simulation of elastica curves in radial directions from the funnel centre as shown in Figure 2.1.2. The elastica curves had both axial and bending stiffness and thus represented the behaviour of the straight timber laths that were bent into shape. The Grasshopper plug-in Kangaroo was used for this form-finding process. In total 32 curves were generated this way and formed the primary structural system.

A closed lofted surface was subsequently created through these curves and used to define the secondary structural layer. Since it was intended to build the gridshell from straight timber laths with a rectangular cross section, it was beneficial to let the secondary structure follow geodesic lines on the surface as these unrolled to straight lines in the plane. The geodesics also had the advantage of avoiding bending about the strong axis of the cross section (not considering any loads), which could potentially be hard to achieve. For the same reason, it was decided to change the primary structure to geodesics as well with the same start and end points as the initial elastica curves. This meant that the curves were not initially in their spatial equilibrium

position but it was assumed that they still approximated the shape of the bent laths and that the secondary layer of random geodesics furthermore braced the structure and held it all in place (Harding et al., 2014).

2.1.2 Structural analysis

The finite element Grasshopper plug-in Karamba was used for the structural analysis of the pavilion in an integrated workflow such that the influence of the density and location of the random geodesics were evaluated immediately. The modelling of this layer needed manual adjustment to avoid cluttering in certain areas so rather than searching for an optimal solution, the workflow allowed many different variations to be explored in a short time guided by human intuition. The random nature of the secondary structural layer had the advantage of making the gridshell less directional and thereby function more like a monocoque structure. The gridshell was analysed under the influence of self-weight and accidental point loads. Following the principle of superposition, the stresses in the members resulting from the applied loads were added to the stresses arising from the initial curvature of the laths.

2.1.3 Modelling and analysis of elastica curves with Karamba

The purpose of this small study is to investigate the outlined procedure and thereby gain a better understanding of the design decisions that were made for the OnGreening Pavilion. It is attempted to perform both the modelling and analysis of an elastica curve with Karamba to use the same software throughout the study.

MODELLING

Karamba's large deformation analysis is used for the shape generation of an elastica curve. In order to evaluate whether the generated shape is in fact an elastica curve, a study carried out by Adriaenssens and Barnes (1999) is used as benchmark. This study uses a dynamic relaxation approach to generate elastica curves from four different axial loads (above the critical Euler load) and compares the width and height ratios with an analytical solution as well. The set-up, material properties and results from this study are shown in Figure 2.1.3, Table 2.1.1 and Figure 2.1.4 respectively.

| Diameter [mm] | Area [mm ²] | Moment of inertia [mm ⁴] | Young's modulus [MPa] |
|---------------|-------------------------|--------------------------------------|-----------------------|
| 126.5 | $12.57 \cdot 10^3$ | $12.57 \cdot 10^6$ | 7958 |

Table 2.1.1 – Derived material properties from the benchmark elastica study assuming a solid circular cross section

A similar set-up within the Karamba framework is attempted to be defined. The first approach applies a pinned support in one end of the straight beam, a moveable support in the other end, a horizontal load at the location of the moveable support of magnitude $2F$ and a small vertical force at the middle to cause out-of-plane buckling. Whilst this set-up generates elastica like

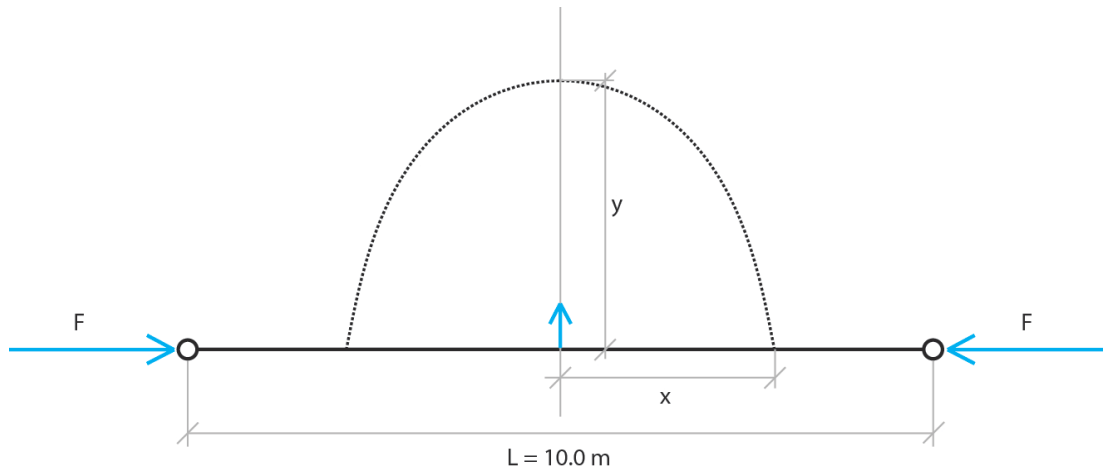
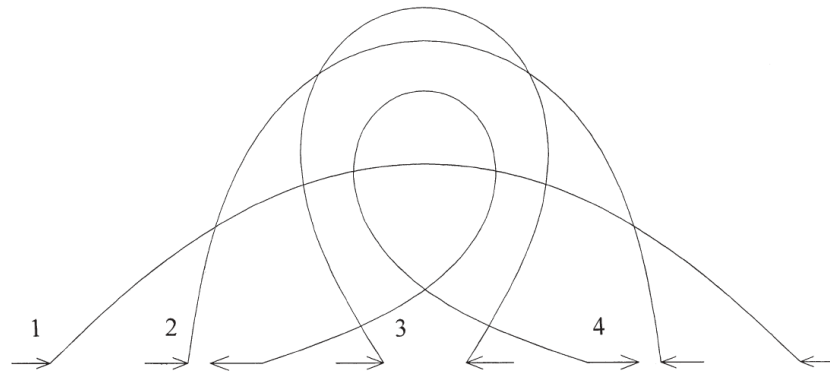


Figure 2.1.3 – Elastica form-finding set-up. A small vertical force at the middle is necessary to make the rod buckle out-of-plane



| Buckled states | 1 | | 2 | | 3 | | 4 | |
|--------------------------|----------|--------|----------|--------|----------|--------|----------|--------|
| Load | 10.48 kN | | 12.67 kN | | 18.46 kN | | 39.48 kN | |
| Central node | x/L | y/L | x/L | y/L | x/L | y/L | x/L | y/L |
| Analytical | 0.4405 | 0.2110 | 0.2800 | 0.3595 | 0.0615 | 0.4015 | -0.1700 | 0.3125 |
| Numerical 64 segments | 0.4413 | 0.2099 | 0.2848 | 0.3572 | 0.0614 | 0.4019 | -0.1699 | 0.3130 |

Figure 2.1.4 – Results from elastica curve study by Adriaenssens and Barnes (1999)

curves from a large deformation analysis, the results are heavily dependent on the magnitude of the small vertical force and the method is therefore not ideal. Instead, the set-up is changed to use prescribed displacements of two pinned supports according to the x/L ratios from the benchmark study (referring to Figure 2.1.4) and a small vertical force at the middle (0.1 kN). This set-up is less sensitive to the magnitude of the vertical force and produces the shapes as shown in Figure 2.1.5.

The deviations of the y/L ratio in comparison to the benchmark study are listed in Table 2.1.2 for the four buckled states. It is seen that shape 1-3 closely approximate the benchmark results

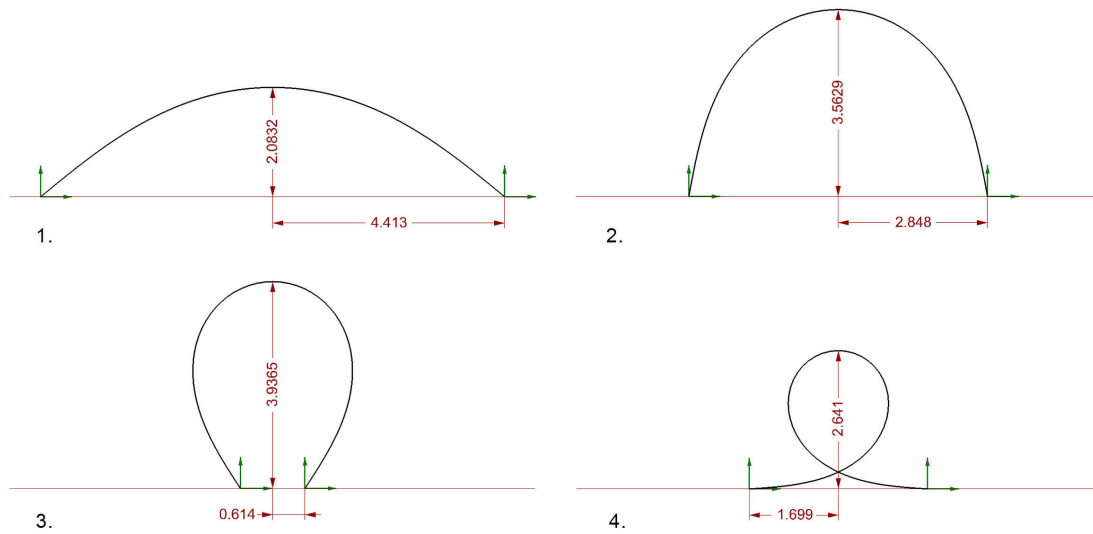


Figure 2.1.5 – Elastica curves generated by Karamba for the four buckled states. The measurements are given in meters

| Buckled state | 1 | 2 | 3 | 4 |
|-------------------|------|------|------|-------|
| y/L deviation [%] | -0.8 | -0.3 | -2.1 | -15.6 |

Table 2.1.2 – Deviation of the y/L ratio between the Karamba results and the benchmark study

whereas shape 4 deviates significantly. The generated shapes from Karamba's large deformation analysis can therefore only be trusted within a certain range. Furthermore, it is not possible to access any information about the forces and moments that arise from the movement. The Karamba development team recognises this drift of the calculated deformed shape from the real behaviour of the structure and has actively decided not to include any force output because of the additional inaccuracy (Preisinger, 2012).

ANALYSIS

The initial prestress from the bending of the rods are calculated in a similar way as described for the OnGreening Pavilion. This calculation uses the relationship between the moment and curvature radius given as $M = \frac{E \cdot I}{R}$ and is integrated in a workflow that uses native Grasshopper components to extract the curvature radius along the curve. Figure 2.1.6 shows the result of such calculation for the four generated elastica curves. The maximum bending moment at the top varies between 22 and 108 kNm.

The shape from the second buckled state with only 0.3% deviation from the benchmark study is used to demonstrate the described principle of superposition to calculate the final stress state when loads are applied. In this case, a vertical point load at the middle of magnitude 10 kN is added and the resulting bending moments from a Karamba analysis are visualised in Figure 2.1.7 (middle). These moments are added to the previously calculated moments from curvature to determine the final moment distribution as shown in Figure 2.1.7 (bottom). It is important

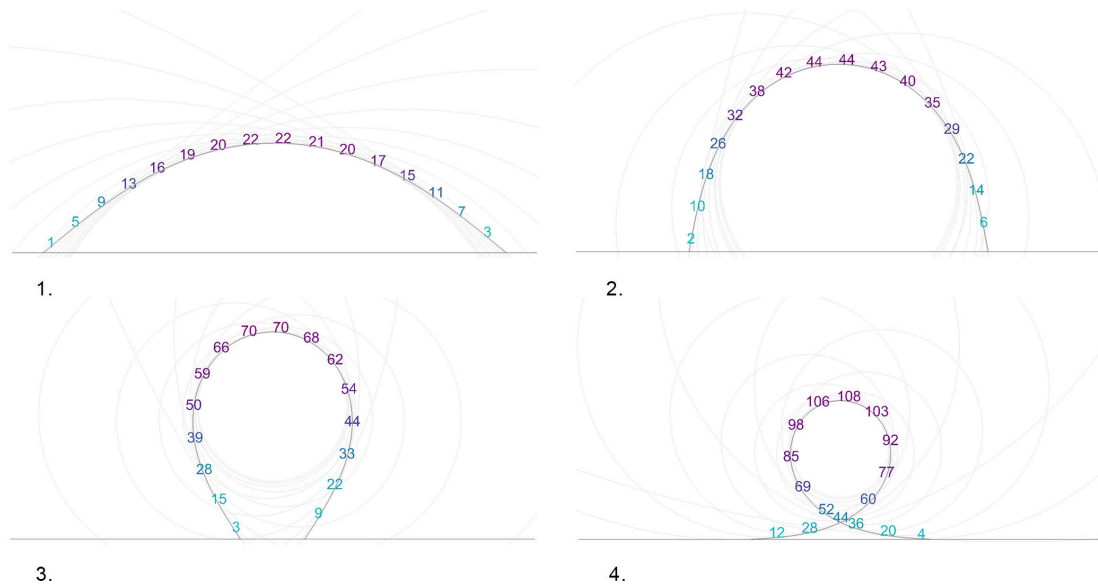


Figure 2.1.6 – Bending moments from curvature for the four buckled states. The measurements are given in kNm. Note that the results are symmetric about a vertical axis but are plotted at slightly different locations along the beam.

to be aware of the sign variation when the moments are added together e.g. the moment from curvature at the top works in the opposite direction of the moment from the applied load, which reduces the final moment value. As a result, the maximum bending moments from superposition are shifted a bit to the left/right side of the middle with a maximum value of 41 kNm. The maximum deflection at the top is 42 mm but is calculated from the applied load only.

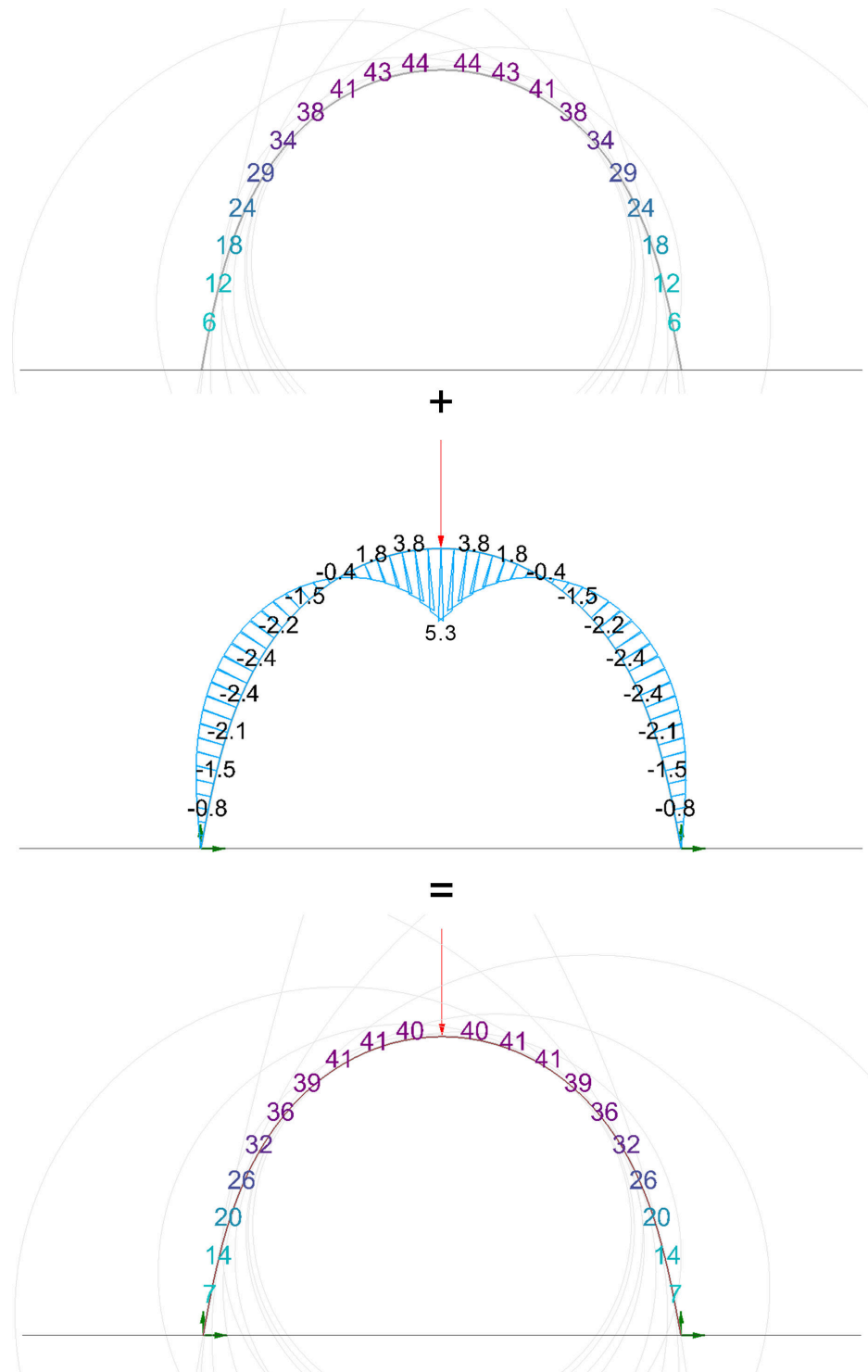


Figure 2.1.7 – Superposition of bending moments (bottom) for elastica curve resulting from initial curvature (top) and point load (middle). The measurements are given in kNm

2.1.4 Remarks

In general, the OnGreening Pavilion highlights several of the common issues related to the design of gridshells made from bending-active laths. It shows how the grid design is highly related to the intended construction method and in this case a sequential assembly strategy was adopted due to the funnel scheme and the choice of using geodesics (to avoid problems related to anisotropic cross section properties).

The OnGreening project used the Grasshopper plug-in Kangaroo for the form-finding of elastica curves. This seems like the better choice as the test of using Karamba's large deformation analysis for this purpose demonstrates that the solution drifts away from the real structural behaviour when the displacements become large. This behaviour is not surprising as Karamba is based on implicit methods, which assume small displacements to begin with. When the displacements are large, which is the case for bending-active structures, it is favourable to use explicit methods and this is exactly what the plug-in Kangaroo is based on.

Neither Kangaroo nor Karamba are capable of outputting any information about the forces from the form-finding process. In light of this limitation, it seems reasonable to use the principle of superposition to take the pre-stress from initial curvature into account. However, when the load case becomes more advanced than a simple point load, it will most likely be difficult to keep track of the signs of the moments and hence calculate the correct moment distribution. It is on the safe side though, if the moments are always added together as absolute values (as it was done for the OnGreening Pavilion).

Whilst the prestress effect is taken into account by this method in relation to the moment distribution, it is not reflected in the displacement calculation and this may become a problem if this is the most critical design parameter.

2.2 Atmeture



Figure 2.2.1 – Atmeture, Letchworth 2014 Loop.pH (2014)

Atmeture was one out of several so-called “ArchiLace” projects designed by Loop.pH (2014). It was built for the “Fire & Fright Festival” in Letchworth in 2014 and had the shape of a 4.3 m tall by 5.5 m wide tunnel. The structure was built from a number of interconnected glass fibre reinforced polymer (GFRP) hoops, which were reinforced in certain locations by bent carbon fibre rods following the hoop pattern. The stability of the structure was ensured by the interlocking of the hoops and the structural analysis of this challenging structure was carried out by Ramboll Computational Design.

2.2.1 Shape generation

The hoop pattern was generated from a number of mesh operations as shown in Figure 2.2.2. From top left to bottom right these included the creation of a triangulated mesh from a three dimensional surface, the dual hexagonal mesh, truncation to create more circular hoops which only intersected at the midpoints of the hexagonal edges and eventually incorporation of the carbon fibre lace (Melville and Nielsen, 2014). The modelling was performed within the Rhino/-Grasshopper environment using the half-edge mesh library Plankton (Piker and Pearson, 2013).

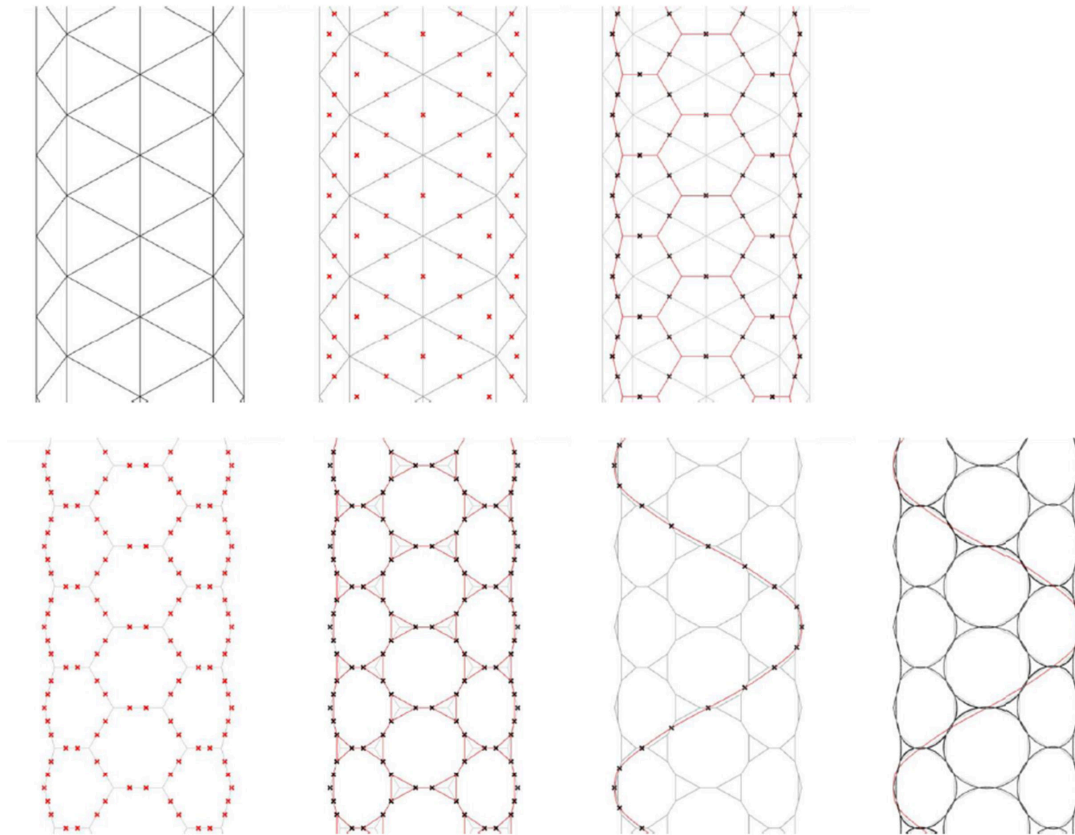


Figure 2.2.2 – ArchiLace shape generation (Melville and Nielsen, 2014)

2.2.2 Structural analysis

Atmeture was a challenging structure to analyse for several reasons: the geometrical complexity, the low stiffness of the GFRP rods resulting in large deformations and the fact that there were no fixings between the hoops causing slippage before the structure interlocked itself. Therefore different physical tests were carried out to observe the structural behaviour and compare that to the results obtained from a finite element analysis in order to evaluate the accuracy of the digital model (Melville and Nielsen, 2014).

Due to the geometrical complexity and in order to investigate several different options with regard to the shape and carbon fibre reinforcement pattern, it was chosen to use the finite element software Karamba. In contrast to the OnGreening Pavilion, the prestress from initial weaving of the carbon fibre rods was simulated by adding bending moments (as equivalent force couples) according to the local curvature. In this case, the desired shape was already defined and hence did not necessitate a form-finding step. It was decided to only model this prestress behaviour for the carbon fibre reinforcement since the closed hoops of GFRP did not exert any forces onto the surrounding structure (zero net force). The slippage effect was not modelled due to the uncertainty of this behaviour.

The structure was tied to the ground by some ballast at the lower parts and subjected to self-weight and wind load. The analysis revealed that local buckling was the critical failure mode for this structure. Different shape variations were investigated and showed that a more squat tunnel cross section was the most efficient. Karamba's linear buckling analysis was used to inform the placement of extra local carbon fibre reinforcement in addition to the initial reinforcement pattern defined by the artist. With these improvements it was shown that the structure was stable for wind speeds up to 13.9 m/s. Figure 2.2.3 shows the final model with reinforcement patterns and supports.

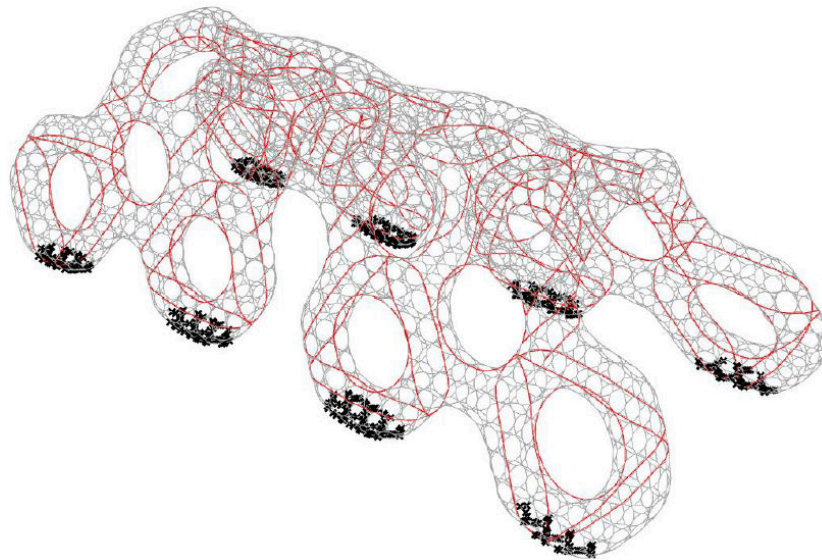


Figure 2.2.3 – Atmeture reinforcement pattern (Melville and Nielsen, 2014)

2.2.3 Modelling of prestress with Karamba

The purpose of the following study is replicate the described method of simulating the prestress effect with Karamba. The same elastica curve (buckled state 2) from Subsection 2.1.3 is used as basis for the testing. The relationship between the moment and curvature radius is used again but this time the equivalent force couple is calculated and applied to the model as external loads. Each force couple corresponds to the forces that are necessary to straighten out that part of the curve.

DEFORMATION

This method is tested on the elastica curve in two cases; firstly with pinned supports in both ends and secondly with one of the supports being released and the results are shown in Figure 2.2.4. For the first case, a second order analysis is used to determine the deformed shape and the maximum deflection at the top is observed to be 12 mm. Ideally the shape would remain unchanged from the prestress load since the elastica curve already represents an initially straight rod that is bent into shape. The small deviation is most likely caused by the inaccuracy related to the form-finding of the elastica curve with Karamba's large deformation analysis (in this case 0.3%).

For the second case, Karamba's large deformation analysis is used to determine the deformed shape from the prestress load. It is observed that the rod almost straightens out completely with a maximum deviation of 230 mm at the middle. The reason why it does not become perfectly straight is because the prestress load is calculated from the initial configuration and the magnitude is therefore constant throughout the analysis even though the curvature changes. The direction is however continuously updated (option for the large deformation analysis), which minimises the deviation. In most cases, the deformation from the initial configuration will be much smaller and the method thus more appropriate.

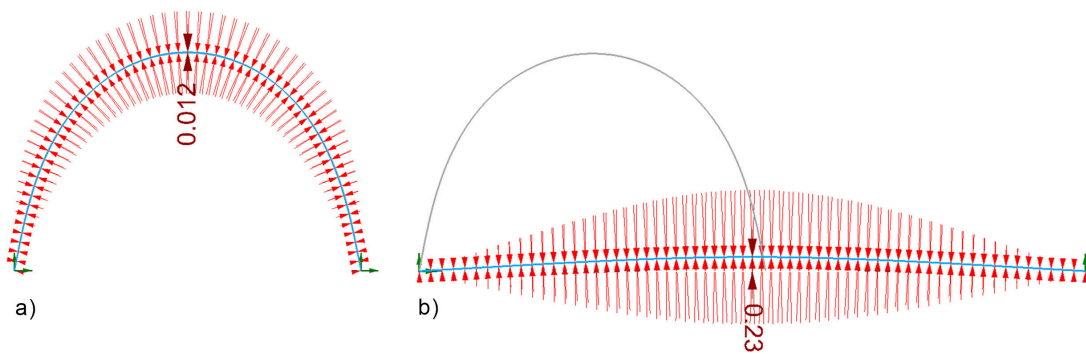


Figure 2.2.4 – Simulating prestress of an elastica curve with Karamba. a) Pinned elastica curve. b) Released elastica curve. The measurements are given in meters

MOMENT DISTRIBUTION

For the first case with two pinned supports, it is possible to extract the moments from the second order analysis. The moment distribution resulting from the prestress load with and without an

additional point load of 10 kN at the middle is shown in Figure 2.2.5. The distribution from prestress without a point load deviates significantly compared to the result from Figure 2.1.7 (top) and is clearly wrong in this case since the curvature is not zero at the locations where the bending moments are zero. This behaviour is a result of the prestress being treated as an external load rather than an internal state in the rod. The distribution from prestress with an additional point load can be directly compared to the plot in Figure 2.1.7 (middle). It is observed that the prestress load results in a small reduction in bending moments and reduces the maximum deflection of the structure from 42 mm (see Subsection 2.1.3) to 33 mm at the top. The moment distribution is however far from the final result obtained from superposition (referring to Figure 2.1.7 (bottom)).

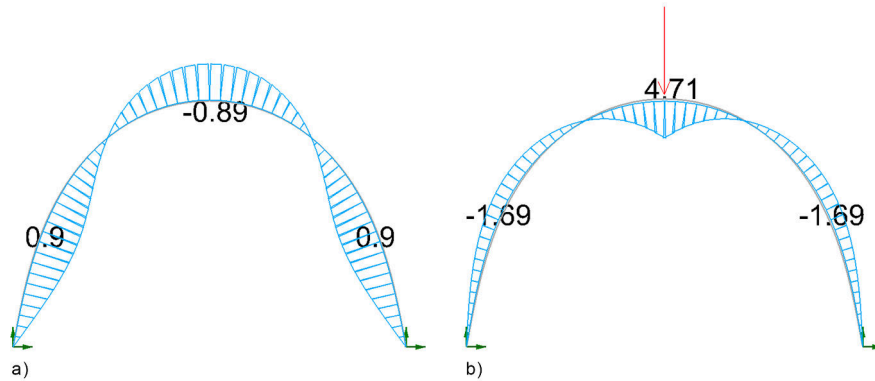


Figure 2.2.5 – Moment plot of a pinned elastica curve with a) prestress and b) prestress and point load using Karamba. The measurements are given in kNm

2.2.4 Remarks

The finite element model was further advanced for this project, in an attempt to include the prestress effect by calculating equivalent bending moment force couples, which were subsequently applied to the structure as an external load. The structural analysis by Ramboll Computational Design did not include any information about the forces in the structure, which suggests that this aspect was less important. Since the strength of both GFRP and carbon fibre is relatively high, the bending stresses in the structure would most likely not be the problem but for a bending-active timber gridshell like the OnGreening Pavilion this aspect could not have been ignored.

Instead, the analysis approach focused on the effect of prestress on the deformation behaviour. Even though the deformation of the elastica curve from the applied prestress load seems to comply with the expected behaviour (referring to Figure 2.2.4), it is questionable whether the behaviour is correct when additional loads are applied. In this case, the prestress effect is responsible for reducing the maximum deflection by 9 mm at the top but the reliability of this behaviour is questioned by the incorrect moment distribution. In general, the modelling of prestress as an external load is confusing as prestress is more intuitively associated with the reaction of a bent rod rather than the action of a load.

In an conversation with Mathias Gmachl, founder of Loop.pH, the author was informed that there was a contradiction with regard to the placement of additional reinforcement between the results from the finite element analysis and the intuition gained by the artist through years of experiments with these materials. The final reinforcement pattern was therefore adapted on site to find a compromise. The uncertainty related to the modelling of the prestress effect on the overall deformation of the structure combined with the slippage issue may explain why that was the case.

2.3 Hybrid Tower



Figure 2.3.1 – Interior of the hybrid Tower at CITA (Thomsen et al., 2015)

Hybrid Tower was part of the “Complex Modelling” research project, which was mainly a collaboration between CITA (Centre for IT and Architecture) and KET (Department for Structural Design and Technology) together with other universities and companies involved with textile design. The 6.9 m tall tower was built in 2015 in the courtyard of the Design Museum Denmark and combined individually soft elements including bending-active GFRP rods, textile membranes and steel wires into a stronger whole. The aim was to develop new modelling pipelines with integrated material properties to explore the design space in between form-finding and the more rigorous finite element analysis and link this to fabrication as well (Thomsen et al., 2015).

2.3.1 Hybrid concept

The overall shape of the tower was defined by overlapping bending-active GFRP rods, which were stacked on top of each other. The in-between areas were filled with a knitted textile membrane made of polyester yarn, which had the purpose of bracing the rod system (similar to the principle behind a tent structure). To better withstand the wind loads, a wheel system of tension cables were added to the central axis of the tower thus increasing the lateral stiffness. The cable system also pulled the membrane inwards, which had the benefit of increasing the double curvature and introducing more prestress to the membrane to make it more efficient. The hybrid behaviour was characterised by the interaction between these three different systems and quantified by adding each system one by one and observing a big increase in stiffness with only a little increase in mass. The system is shown in Figure 2.3.1.

2.3.2 Design analysis workflow

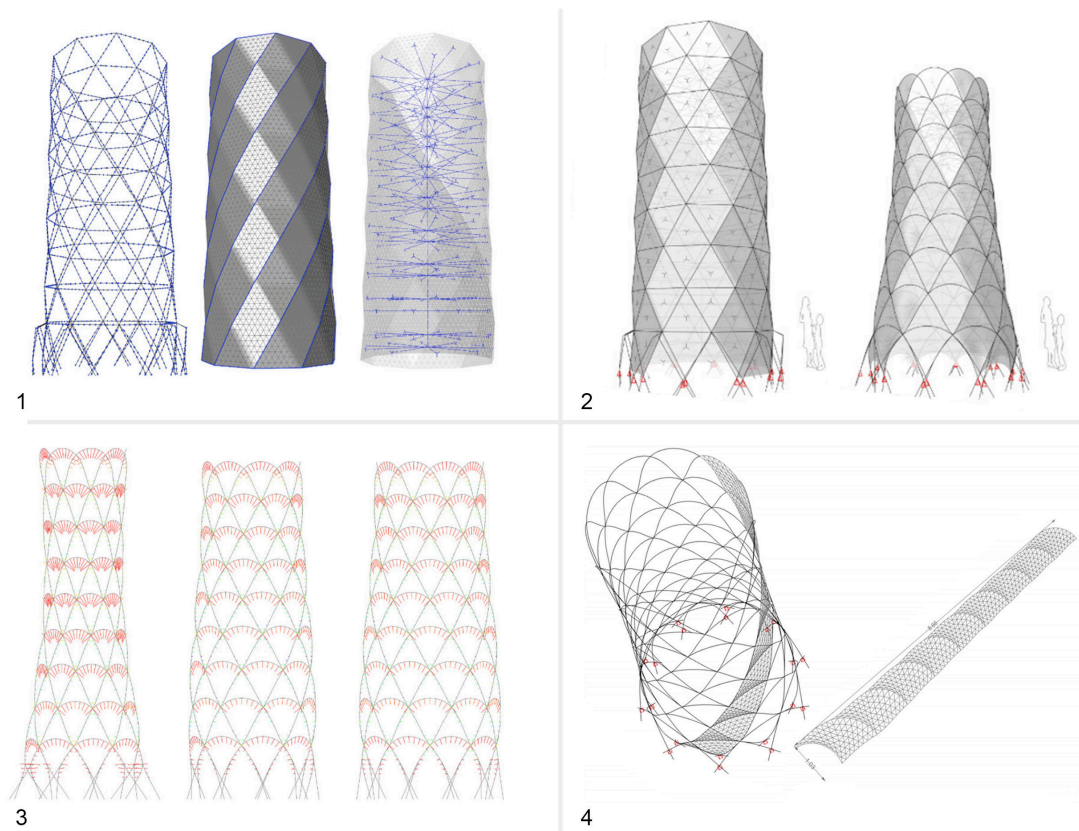


Figure 2.3.2 – Hybrid Tower modelling workflow (Thomsen et al., 2015)

A modelling workflow, which integrated both geometry, form-finding, analysis and fabrication was developed to design the tower. The Rhino/Grasshopper environment was used in a combination with the physics constraint solver Kangaroo2 (Piker, 2015b). The workflow was divided

into four steps as illustrated in Figure 2.3.2:

1. Modelling of the coarse geometry of the three structural systems; polylines representing the bending-active GFRP rods, a mesh representing the membrane and lines representing the cables.
2. Form-finding using Kangaroo2 to arrive at the more accurate tower geometry
3. Analysis and visualisation of the bending radii of the rod elements to evaluate if it was possible to built the form-found shape
4. Unrolling the membrane to the plane in order to directly create knitting patterns and evaluate maximum dimensions related to the fabrication equipment

This modelling workflow allowed multiple configurations to be explored and enabled a more informed process to arrive at the final design.

2.3.3 Structural analysis

Whilst the integrated bending radii evaluation helped to obtain a realistic idea about the utilisation of the rods, further and more detailed analysis using the finite element software SOFiSTiK was necessary because the Kangaroo2 simulation did not mimic beam torsion and membrane behaviour accurately. Due to the large amount of bending-active elements and their complex spatial configuration, it was too slow to form-find the shape of these elements by the typical method in finite elements analysis where the rod ends are pulled to their target positions by means of a pre-tensioned cable. As a result, it was decided to use the form-found shape of the rods from Kangaroo2 and superimpose the prestress from initial bending to the results obtained from the finite element analysis. The shape of the membrane and cables were form-found with SOFiSTiK by adding prestress to the cables. The tower was subjected to wind loads and the stresses and deflections were evaluated without any feedback loop to the previously described modelling workflow.

2.3.4 Improvements

The author has through a collaboration with Anders Holden Deleuran (CITA) and Gregory Quinn (KET) gained further knowledge about the improvements since the tower project. The modelling of the coarse geometry followed by the form-finding process has been integrated into an even more smooth workflow between Rhino/Grasshopper and Kangaroo2. Furthermore, the bending radii analysis has been extended to include the calculation of bending moments using real material and cross section properties. In general, more experiments with real stiffness values as input for existing Kangaroo2 components have been further explored to simulate accurate structural behaviour and remarkable stability has been observed. However, biaxial bending and torsion of beams and general shell behaviour have still not been resolved.

2.3.5 Remarks

The Hybrid Tower demonstrates the usefulness of integrating structural and fabrication constraints into one modelling workflow. The unique about this project is that both form-finding, bending radii evaluation and unrolling of the membrane are performed with the same plug-in (Kangaroo2), which inherently deals with the large deformations related to this kind of form-active structure. Common for the modelling workflow developed as part of the Hybrid Tower project and the improvements afterwards is that native Kangaroo2 components are used with real material properties as input to simulate accurate structural behaviour. The forces are then back-calculated based on the updated particle positions. However, this is not a very efficient way of doing it as it essentially calculates everything twice and also requires more attention towards the data structure in order to map the calculated information back to the initial geometry. As the Kangaroo2 plug-in is relatively new (released spring 2015) and the developments with regard to structural properties are in an early stage, there is still plenty of opportunity to investigate further improvements of this workflow and thereby avoid the necessity to rely upon a separate structural analysis software package in the early stage of a design.

2.4 Research direction

This literature and software review highlights the challenges of designing and analysing form-active structures due to the large deformations. The advantage of integrating design and analysis in one modelling environment is that multiple configurations can be explored in a short time and informed decisions can be made to enrich the design as a whole. For form-active structures, this integration has the additional advantage of preserving information between the form-finding and analysis stage, which is crucial due to the inherent dependency.

The OnGreening Pavilion and Atmeture project demonstrated that the integrated finite element software Karamba had limited capabilities when trying to analyse these kind of structures. The two projects tried to include the prestress effect from initial bending of the elements with a focus on stresses and deflections respectively but neither of them succeeded in including the effect on both.

Even though the Hybrid Tower project eventually had to interrupt the workflow by exporting the model to a more advanced finite element software, it showed great potential in integrating design and analysis in one environment with Kangaroo2.

To achieve the overall aim of improving the design and analysis workflow of form-active structures within the Rhino/Grasshopper environment, the following actions are identified:

- Development of a digital tool that extends the Kangaroo2 plug-in to accurately model structural behaviour
- Validation of the structural response through comparisons with analytical solutions or results from other finite element software

Chapter 2: Literature & software review

- Focus on the visualisation of the analysis results to make the structural behaviour more intuitive to understand

In order to make this research compatible with the given time frame, the author has decided to focus on bending-active structures in particular.

Chapter 3

Kangaroo2

Kangaroo is a plug-in for Grasshopper developed by Daniel Piker in 2010, which simulates physical behaviour at interactive speed using dynamic relaxation techniques (Piker, 2016b). It has particularly proven useful for the form-finding of structures, which are built from flexible materials and involve large deformations.

The first version of Kangaroo used a force-based method, which calculated the resultant force on each particle in the system and used Newton's 2. Law of motion to derive the acceleration, velocity and lastly the new position of the particle within a specified time step. Viscous or kinetic damping was used to reduce the oscillations of the particles and thereby arrive faster at the static equilibrium solution (Williams et al., 2014). The author has previously implemented dynamic relaxation from scratch in the Processing environment based on a similar force-based approach (Brandt-Olsen, 2014).

The problem with this force-based method is its stability problems when the stiffness values become high, which often results in an “explosion” of the geometry. This happens because the force vector overshoots its target position by a large amount, which in turn results in an even larger “reaction” in the opposite direction. The behaviour is directly related to the fact that the magnitude of the force vector is dependent on both how far the particle is from its target position and also how important this force is relative to other forces. The situation is illustrated in Figure 3.0.1. Here the particle in the middle is attracted to each of the corner points and the magnitude of the force vector is proportional to the distance between the particle and the attractor point. The figure to the left shows the position of the particle if no additional weighting is introduced and the figure to the right shows the position when the weighting of one of the force vectors is doubled. It is clear that as the weighting increases, the new position of the particle passes beyond the target point (attractor point with highest weighting).

To limit this unstable behaviour, the time step can be reduced or higher-order integration schemes such as Runge-Kutta (Weinsstein, 2016) can be used but both at the cost of computational speed.

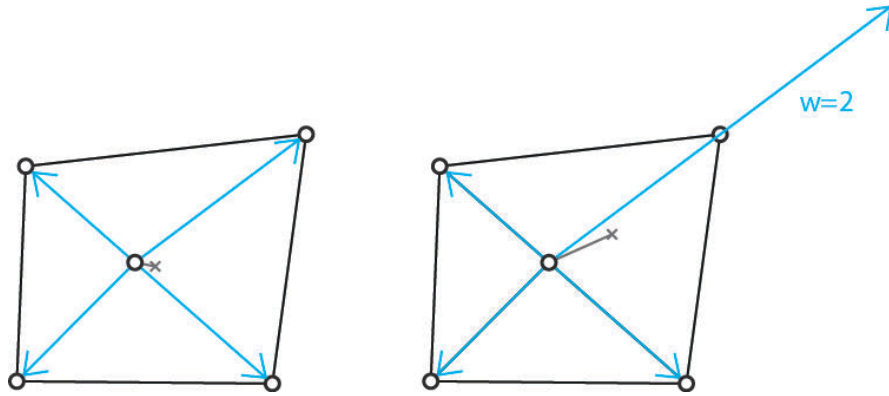


Figure 3.0.1 – Force-based approach. It is clear how the scaling of one force vector (to increase its relative importance) easily makes the particle overshoot its target position

Up until now, Kangaroo has therefore mainly been used for form-finding purposes with arbitrary stiffness values to speed up the process and avoid stability problems. This has led to a general misunderstanding that Kangaroo is a particle-spring system, which only simulates plausible physical behaviour. This is however not the case: Kangaroo is perfectly capable of producing accurate and meaningful structural results given the correct stiffness values.

In 2014, Daniel Piker presented a complete re-write of the solver, which specifically aimed to solve this stability issue and allowed users to script their own physical behaviours through an API. This new version is also known as Kangaroo2 and has been inspired by position-based dynamics (Piker, 2015c).

Rather than trying to reinvent the wheel, this platform is used as the basis for the software development as part of this thesis. At the time of writing, Kangaroo2 is a 3 DOF system meaning that it operates on 3d points rather than solids (6 DOF system). However, this is not considered to be a limitation for the scope of this project.

3.1 Goals

The key concept behind Kangaroo2 is that every constraint is expressed as a *goal*. A goal is a function (set of rules) that specifies where a subset of the particles want to move to and how strongly they want to get there. This general definition makes it possible to combine both geometrical constraints and structural behaviour into one system, which is a quite unique property. In contrast to the force-based method, a goal is separated into a so-called *move vector* and a *weighting*. The move vector specifies the relative movement from the particle's current position to its target position and the weighting indicates the strength of that movement. The different goals acting on each particle are subsequently combined in a weighted-average manner (Piker, 2015c) such that the new position is calculated from

$$P_{i,new} = P_{i,cur} + \frac{\sum_{j=1}^n \omega_j \cdot G_j}{\sum_{j=1}^n \omega_j} \quad (3.1.1)$$

Here i refers to the particle index, n is the number of goals acting on that particle, ω is the weighting and G is the move vector for goal j .

This ensures that overshooting never occurs since the new position always lies in the convex hull of the target positions. If the weighting of one goal is set to an infinite large number, the particle will move to the exact position of the target point, which is the desired behaviour. This is illustrated in Figure 3.1.1, which can be directly compared to the behaviour shown in Figure 3.0.1.

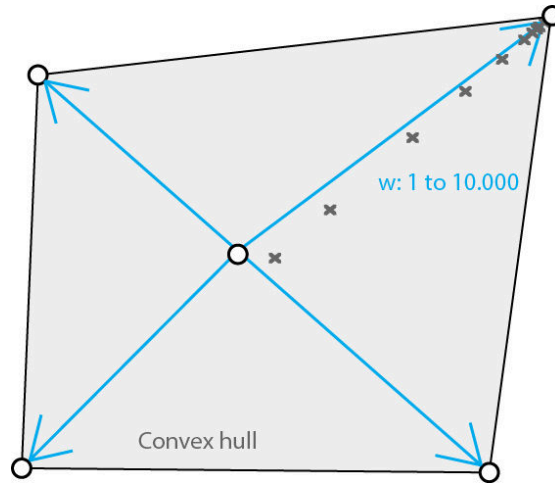


Figure 3.1.1 – Position-based approach. If the weighting of one goal becomes infinitely big, the particle will move to the exact target point rather than overshooting it.

3.1.1 Scripting custom goals

In April 2016, Daniel Piker made all the existing goals in Kangaroo2 open source (Piker, 2016a), which is a huge learning resource and thus encourages the development of new custom goals. The general code structure of such goal is shown in Listing 3.1. The idea is to create a new class (in this case *MyCustomGoal*), which inherits its properties and methods from another class called *GoalObject*. The constructor has to include some geometry, which can be referenced from Rhino/Grasshopper (the objects related to the goal). It is important to define the goal for the minimum subset of geometrical objects as the rest will be combined by the solver.

The properties in the constructor include three arrays: *PPos*, *Move* and *Weighting*. *PPos* contains an array of points that the goal acts on (derived from the geometry input) and the length of this array determines the length of the two other arrays. *Move* is an array of vectors

Listing 3.1 – K2 custom goal structure

```

1 public class MyCustomGoal : GoalObject
2 {
3     public MyCustomGoal(Point3d P, double k)
4     {
5         PPos = new Point3d[1]{P};
6         Move = new Vector3d[1];
7         Weighting = new double[1]{k};
8     }
9
10    public override void Calculate(List<KangarooSolver.Particle> p)
11    {
12        Point3d ThisPt = p[PIndex[0]].Position;
13        .
14        .
15        .
16        Move[0] = XX;
17    }
18
19    public override object Output(List<KangarooSolver.Particle> p)
20    {
21        var Data = new object[1]{Move[0] * Weighting[0]};
22        return Data;
23    }
24 }

```

(one for each point) and is often only initialised in the constructor, as it is unknown at this stage where the points want to move to. Weighting is an array of numbers that describes how strongly the goal affects each point. These values are often well-defined at this stage and can therefore already be assigned to the array in the constructor.

The custom goal class contains two methods my default: *Calculate()* and *Output()*. Inheriting these methods ensures that the goals are consistently defined and is compatible with the solver but they have to be overwritten as they are uniquely defined for each goal. It is mandatory to overwrite the Calculate method whereas the Output method is optional. Both methods take a list of particles as input (happens automatically behind the scenes) and this part makes it possible to calculate the move vectors based on the particles current positions as the solver iterates. The first line of code in the Calculate method shows how to access the current position of the first particle that was assigned to the PPos array (index 0). The behaviour of a goal is often related to its rest/initial state and it is therefore necessary to store this information as global class variables in order to access it in the Calculate method. Eventually, it is possible to assign vector(s) to the Move array.

The Output method makes it possible to output some data related to each goal. Most of the existing goals in Kangaroo2 do not have any outputs but there are a few exceptions including the “Length(Line)” and the “Planarize” goal, which output the line geometry and the twist amount of a mesh face. This data can be retrieved from the output of the solver component. The return value of this method is of type *object* and this ensures that a number of different types e.g. *Point3d*, *Vector3d* and *double* can be stored in one array but necessitates casting of the results afterwards. This Output method is crucial for the software development as part of this thesis.

3.1.2 Example: perpendicularity constraint

This example demonstrates the principles outlined in Subsection 3.1.1. The aim is to script a custom goal, which tries to make two line segments perpendicular to each other in 2D. The geometrical behaviour of such goal is sketched in Figure 3.1.2. The idea is that the first line (Ln_A) remains unchanged while the second line (Ln_B) rotates with a certain amount. The rotation is invoked by two equal and opposite move vectors. The magnitude can be determined from simple vector calculus:

1. Determine the perpendicular direction to the first line segment ($Ln_{A,perp}$) by taking the cross product of the first line vector (Ln_A) and a unit Z vector
2. Project the second line vector (Ln_B) onto the perpendicular direction via the vector dot product
3. Scale the perpendicular vector ($Ln_{A,perp}$) according to this projection and subtract the second line vector (Ln_B) from it in order to find the desired movement (d) in one end of the second line
4. Divide this movement by two and apply an equal and opposite move vector (rot) to the other end of the second line (Ln_B)

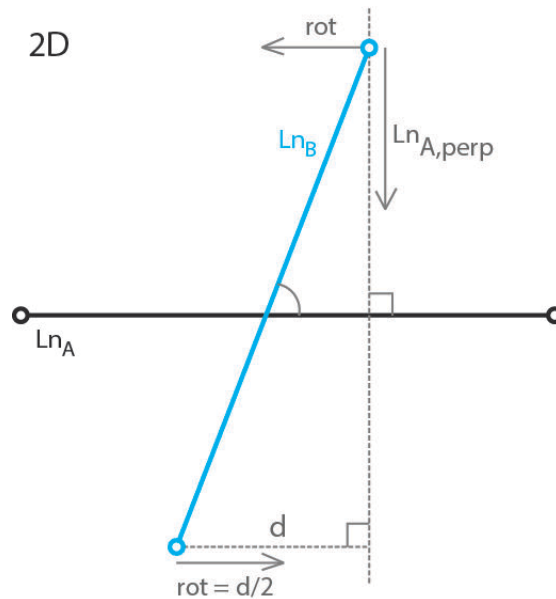


Figure 3.1.2 – Perpendicular goal behaviour

The described method is translated into a new custom Kangaroo2 goal as shown in Listing 3.2. The goal requires two lines and a strength input and outputs the current angle between the two lines during the iteration process.

Listing 3.2 – K2 custom perpendicularity goal

```

1 public class perpGoal : GoalObject
2 {
3     double angle;
4
5     public perpGoal(Line lnA, Line lnB, double strength)
6     {
7         PPos = new Point3d[4]{lnA.From, lnA.To, lnB.From, lnB.To};
8         Move = new Vector3d[4];
9         Weighting = new double[4]{strength, strength, strength, strength};
10    }
11
12    public override void Calculate(List<KangarooSolver.Particle> p)
13    {
14        //Current position of particles
15        Point3d lnA_Start = p[PIndex[0]].Position;
16        Point3d lnA_End = p[PIndex[1]].Position;
17        Point3d lnB_Start = p[PIndex[2]].Position;
18        Point3d lnB_End = p[PIndex[3]].Position;
19
20        //Lines as vectors
21        Vector3d vecA = lnA_End - lnA_Start;
22        Vector3d vecB = lnB_End - lnB_Start;
23
24        //Current angle between lines (degrees)
25        angle = Vector3d.VectorAngle(vecA, vecB) * (180.0 / Math.PI);
26
27        //Create perpendicular vector to lineA
28        Vector3d perpA = Vector3d.CrossProduct(vecA, new Vector3d(0, 0, 1));
29        perpA.Unitize();
30
31        //Calculate projection of lineB onto perpendicular direction to lineA
32        double proj = Vector3d.Multiply(vecB, perpA);
33        Vector3d perpA_scale = proj * perpA;
34
35        //Calculate rotation vector
36        Vector3d rot = perpA_scale - vecB;
37        rot /= 2.0;
38
39        //Define move vectors
40        Move[0] = new Vector3d(0, 0, 0);
41        Move[1] = new Vector3d(0, 0, 0);
42        Move[2] = -rot;
43        Move[3] = rot;
44    }
45
46    public override object Output(List<KangarooSolver.Particle> p)
47    {
48        var Data = new object[1]{angle};
49        return Data;
50    }
51 }

```


The custom goal is tested in a simple set-up as shown in Figure 3.1.3. The two grey lines are given as input to the perpendicular goal and a strength value is specified. The move vectors from the goal are visualised and the blue line is the result from the simulation. The angle between the two line segments is output from the solver and is exactly 90 degrees.

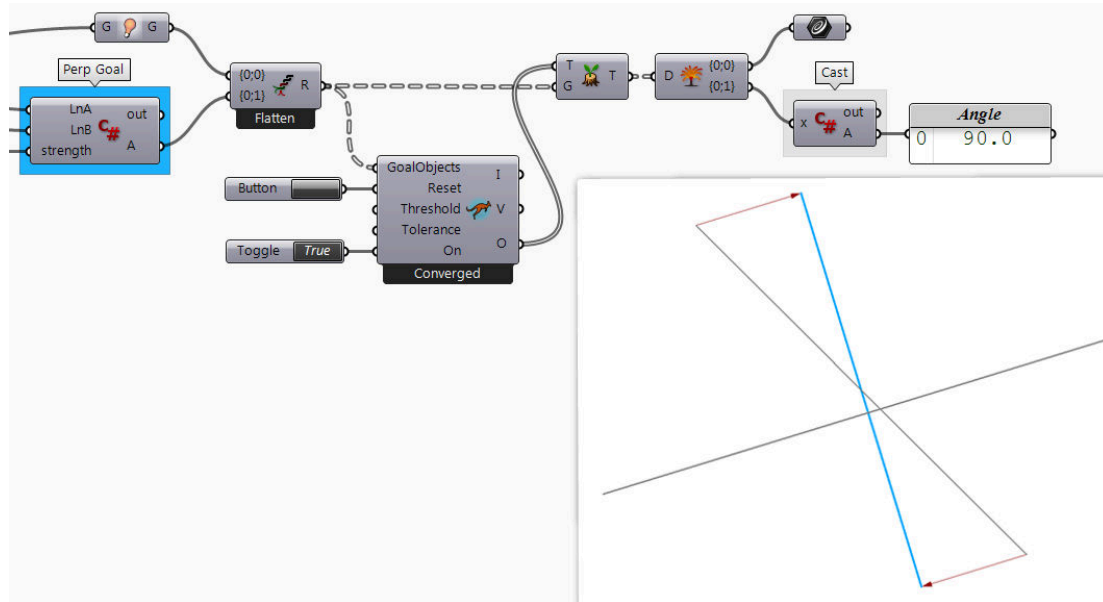


Figure 3.1.3 – Result from perpendicular goal

The goal is useful e.g in the context of Thrust Network Analysis. The principle behind this form-finding method is to generate two reciprocal graphs: a form and a force diagram. The form diagram represents the projection of mesh (shell) to the XY plane and the topology of the force diagram can be created from the dual graph of the form diagram. The form diagram only represents a shell in pure compression if the dual graph fulfils a perpendicularity constraint such that the forces only act along the lines in the form diagram. The polygons in the force diagram then describe the horizontal equilibrium in each vertex. The scale of the force diagram can be chosen arbitrarily. The reader is referred to Block (2009) for further details of this form-finding technique. Figure 3.1.4 (left) shows a form diagram (black) and its dual graph (blue). It is clear that the corresponding edges are not perpendicular to each other. The custom perpendicular goal is defined for all edge pairs and a “Length” goal for each line in the dual graph is added to ensure that the lines do not shrink to an infinitesimal size during the iteration process. The move vectors from the goals are visualised and highlight how multiple conflicting goals can exist in each point. Figure 3.1.4 (right) shows the result from the simulation and the current angles between the edge pairs. It is observed that the angles are only close to 90 degrees, which means that it is not possible to satisfy all goals simultaneously in this case. However, it demonstrates how the solver automatically combines the goals and finds a solution that is as close as possible to the desired behaviour (see Subsection 3.3).

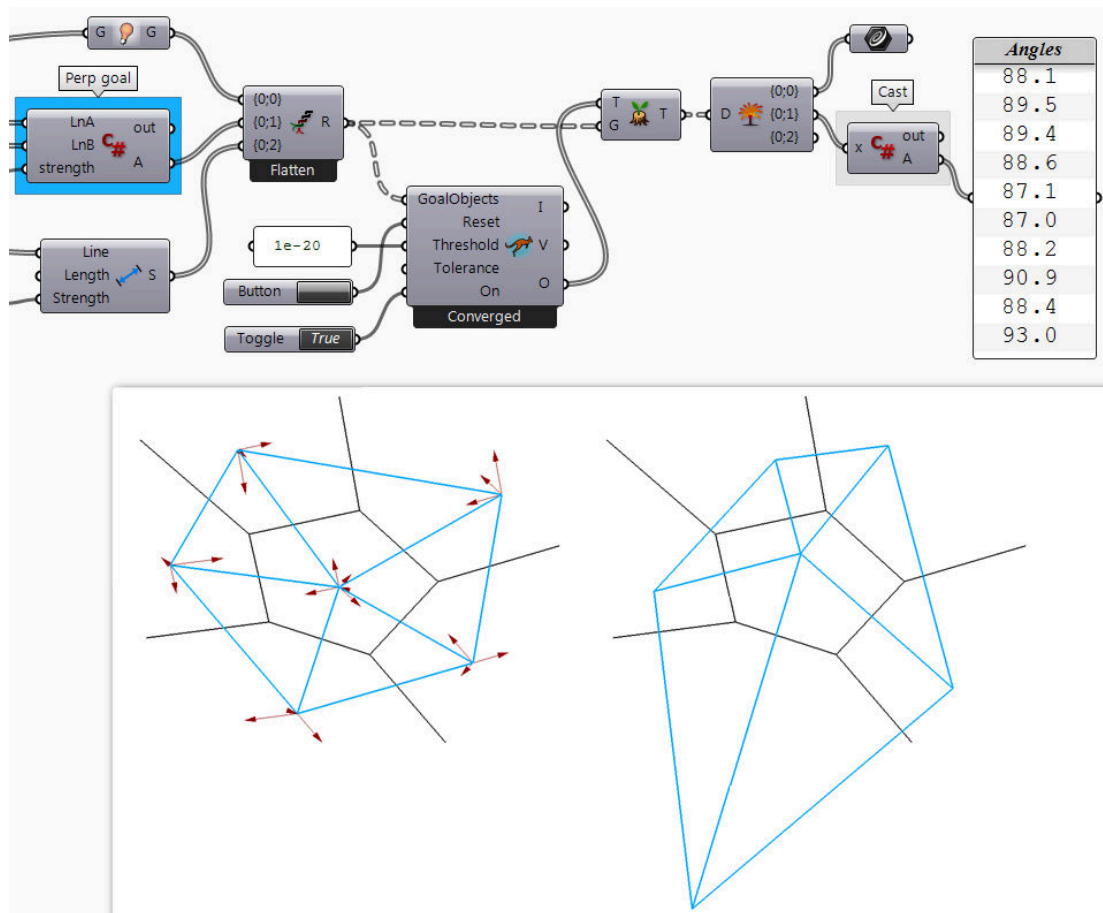


Figure 3.1.4 – Perpendicular goal for Thrust Network Analysis. Left: Form diagram (black) and its dual graph (blue). Right: Form diagram (black) and its force diagram (blue). In this case it is not possible to find a dual graph, which fulfils the perpendicularity constraint entirely.

3.2 Workflow

The workflow in Grasshopper is usually characterised by a directed acyclic graph (DAG). This means that if the input and output values are visualised as vertices and the functions (components) as edges then the connectivity forms a graph where no directed path loops back to itself (Wikipedia, 2016). In other words, input becomes output which becomes input again etc. such that a Grasshopper definition is defined from left to right.

Kangaroo2 breaks this work flow by sending data back and forth between the goals and the solver as shown in Figure 3.2.1 and this is an important observation to understand how to customise goals. When scripting a custom goal, this feature enables the definition of a certain behaviour based on information of the new positions of the particles even though this is only calculated in the solver component afterwards. In a normal DAG workflow, this information would therefore not be accessible at this point in the model history. Sending data back and forth between the goals and the solver automatically happens behind the scenes but it is an important detail to

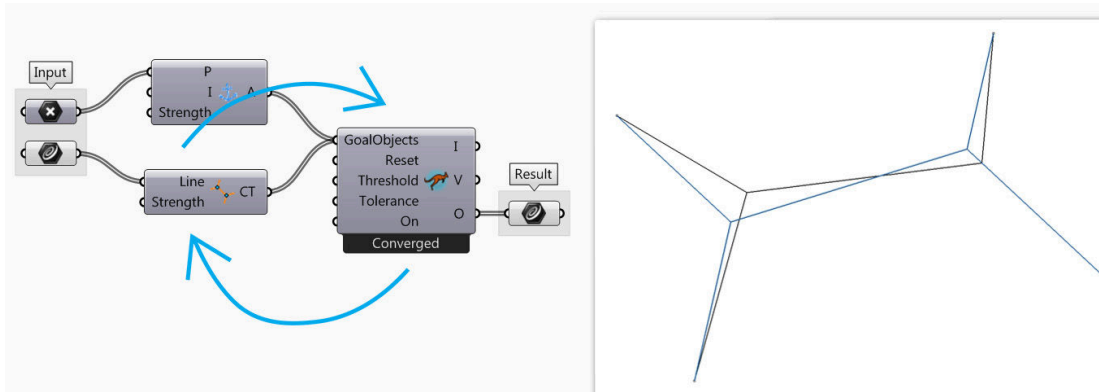


Figure 3.2.1 – Kangaroo2 workflow. In this case showing a simple example of a Steiner Tree using the Constant Tension goal

be aware of when customising Kangaroo2. A hint to this workflow is also noticeable from the code structure as it uses `List<KangarooSolver.Particle> p` as the argument for the Calculate and Output methods.

3.3 Solver

The solver is responsible for combining all the different types of goals and calculating the new positions of the particles according to Equation 3.1.1. These updated positions are passed back to the goals, which in turn calculate the new move vectors that are combined into new positions. This iterative process is repeated and by doing so the *total squared error of the distances between the particles current positions and their target positions is minimised* (Piker, 2015c). This minimisation is directly related to the energy of the system and this relation makes it possible to simulate accurate structural behaviour given the appropriate stiffness values as the displacements of a structure always follow the path that minimises the total potential energy. The convergence towards a static equilibrium solution is similarly related to the energy of the system and is detected when the total kinetic energy is smaller than a certain small threshold value (default is $1 \cdot 10^{-15}$).

By default, three different solver types are available in the Kangaroo2 plug-in: the standard solver with continuous output, which iterates until the system converges or the user disables it. A “Zombie Solver”, which only outputs the final result when the system converges or a specified number of iterations is reached. Lastly a “Bouncy Solver”, which preserves momentum during the simulation and thus better reflects real world physical behaviour but at the cost of computational speed. However, it is also possible to script custom solvers via the API, which is useful to realise certain work flows, extract other information from the system, which is not output as default values from the existing solver components or for debugging of custom goals.

Listing 3.3 shows the general code structure for scripting custom solvers. In this case, a Kangaroo2 system is initialised together with some general properties including the tolerance

Listing 3.3 – K2 solver custom iteration

```

1 KangarooSolver.PhysicalSystem PS = new KangarooSolver.PhysicalSystem();
2 List<IGoal> GoalList = new List<IGoal>();
3
4 double tolerance = 0.001;
5 double threshold = 1e-15;
6 int counter = 0;
7
8 foreach(IGoal G in k2Goals)
9 {
10     PS.AssignPIndex(G, tolerance);
11     GoalList.Add(G);
12 }
13
14 do
15 {
16     PS.Step(GoalList, false, threshold);
17     counter++;
18 } while(PS.GetvSum() > threshold && counter < 100);

```

(two particles are combined into one if the distance between them is smaller than this value), threshold (kinetic energy value which specifies when the system has converged) and a counter (to avoid an infinite loop if the system never reaches a kinetic energy value lower than the threshold). The next step is to create a new list of goals from the predefined goals (*k2Goals*), with the difference that each individual goal becomes aware of the particle indexes it acts on in relation to the entire Kangaroo2 system. A Step function is responsible for breaking the entire iteration process down into smaller parts (it can be compared to the time-step in the force-based approach) such that it iterates until either reaching 15 ms or the kinetic energy threshold. The do...while loop ensures that these smaller iteration steps continues until either reaching the kinetic energy threshold or a maximum number of counts.

3.4 Convergence speed

The procedure of using the weighted average to calculate the updated positions only converges slowly since the move vectors become smaller and smaller in each step as the system approaches equilibrium. The particles are therefore assigned with masses to give them inertia and thus carry them faster towards their target positions. In that regard, it becomes crucial to introduce damping to the system to avoid that the particles overshoot their equilibrium positions. A new damping scheme has been implemented in Kangaroo2, which differs from both viscous and kinetic damping.

The damping scheme is based on the projection of each particle's acceleration vector onto the velocity vector (during each iteration) to determine if they point in the same direction. If the projection value is positive, the acceleration and velocity vectors point in the same direction, which indicate that the particle is moving towards its target position and hence no damping is applied. If the projection value is zero or negative, the acceleration and velocity vectors point in the opposite (or perpendicular) direction, which indicate that the particle has moved beyond its target position and hence a strong damping factor is applied. The advantage of this damping scheme is that it avoids the velocity discontinuity from kinetic damping and is applied on an

individual basis such that only the movement of specific particles are slowed down rather than the entire system (Piker, 2015c).

The functionality of the Kangaroo2 solver has not been released as open source, which means that there are a few steps in the methodology that is still unclear to the author. For example, it is unknown how the damping is introduced. In a force based approach, the damping is applied to the velocity vectors, which are then used to calculate the updated positions. This is a natural workflow as the particle positions are calculated at the end. In the position based approach, the updated positions are directly calculated from the weighted average (see Equation 3.1.1), which means that the velocity and acceleration vectors become irrelevant. But since the damping scheme in Kangaroo2 is based on the velocity and acceleration vectors, these must be calculated somehow (unknown which time step is used) and turned back into positions that are affected by the damping. However, this lack of knowledge of the specific technical details is not a limitation for using the software.

Chapter 4

K2Engineering

This chapter describes the methods that have been implemented as part of the software development and are all related to a 3 DOF system. Several simple test cases are used throughout the process to validate the results and thereby gain confidence in the tool (see Appendix A). The software is developed as a separate plug-in to Grasshopper and can be considered as an extension to Kangaroo2 with calibrated structural values hence the name “K2Engineering”. The author has decided to make this plug-in open source under the Apache 2.0 licence (OSI, 2004) and has made all source code written in C# available on Github (Brandt-Olsen, 2016).

In addition to the new goals with calibrated structural behaviour, a big part of the development has also included the visualisation of forces in relation to the three dimensional geometry. The visualisations help to obtain an intuitive understanding of the structural behaviour and identify load paths without inspecting the specific values.

It is important to clarify that in most cases it is possible to simulate the calibrated structural behaviour using the native Kangaroo2 components. This is accomplished by inputting the appropriate axial and bending stiffness to the strength inputs. The contribution of this research is therefore mainly related to the output of meaningful structural values, which are not back-calculated from the displacements in the end and thus avoid duplicate functionality. In addition, this approach makes it more clear which properties are needed for the calibration and in general has to be considered as the early stage of a larger potential centred around this Kangaroo2 framework. Notice that at the time of writing these custom goals, the native Kangaroo2 goals had not yet been made open source.

4.1 Supports

In any building system it is necessary to have supports to prevent the structure from moving around in space. With the current Kangaroo2 version it is only possible to control the translational degrees of freedom i.e. restrain the movement in the X, Y and Z directions. A pinned

support implies that all three directions are fixed, whilst a roller support is able to move in at least one direction. Even though it is a 3 DoF system, it is still possible to mimic a rigid support by adding two consecutive pinned supports along a line, which has the effect of creating a reaction force couple to fix the rotation.

A support can be considered as a point that is attached to a particle of infinite mass with a zero length spring (Piker, 2015a). This means that Hooke's Law can be used to calculate the reaction force:

$$F = -k \cdot x \quad (4.1.1)$$

Here k specifies the stiffness of the spring (any arbitrary large number) and x is the extension, which in this case corresponds to the movement of the point due to the applied loads on the structure.

4.1.1 Implementation

This behaviour is implemented as a support goal by specifying the position of the support, the strength of the spring and which directions are fixed (see Table 4.1.1).

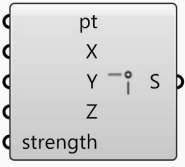
| GH Component | Input | Output |
|---|---|---------------------------------|
|  | Position X fixity Y fixity Z fixity Spring strength | Position Reaction force [kN] |

Table 4.1.1 – Input and output from the support goal

The functionality is translated into code as shown in Listing 4.1 and briefly described in the following:

Line 3-6: Four variables are declared as global class properties in order to make them accessible in the Calculate method. They include the original support point (which will function as a target when the point starts to move) and three boolean values specifying in which directions (if any) the point is allowed to move.

Line 10-17: The four class properties are initialised based on the input to the component. The goal only acts on one point, which means that only one value has to be specified for each of the mandatory arrays. PPos contains the specified support point, the Move array is only instantiated as it is unknown at this point where the particle wants to move to and the weighting corresponds to the specified strength value.

Line 22-37: The current position of the particle during the simulation is stored as a variable. The move vector is easily calculated since the extension for a zero-length spring exactly corresponds

Listing 4.1 – K2Engineering support goal

```

1  public class SupportGoal : GoalObject
2  {
3      Point3d Target;
4      bool xFixed;
5      bool yFixed;
6      bool zFixed;
7
8      public SupportGoal(Point3d Pt, bool x, bool y, bool z, double k)
9      {
10         PPos = new Point3d[1] { Pt };
11         Move = new Vector3d[1];
12         Weighting = new double[1] { k };
13
14         Target = Pt;
15         xFixed = x;
16         yFixed = y;
17         zFixed = z;
18     }
19
20     public override void Calculate(List<KangarooSolver.Particle> p)
21     {
22         Point3d currentPt = p[PIndex[0]].Position;
23         Vector3d moveTotal = Target - currentPt;
24         if (!xFixed)
25         {
26             moveTotal.X = 0.0;
27         }
28         if (!yFixed)
29         {
30             moveTotal.Y = 0.0;
31         }
32         if (!zFixed)
33         {
34             moveTotal.Z = 0.0;
35         }
36         Move[0] = moveTotal;
37     }
38
39     public override object Output(List<KangarooSolver.Particle> p)
40     {
41         var Data = new object[2] { p[PIndex[0]].Position, Move[0] * Weighting[0] * 1e-3 };
42         return Data;
43     }
44 }
45

```

to the distance between the original (target) point and the current position. Due to the large weighting, this means that the point is moved back to the target position in each iteration. If the support is allowed to move in one direction, the move vector is manipulated such that the component in the released direction equals zero.

Line 42: The current position of the support point and the reaction force (calculated by multiplying the move vector with the spring stiffness) are output from the goal.

4.2 Bar

The simplest structural element is a bar, which only transfers forces through axial action. It is assumed to be made of a linear elastic material, which implies that the strain is proportional to stress (Hooke's Law). This relationship can be rewritten as:

$$\sigma = E \cdot \varepsilon \Leftrightarrow F = E \cdot A \cdot \frac{L - L_0}{L_0} \Leftrightarrow F = \frac{E \cdot A}{L_0} \cdot x \quad (4.2.1)$$

Where F is the axial force [N], E is the Young's modulus [MPa], A is the cross section area [mm^2], L_0 is the rest length of the bar element [m], L is the current length [m] and x is the extension [m]. The term $\frac{E \cdot A}{L_0}$ is also known as the axial stiffness. If the bar is stretched and therefore in tension, it will try to contract to its initial rest length and as a result have reaction forces pointing inwards from the ends. On the other hand, if the bar is shortened and therefore in compression, it will try to stretch out to its initial rest length and thus exert outwards reaction forces from the ends. The behaviour is illustrated in Figure 4.2.1.

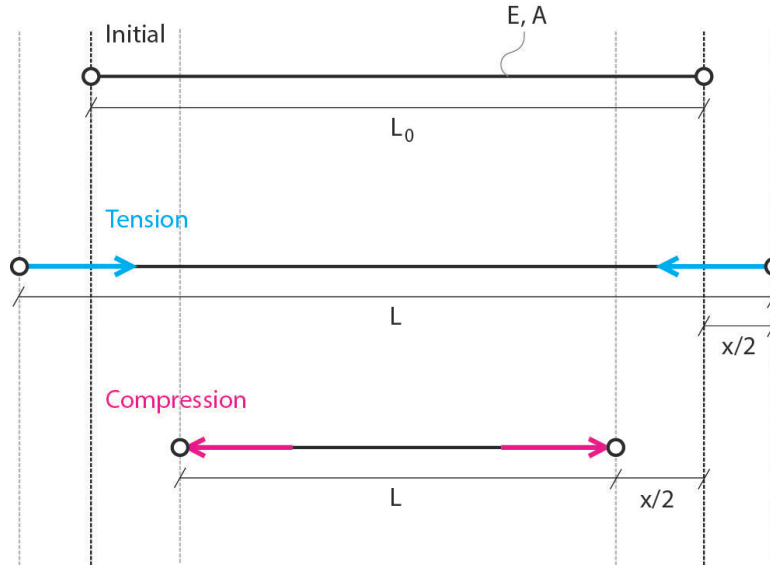


Figure 4.2.1 – Axial behaviour of a bar element

4.2.1 Implementation

To implement this as a Kangaroo2 goal, it is necessary to express the force F from Equation 4.2.1 as a “Move vector” and a “Weighting”. Since x is the total extension of the bar and the move vector only describes the movement of one end point, it has to be halved and thus equal $\frac{x}{2}$. To account for that, the axial stiffness has to be doubled such that the axial force F remains the same. As a result, the weighting equals twice the axial stiffness.

The smallest subset of geometrical objects to define a bar goal is one line segment. In addition to that, it is necessary to specify the Young's modulus and the cross section area in order to calculate the axial forces (the rest length is derived from the line segment automatically). With these inputs it is possible to script a bar goal with meaningful structural output (see Table 4.2.1).


| GH Component | Input | Output |
|---|--|---|
|  | Line [m] Young's modulus [MPa] Area [mm ²] | Particle index start Particle index end Updated line geometry Axial force [kN] Axial stress [MPa] |

Table 4.2.1 – The input and output from a bar goal

This functionality is translated into code as shown in Listing 4.2 and briefly described in the following:

Line 3-5: Three variables are stored as global class properties in addition to the obligatory PPos, Move and Weighting arrays. They include the rest length, the cross section area and a boolean for whether the member is in compression or not. Thereby they can be accessed in the Calculate method.

Line 13-15: The obligatory arrays are initialised. PPos contains the start and end point of the line, Move is only instantiated as the move vectors are unknown at this point and the Weighting is set to twice the axial stiffness as explained above.

Line 20-38: The current positions of the start and end point are stored and used to calculate the length of the line in each step of the simulation. This is in turn used to calculate the extension by subtracting the rest length from the current length. The sign of the extension implies whether the member is in tension or compression, which is captured by the boolean class property. The sign is also useful to consistently specify the directions of the move vectors such that they are automatically reversed when the axial force shifts between tension and compression.

Line 43-51: The desired output is extracted and added to a Data array. The axial force is calculated by multiplying the first item in the Weighting array with the first item in the Move array as this corresponds to Equation 4.2.1. This is where the boolean class property becomes useful as it helps to keep the sign of the axial force, which is otherwise lost by taking the length of the Move vector. The axial stress is subsequently determined by dividing the force with the cross section area. Furthermore, the particle index of the start and end point of the line and the updated line geometry are output.

Listing 4.2 – K2Engineering bar goal

```

1 public class BarGoal : GoalObject
2 {
3     double restLength;
4     bool isCompressionMember;
5     double area;
6
7     public BarGoal(Line L, double E, double A)
8     {
9         restLength = L.From.DistanceTo(L.To);
10        isCompressionMember = true;
11        area = A;
12
13        PPos = new Point3d[2] { L.From, L.To };
14        Move = new Vector3d[2];
15        Weighting = new double[2] { (2 * E * A) / restLength, (2 * E * A) / restLength };
16    }
17
18    public override void Calculate(List<KangarooSolver.Particle> p)
19    {
20        Point3d ptStart = p[PIndex[0]].Position;
21        Point3d ptEnd = p[PIndex[1]].Position;
22
23        Vector3d forceDir = new Vector3d(ptEnd - ptStart);
24        double currentLength = forceDir.Length;
25        forceDir.Unitize();
26
27        double extension = currentLength - restLength;
28        if (extension > 0.0)
29        {
30            isCompressionMember = false;
31        }
32        else if (extension < 0.0)
33        {
34            isCompressionMember = true;
35        }
36
37        Move[0] = forceDir * (extension / 2);
38        Move[1] = -forceDir * (extension / 2);
39    }
40
41    public override object Output(List<KangarooSolver.Particle> p)
42    {
43        double factor = 1.0;
44        if (isCompressionMember)
45        {
46            factor = -1.0;
47        }
48        double force = factor * Weighting[0] * Move[0].Length;
49
50        var Data = new object[5] { PIndex[0], PIndex[1], new Line(p[PIndex[0]].Position,
51            p[PIndex[1]].Position), force / 1000.0, force / area };
52        return Data;
53    }
54 }

```

4.3 Cable

A cable is a special case of the bar element, which only works in tension and becomes slack in compression. Furthermore, it has the option to be prestressed in order to reduce the deflections. A prestressed cable is already in tension before the external loads are applied and the strength of the prestress is typically specified as a tension force. Figure 4.2.1 shows that this tension state implies that the cable wants to shrink back to its rest length and it can thus be simulated by calculating a new rest length that is smaller than the initial length of the element. Rewriting Equation 4.2.1 gives

$$x = \frac{F \cdot L_0}{E \cdot A} \quad (4.3.1)$$

The rest length corresponding to a certain pretension force F is therefore $L_{rest} = L_0 - x$.

4.3.1 Implementation

Due to the many similarities with the bar goal, only the differences with regard to the implementation are mentioned in the following. Table 4.3.1 shows the cable Grasshopper component with one additional input parameter for the pretension value.

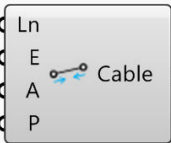
| GH Component | Input | Output |
|---|---|---|
|  | Line [m] Young's modulus [MPa] Area [mm2] Optional pretension [kN] | Particle index start Particle index end Updated line geometry Axial force [kN] Axial stress [MPa] |

Table 4.3.1 – The input and output from a cable goal

The few changes in the code as shown in Listing 4.3 include:

Line 3-4: It is no longer necessary to store a boolean value for whether the element is in tension or compression so this variable is removed.

Line 6: The pretension parameter F is added as an argument to the constructor method.

Line 15: A new rest length is calculated from Equation 4.3.1. The default pretension value is zero, which means that the rest length equals the initial length if no pretension value is specified.

Line 29-35: The move vectors are set to zero by default and are only changed if the element is in tension. This means that the cable does not exert any axial forces if it is in a neutral or compression state.

Listing 4.3 – K2Engineering cable goal

```

1 public class CableGoal : GoalObject
2 {
3     double restLenght;
4     double area;
5
6     public CableGoal(Line L, double E, double A, double F)
7     {
8         restLenght = L.From.DistanceTo(L.To);
9         area = A;
10
11         PPos = new Point3d[2] { L.From, L.To };
12         Move = new Vector3d[2];
13         Weighting = new double[2] { (2 * E * A) / restLenght, (2 * E * A) / restLenght };
14
15         restLenght -= (F * 1000 * restLenght) / (E * A);
16     }
17
18     public override void Calculate(List<KangarooSolver.Particle> p)
19     {
20         Point3d ptStart = p[PIndex[0]].Position;
21         Point3d ptEnd = p[PIndex[1]].Position;
22
23         Vector3d forceDir = new Vector3d(ptEnd - ptStart);
24         double currentLength = forceDir.Length;
25         forceDir.Unitize();
26
27         double extension = currentLength - restLenght;
28
29         Vector3d forceStart = new Vector3d(0, 0, 0);
30         Vector3d forceEnd = new Vector3d(0, 0, 0);
31         if (extension > 0.0)
32         {
33             forceStart = forceDir * (extension / 2);
34             forceEnd = -forceDir * (extension / 2);
35         }
36
37         Move[0] = forceStart;
38         Move[1] = forceEnd;
39     }
40
41     public override object Output(List<KangarooSolver.Particle> p)
42     {
43         double force = Weighting[0] * Move[0].Length;
44         var Data = new object[5] { PIndex[0], PIndex[1], new Line(p[PIndex[0]].Position,
45             p[PIndex[1]].Position), force / 1000.0, force / area };
46         return Data;
47     }
48 }

```

The cable goal is an example of a functionality, which is more difficult to simulate with a calibrated structural behaviour using the native Kangaroo2 goals. The new rest length can be calculated based on a pretension force (see Equation 4.3.1) and provided as input for the “Length(Line)” goal. However, if the length of the element during the simulation becomes smaller than this rest length, it will still exert axial forces even though it is in compression. To avoid this, one option is to set the rest length to zero such that any deformation will result in an elongated element and therefore always be in tension but that removes the “materiality” from the simulation and thus makes it more applicable as a form-finding goal.

4.4 Rod

A rod is an element, which resists the applied external load by bending action (out-of-plane forces). Typically, six degrees of freedom are necessary to describe this behaviour as the bending moments are related to the rotations. However, this is not possible in a 3 DoF system like Kangaroo2 and hence a different method to mimic the same behaviour is desired. Such method has been developed by Adriaenssens and Barnes (1999), which is based on the *continuity* of spline elements i.e. it requires two consecutive line segments to model the bending behaviour in one node as illustrated in Figure 4.4.1.

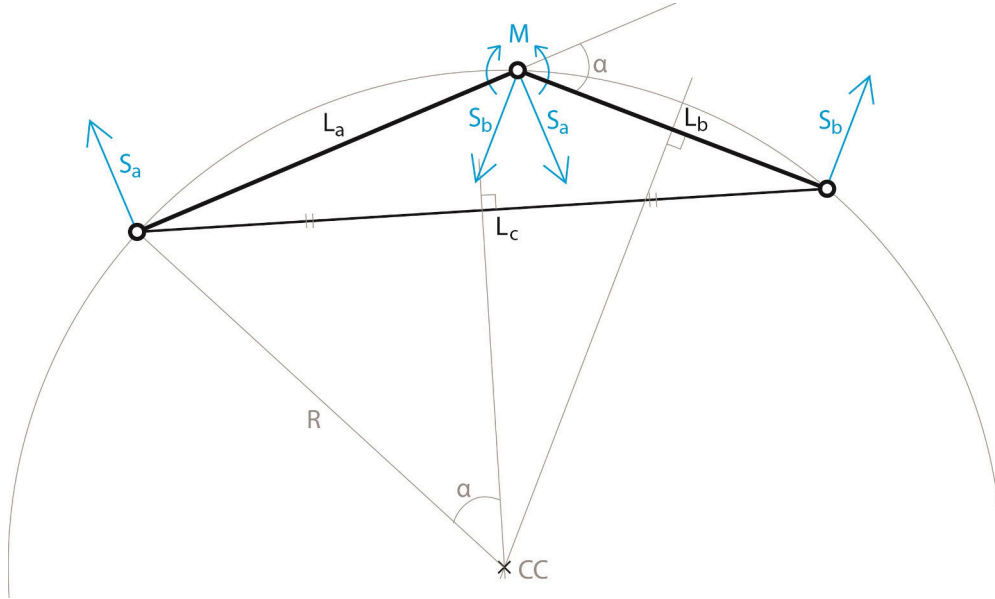


Figure 4.4.1 – Modelling of bending in a 3 DoF system

The method takes advantage of the relationship between the moment and curvature defined as

$$M = \frac{E \cdot I}{R} \quad (4.4.1)$$

Where R is the curvature radius and $E \cdot I$ is the bending stiffness. Given two consecutive line

segments, the radius of the circle passing through the end points is defined as

$$\sin(\alpha) = \frac{L_c/2}{R} \Leftrightarrow R = \frac{L_c}{2 \cdot \sin(\alpha)} \quad (4.4.2)$$

From that expression, the magnitude of the moment acting in the shared node is calculated. This moment is then translated into an equivalent force couple acting perpendicular to each line segment

$$M = S_a \cdot L_a = S_b \cdot L_b \quad (4.4.3)$$

These forces correspond to the shear forces in each element. The directions of the shear forces reflect how each element attempts to resist an applied load by rotation. By combining the equations above, the magnitude of the shear forces S_a and S_b are calculated as

$$\frac{E \cdot I}{\frac{L_c}{2 \cdot \sin(\alpha)}} = S_a \cdot L_a \Leftrightarrow S_a = \frac{2 \cdot E \cdot I \cdot \sin(\alpha)}{L_c \cdot L_a} \quad (4.4.4)$$

$$S_b = \frac{2 \cdot E \cdot I \cdot \sin(\alpha)}{L_c \cdot L_b} \quad (4.4.5)$$

4.4.1 Moments and shear forces

The method above describes how to calculate the bending moment and shear forces for two consecutive line segments. This subset of the spline is sufficient to calculate the moment in the shared vertex as shown in Figure 4.4.2 as the moment in one vertex is not affected by the moments resulting from the neighbouring elements.

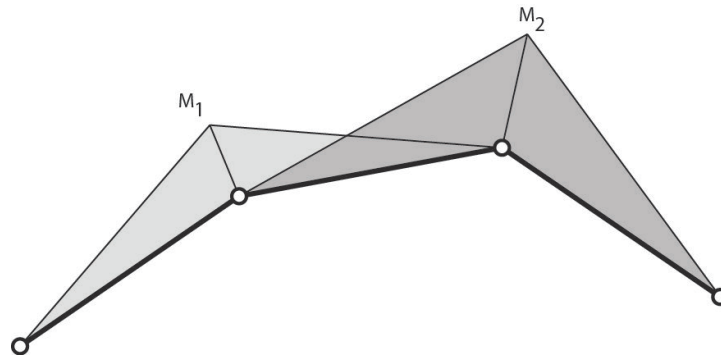


Figure 4.4.2 – Bending moments for a spline

However, this is not the case for the shear forces as illustrated in Figure 4.4.3. From three line segments of a spline it is possible to create two subsets of consecutive lines. Figure 4.4.3 (top) shows the shear forces for each subset according to the described theory. The curvature for the subset to the left is smaller than the curvature for the subset to the right and hence the

magnitude of the shear forces is smaller (as the lengths of the spline segments are constant). These shear forces represent the distribution as shown in Figure 4.4.3 (middle). It is observed that the line segment in the middle is influenced by two different shear values of opposite sign. By adding the values together for each line segment, the final shear distribution of the spline is obtained as seen from Figure 4.4.3 (bottom).

In other words, the shear value for each line segment is equal to the difference in moments at its end points. This is not surprising as shear is defined as the rate of change of the bending moments. However, it highlights an important point in this context; given two consecutive lines it is only possible to calculate the moment. The shear calculation requires information from the other rod goals as well.

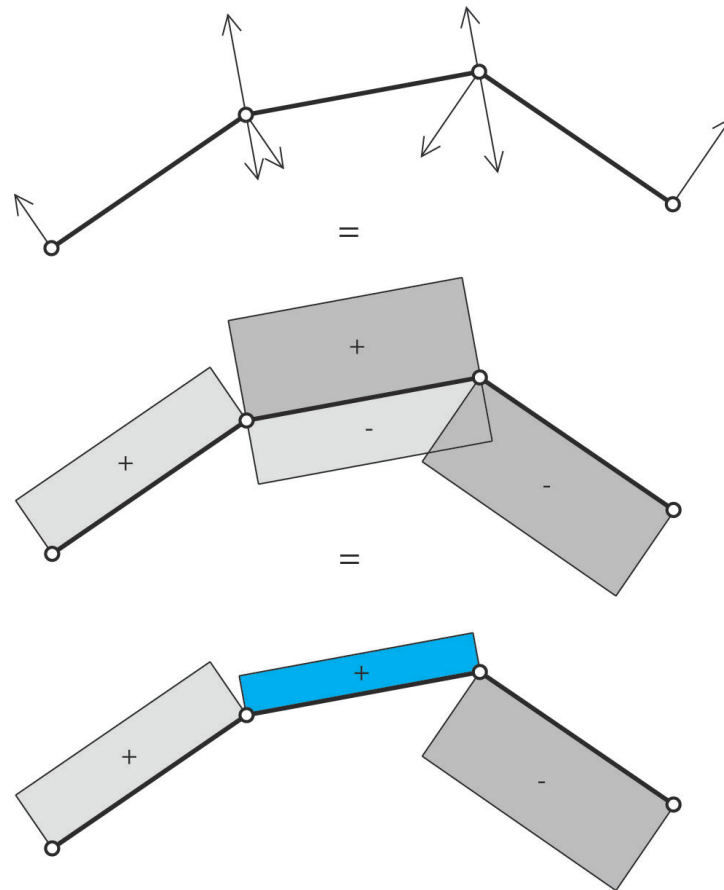


Figure 4.4.3 – Shear forces for a spline

4.4.2 Actively bent or fabricated curved elements

With this definition it is possible via the angle parameter α to specify the rest configuration of the spline i.e. whether it is unstressed in its initial spatial shape or as a straight line. If the spline is initially straight then $\alpha = 180^\circ - \angle L_a L_b$ as shown in Figure 4.4.1, which means that

there exists a moment from the beginning in the shared vertex based on the curvature of the circle through the three points. On the other hand, if the spline geometry is in rest in its initial configuration e.g. like a frame then $\alpha = \angle L_{a,r} L_{b,r} - \angle L_a L_b$ where the subscript r symbolises the angle from the rest state. This is equivalent to subtracting the moment from the deformed configuration with the moment from the rest state as described by Adriaenssens and Barnes (1999). As a result, the line segments seek to maintain the angle from the rest state rather than straighten out completely.

This is a key feature to define prestress in bent elements made from initially straight/planar materials and this way the prestress is modelled as an internal state in the element rather than an external load as described for the Atmeture project in Chapter 2. Furthermore, this means that the intended spatial shape can be defined beforehand and from there find the static equilibrium configuration as if it was initially straight. This is a big advantage compared to the methods used for large deformations in a finite element software where the *path* of the structure's deformation is of great importance for the final result.

4.4.3 Implementation

The smallest subset of geometrical objects to define a rod goal is two consecutive line segments. Furthermore, it is necessary to specify the Young's modulus, the moment of inertia, the distance to the fibre where the stress is calculated and whether the elements want to straighten out or are in rest in their current configuration. The goal outputs the particle index of the shared vertex, the bending plane, the moment and bending stress. The developed Grasshopper component with its input and output values are shown in Table 4.4.1.

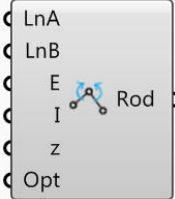
| GH Component | Input | Output |
|---|---|--|
|  | LineA [m] LineB [m] Young's modulus [MPa] Moment of inertia [mm ⁴] Distance to fibre [mm] Rest angle option (0: <i>straight</i> , 1: <i>current</i>) | Shared particle index Bending plane Moment [kNm] Bending stress [MPa] |

Table 4.4.1 – The input and output from a rod goal

To implement this functionality with Kangaroo2, the shear force from Equation 4.4.5 has to be separated into a move vector and a weighting. The “geometrical” part of the expression is defined as the move vector and the “material” part i.e. the bending stiffness is defined as the weighting. The code as shown in Listing 4.4 is briefly explained in the following:

Line 3: Three variables including the moment of inertia, distance to the outer fibre and the rest angle are defined as global class properties in addition to the mandatory arrays.

Line 7-9: The mandatory Kangaroo2 arrays are initialised and it has been decided that the rod goal acts on four points (end points of both line segments) to make the code more transparent.

This means that point two and three are identical but the Kangaroo2 solver will take care of that when the physical system is created.

Line 13-18: The rest angle from the initial configuration is defined based on the two different options.

Line 23-31: The current positions of the particles in each step of the iterations are stored and used to define “edge vectors”. These vectors are useful to calculate the current angle between the two line segments and thereby define the angle parameter α but also to obtain the normal vector to the plan spanned by the two edge vectors via the cross product.

Line 33-45: The normal vector and edge vectors are subsequently used to calculate the perpendicular directions (shear vectors) to the line segments again via the cross product. The magnitude of the shear vectors corresponds to the geometrical part of Equation 4.4.5. Eventually the move vectors are defined with special attention to the sign of the shear vectors.

Line 50: The dots symbolise the repetition of lines 23-29 + 31.

Line 51-60: The desired output is calculated, which include the shared particle index, the moment, the bending plane and stress. The moment is calculated from the first item in the move vector array multiplied with the first item in the weighting array and the length of the first segment. The bending plane is defined from the normal vector and the average of both move vectors acting in the shared vertex. Lastly, the stress is obtained from the knowledge of the moment, moment of inertia and distance to the outer fibre.

Since it is not possible to output the shear forces directly from the rod goal, this functionality is implemented in another Grasshopper component, which collects the necessary information from the other goals after the solver. It takes a list of lines as input and identifies the corresponding bending planes and moments at its end points. As the moments are absolute values due to the calculation method, the orientations of the bending planes become essential to identify the signs of the moments and thereby calculate the difference.

4.4.4 Beam element

Whilst the bar and cable goal function on their own, the rod goal always need to be accompanied by the bar goal in order to avoid drifting of the nodes as there are otherwise no forces to maintain the distances between them. Thus, a bar goal combined with a rod goal represent a beam element. The reason why these are not combined into a beam goal (possibly with self-weight included) is to be more transparent about the implemented methods and thereby clearly show which input is necessary to define certain structural behaviours. The general understanding of a beam element is related to one line segment only, whereas a beam element in this set-up requires two consecutive lines and sorting along the splines. A beam element in this set-up would therefore most likely be unintuitive and used in a wrong way.

Listing 4.4 – K2Engineering rod goal. Simplified with help from Daniel Piker

```

1  public class RodGoal : GoalObject
2  {
3      double inertia, zDist, restAngle;
4
5      public RodGoal(Line LA, Line LB, double E, double I, double z, int opt)
6      {
7          PPos = new Point3d[4] { LA.From, LA.To, LB.From, LB.To };
8          Move = new Vector3d[4];
9          Weighting = new double[4] { E*I*1e-6, E*I*1e-6, E*I*1e-6, E*I*1e-6 };
10         inertia = I;
11         zDist = z;
12
13         if (opt == 0) {
14             restAngle = Math.PI;
15         }
16         else if (opt == 1) {
17             restAngle = Vector3d.VectorAngle(new Vector3d(PPos[0] - PPos[1]), new
18                 Vector3d(PPos[3] - PPos[2]));
19         }
20     }
21
22     public override void Calculate(List<KangarooSolver.Particle> p)
23     {
24         Point3d P0 = p[PIndex[0]].Position;
25         Point3d P1 = p[PIndex[1]].Position;
26         Point3d P2 = p[PIndex[2]].Position;
27         Point3d P3 = p[PIndex[3]].Position;
28         Vector3d V01 = P1 - P0;
29         Vector3d V23 = P3 - P2;
30         Vector3d V03 = P3 - P0;
31         double currentAngle = restAngle - Vector3d.VectorAngle(-V01, V23);
32         Vector3d n = Vector3d.CrossProduct(-V01, V23);
33
34         Vector3d shearA = Vector3d.CrossProduct(-V01, n);
35         Vector3d shearB = Vector3d.CrossProduct(V23, n);
36         shearA.Unitize();
37         shearB.Unitize();
38         double shearAVal = (2.0 * Math.Sin(currentAngle)) / (V01.Length * V03.Length);
39         double shearBVal = (2.0 * Math.Sin(currentAngle)) / (V23.Length * V03.Length);
40         shearA *= shearAVal;
41         shearB *= shearBVal;
42
43         Move[0] = shearA;
44         Move[1] = -shearA;
45         Move[2] = shearB;
46         Move[3] = -shearB;
47     }
48
49     public override object Output(List<KangarooSolver.Particle> p)
50     {
51         ...
52         double moment = Move[0].Length * Weighting[0] * V01.Length * 1e3;
53         double bendingStress = (moment * zDist) / inertia;
54         Vector3d planeYAxis = -(Move[1] + Move[2]) / 2.0;
55         Vector3d planeXAxis = Vector3d.CrossProduct(planeYAxis, n);
56         planeYAxis.Unitize();
57         planeXAxis.Unitize();
58         Plane pl = new Plane(P1, planeXAxis, planeYAxis);
59
60         var Data = new Object[4] { PIndex[1], pl, moment * 1e-6, bendingStress };
61         return Data;
62     }
63 }

```

4.4.5 Example: Elastica

The developed rod goal is used to model elastica curves using the same set-up from Subsection 2.1.3 and the resulting shapes are similarly compared to the benchmark study by Adriaenssens and Barnes (1999). In this case, it is possible to generate the shapes by applying forces to the end points of the line instead of prescribing the displacement thus exactly replicating the benchmark study. The four different buckled states as well as plots of the bending moments and axial forces are shown in Figure 4.4.4.

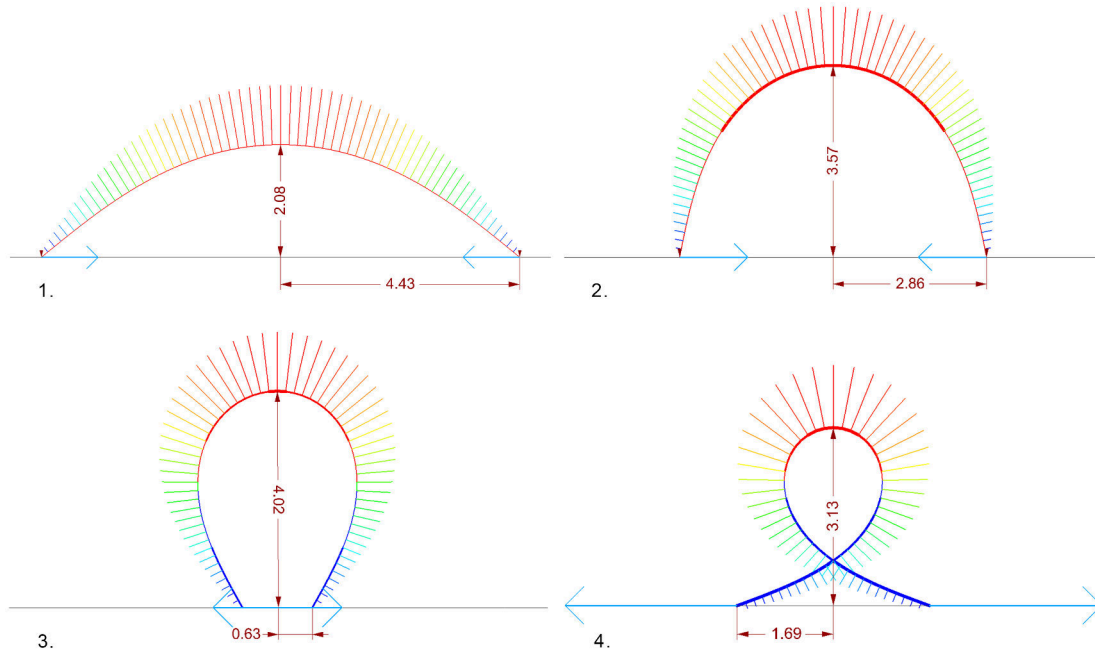


Figure 4.4.4 – Elastica curves generated by K2Engineering for the four buckled states. The measurements are given in meters

Table 4.4.2 shows the deviation of the width/height ratios in comparison with the benchmark study. It is seen that the results are mostly within 1% deviation for all four buckled states, which emphasise a reliable implementation of the rod goal. The third buckled state deviates slightly more in the horizontal direction (3%) but is still evaluated to be within an acceptable range. Furthermore, the maximum axial forces and bending moments are directly extracted from the analysis. The maximum compression force occurs at the top of each elastica curve and is equal to the applied force at the end points. The maximum bending moment also occurs at the top and increases in each state due to the more tight curvature.

Similar to the study in Subsection 2.1.3, the second buckled state is used as initial geometry, pinned at both ends and subjected to a vertical point load of 10 kN at the middle. The behaviour with and without prestress is shown in Figure 4.4.5. In the case where prestress is included, the largest moment is reduced and shifted a bit to the right/left of the middle due to the applied point load that contributes with a moment in the opposite direction at the top. The results are very similar to the moment distribution calculated from superposition (referring to Figure 2.1.7

| Buckled state | 1 | 2 | 3 | 4 |
|-------------------|-------|-------|-------|-------|
| Force [kN] | 10.48 | 12.67 | 18.46 | 39.48 |
| x/L deviation [%] | 0.3 | 0.3 | 2.8 | -0.5 |
| y/L deviation [%] | -0.9 | -0.1 | 0.02 | 0.1 |
| N max [kN] | -10.5 | -12.6 | -18.4 | -39.3 |
| M max [kNm] | 21.8 | 45.2 | 74.2 | 123.7 |

Table 4.4.2 – Elastica curves obtained from K2Engineering. The deviation is measured with regard to the benchmark study by Adriaenssens and Barnes (1999) as outlined in Chapter 2

(bottom)), which emphasise that this is an appropriate method to include the prestress effect if a standard finite element software is used. In the case where prestress is not included, the moment distribution and the maximum displacement are similarly consistent with the Karamba results (referring to Figure 2.1.7 (middle)). The small deviation in both cases is due to the shape difference from the two form-finding methods.

Whilst the principle of superposition is useful to take the prestress effect into account with regard to the moment distribution, the influence on the deflection is ignored. In this example, the prestress is responsible for increasing the maximum displacement of the structure (46 mm compared to 42 mm at the top). This behaviour is opposite to the observations made from the Karamba analysis where prestress was attempted to be modelled as an external load (referring to Subsection 2.2.3). Here the prestress was responsible for decreasing the maximum displacement to 33 mm (referring to Figure 2.2.5). The method used for the Atmeture project therefore seems inadequate to mimic the effect of prestress on the deflections.

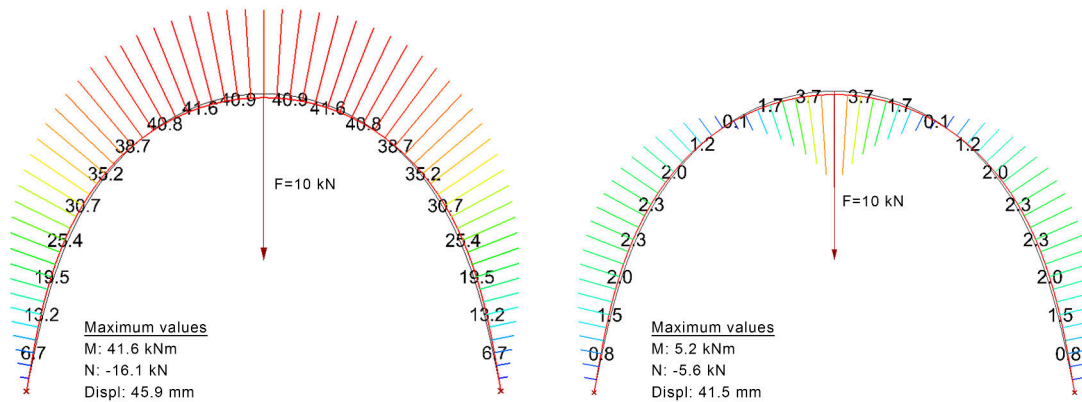


Figure 4.4.5 – Elastica behaviour with (left) and without (right) prestress and a point load of 10 kN applied at the middle

4.5 Goal limitations and further developments

4.5.1 Biaxial bending and torsion

The implementation of the rod goal is limited to circular cross sections and necessitates two consecutive line segments in order to mimic bending in a 3 DoF system. The bending resistance is defined within the plane of the two line segments. For a spatial rod element this results in varying bending planes along the length. The moments are therefore an expression of the “resulting moments” rather than the moments related to the two cross section axes. But since it is assumed that the rod is made of a circular cross section with isotropic properties, this is not a problem. However, it is important to notice that this definition does not take torsion into account.

When rectangular cross sections are used, this bending definition is no longer sufficient because it does not properly account for the anisotropic behaviour due to the lack of orientations in a 3 DoF system. In an attempt to address this shortcoming, Barnes et al. (2013) have extended the method to account for biaxial bending and torsion. This definition only relies upon three degrees of freedom but requires three consecutive line segments. The disadvantage of the method is that it imposes a restriction on the curvature radius, which means that it cannot model straight elements.

More advanced methods to model biaxial bending and torsion have been developed in the computer industry to e.g simulate ropes and hair (Umetani et al., 2014) (Grinspun et al., 2008). More recently this approach has made its way to the building industry for the form-finding of elastic gridshells made from rectangular cross sections (Peloux et al., 2015). The different variations are all based on a curve-angle representation to reduce the system from 6 DoF to 4 DoF and require the specification of the cross section orientation along the rod element either by frames or so-called ghost points. The frame method is based on the concept of defining a zero-twisting frame (also known as a Bishop frame) along the centreline of the rod, which is calculated from the curvature normal and automatically adapts during the simulation. The angle between this frame and the material frame is then used to calculate the bending about both axes and the twist. The biaxial bending behaviour can be translated into forces in a similar way as the simple rod goal but with move vectors acting in two planes. The torsion behaviour is more difficult to translate into move vectors and the math behind this is very advanced and hence outside the scope of this thesis. A biaxial bending and torsion goal is however evaluated to be the most important improvement of this K2Engineering plug-in in the future. Alternatively, it can be implemented according to the method described by Olsson (2012) when Kangaroo2 is extended to a 6 DoF system.

4.5.2 Constant strain triangle

A natural next step is also to implement a constant strain triangle goal (CST), which can be used to model membranes. Whilst the current cable goal can be used to approximate the behaviour,

it is desirable to include the membrane specific behaviour, which includes shear stiffness and a more direct interaction between the warp and weft direction of the material. Both form-finding and analysis methods are described by Barnes (1999).

4.6 Convergence

Convergence implies that the movement of the system is so small that it can be considered to be in static equilibrium. As described in Chapter 3, this is determined from a threshold value of the kinetic energy. In most cases, a value of $1 \cdot 10^{-15}$ has proven to be sufficient in order to obtain results that are consistent with analytical solutions or results from other finite element software (referring to Appendix A). The reaction forces are sometimes a bit more sensitive to the threshold value due to their high stiffness values. This behaviour is noticeable if the maximum reaction force constantly changes between two values where one is often much larger than the other. In that case, it is useful to reduce the convergence threshold to $1 \cdot 10^{-20}$. The ratio between the sum of the applied external loads and the reaction forces (which should equal 1.0) is a useful measure to estimate the convergence status.

The convergence speed is generally slower when axial and bending goals with big differences in stiffness are combined. In kangaroo2 terminology, this is referred to as soft and hard constraints (Piker, 2015c). Unless the cross section profile is small, the bending stiffness EI will be much larger than the axial stiffness EA . In that case, experiments have shown that the convergence can be speeded up by slowly increasing the bending stiffness during the simulation. However, in some cases this method is not applicable as undesirable deformations occur, which take a long time to recover from. It also means that the user has an active role during the simulation, which is undesirable.

During the implementation phase, the author experienced some unexpected convergence behaviour related to the chosen model scale. At first the model scale was chosen to be in millimetres to be consistent with the standard units for cross section and material properties. However, this resulted in a remarkably slow convergence speed, which questioned the entire motivation for the calibrated structural behaviour. Through a lot of tests, the problem was narrowed down to the model scale, which was a bit surprising. Furthermore, it only seemed to be related to the bending goal. The issue is still unexplained but has been resolved by changing the model scale to metres instead. For convenience, the units for the cross section and material properties are unchanged as input values but automatically converted to appropriate units inside the component.

4.7 Stress summation

In addition to the goals themselves, other relevant functionalities have been implemented, which operate on the output from the goals. The summation of axial and bending stresses is one such

functionality, which is necessary to verify that a beam has sufficient load bearing capacity. The axial stress is constant throughout the cross section of the beam whereas the bending stress varies linearly as shown in Figure 4.7.1. The total stress value is therefore calculated from the summation of the axial stress and the bending stress at the outer fibre.

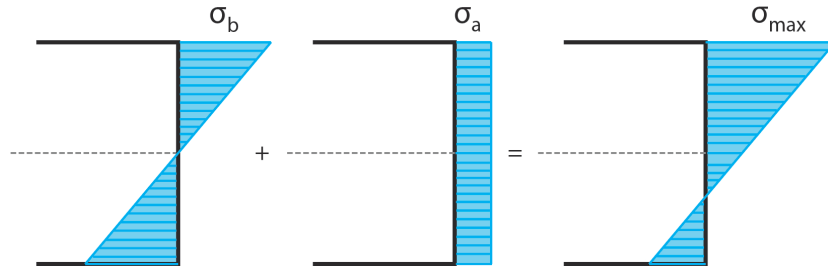


Figure 4.7.1 – Summation of axial and bending stresses

4.7.1 Implementation

The calculation of the total stress value is straight forward but since the bending stress is calculated per vertex, the axial stress is calculated per line and multiple beams might meet in one point, the main task for the component is to keep track of the data structure to get the correct results. It is important that intersecting rods are separated into different branches in a tree structure because otherwise it is impossible to tell which bending moment belongs to which line at the intersection. As it is desirable to get one total stress value per line segment, the procedure is to determine the bending stresses at the start and end point of the line, take the maximum value and add that to the axial stress. Absolute values are used since the bending stresses have no sign. The necessary input values for the stress summation component is shown in Table 4.7.1.

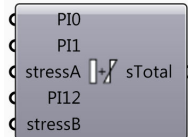
| GH Component | Input | Output |
|---|---|--------------------|
|  | Start particle index (bar) End particle index (bar) Axial stress [MPa] Shared particle index (rod) Bending stress [MPa] | Total stress [MPa] |

Table 4.7.1 – Input and output from the stress summation component

4.8 Non-linear buckling

The Kangaroo2 platform is also useful for the implementation of a buckling analysis. Buckling is an instability mode for a structural member in compression, which leads to a sudden failure

before the ultimate compression strength of the material is reached. The behaviour is characterised by a significant change in the displacements for only a small load increment and is referred to as the bifurcation point. In the context of bending-active structures, where the members mostly work in bending and compression e.g. when being part of a gridshell, it is important to be able to evaluate the buckling capacity. Especially since buckling is considered to be the dominant failure mode for reticulated shells (Malek, 2012). Furthermore, it is interesting to study the influence of prestress on the buckling capacity, which is made possible with this tool. Figure 4.8.1 shows a bending-active steel ribbon placed in a lawn with curious goats challenging its stability by jumping up and down on it. The image is extracted from a YouTube video (Murs, 2014) from which it is evident that the prestress from the initial bending of the ribbon results in a very ductile and bouncy behaviour where the ribbon buckles into different shapes but restores its initial configuration afterwards (helped by the ground).



Figure 4.8.1 – Goats balancing on a flexible steel ribbon (Murs, 2014)

The buckling capacity is typically quantified by means of a buckling load factor (BLF), which expresses how much the applied load can be scaled before buckling occurs. There are two strategies to obtain such measure; a linear and a non-linear. The linear approach is an eigenvalue problem of the system's stiffness matrix, which is either increased or decreased by a geometric stiffness matrix based on the initial stress state. This implies that the deformed shape is not taken into account, which means that this approach often overestimates the buckling load factor (Cook, 1995). The buckling modes from the analysis only highlight the problematic areas but the scale of the deformations are arbitrary and therefore cannot be used to evaluate how critical a certain mode is. The method is however widely used due to its simplicity. The non-linear approach aims to take the deformed shape into account by using an incremental load procedure, which performs a number of equilibrium iterations in between each load increment and traces

the displacements after each step. This analysis is more difficult to perform and requires the specification of several parameters that all have an influence on the result. The output of the analysis is a graph rather than one number and it is therefore up to the user to estimate the buckling load factor from this information. However, this makes it possible to evaluate how critical the deformation is for certain load factors.

The Karamba plug-in contains a buckling modes component, which implements the linear buckling approach. One of the inputs is a value for the maximum number of iterations that are used to calculate the buckling load factors. This often leads to the misunderstanding that Karamba implements the non-linear approach but that is not the case. The number of iterations in this case refers to the iterative approach that is necessary to perform an eigendecomposition of the stiffness matrix rather than the number of iterations to reach equilibrium in each load increment. One of the disadvantages with the implementation in Karamba is that it is impossible to extract any information if buckling is detected. This means that it is difficult to locate the problem and it also makes it impossible to e.g. ignore the first buckling mode which may be very local. This is generally not a problem in other finite element software as those programs are capable of outputting buckling load factors less than 1.0.

The non-linear solution strategy makes dynamic relaxation a useful technique and hence points towards Kangaroo2 for the implementation in Grasshopper. The author has previously implemented such non-linear buckling analysis procedure in Grasshopper using Kangaroo2 (Brandt-Olsen, 2015) but this development only focused on the simulation of continuous shells and without calibrated structural properties thus only used for ranking structures relative to each other. The implementation has therefore been generalised and refined as described in the following.

4.8.1 Implementation

The procedure as outlined in Brandt-Olsen (2015) is implemented in a separate buckling analysis Grasshopper component, which utilises the possibility to script the Kangaroo2 solver. The component with its input and output is shown in Table 4.8.1. The analysis is generalised to work for any kind of structure by defining the input as a list of permanent Kangaroo2/K2Engineering goals (“PGoals”) and a list of K2Engineering load goals (“LGoals”). The permanent goals describe the structure’s behaviour with respect to the initial configuration and remain unchanged during the simulation. They can be both native Kangaroo2 goals or the calibrated K2Engineering goals. The load goals are on the other hand continuously scaled according to the start load factor (“LFStart”) and the step size (“LFStep”). During each load increment, the Kangaroo2 solver performs a certain amount of iterations to reach equilibrium based on all the goals and a user defined threshold value (“thres”). The updated positions are subsequently used to calculate the RMS value of all the nodal displacements and from that determine whether buckling occurs or the simulation shall continue. The buckling criteria is defined from an angle value (“alfa”) related to the gradient of the load-displacement curve and a maximum displacement value (“dMax”), which is useful if the structure exhibits a very ductile behaviour. The component outputs the

overall buckling load factor, all the preceding load factors, displacement RMS values, vertices and the output from the permanent goals for each load increment.

The code behind this non-linear buckling analysis component is available on Github (Brandt-Olsen, 2016) and a simple example of column buckling can be found in Appendix A.


| GH Component | Input | Output |
|---|---|--|
|  | Permanent K2 goals K2Eng load goals Start load factor Load factor step size Angle criteria[°] Max displacement criteria [m] Equilibrium threshold Output option <i>(True: output all)</i> | Buckling load factor Load factors Displacement RMS values [m] Vertices Goal output |

Table 4.8.1 – The input and output from the buckling analysis component

4.8.2 Example: Elastica buckling with and without prestress

This example studies the buckling behaviour of three different elastica curves, that are modelled as either bending-active (with prestress) or fabricated curved elements (without prestress) and subjected to three different load scenarios. The elastica curves are made from a GFRP (glass fibre reinforced polymer) material and all have the same total length of 3.0 m but with different length/height ratios in order to evaluate the influence of curvature on the buckling capacity as well. Table 4.8.2 shows the properties that are used for the set-up and Figure 4.8.2 illustrates the different configurations, which in total result in 18 cases that are investigated.

| Elastica | Length(x) | Height(y) | Segment | Diameter | Young's modulus | UDL | P(L/2) | P(L/3) |
|----------|-----------|-----------|---------|----------|-----------------|-------|--------|--------|
| 1 | 2.75 m | 0.54 m | 50 mm | 8 mm | 45 000 MPa | 5 N/m | 2 N | 2 N |
| 2 | 2.50 m | 0.72 m | | | | | | |
| 3 | 2.00 m | 0.98 m | | | | | | |

Table 4.8.2 – The properties used for the buckling study of elastica curves

The results from each of the 18 cases are shown in Figure 4.8.3, Figure 4.8.4 and Figure 4.8.5. For the cases without prestress, the buckling analyses are performed with both Karamba and Kangaroo2 in an attempt to increase the reliability of the Kangaroo2 implementation (even though they are based on two different methods).

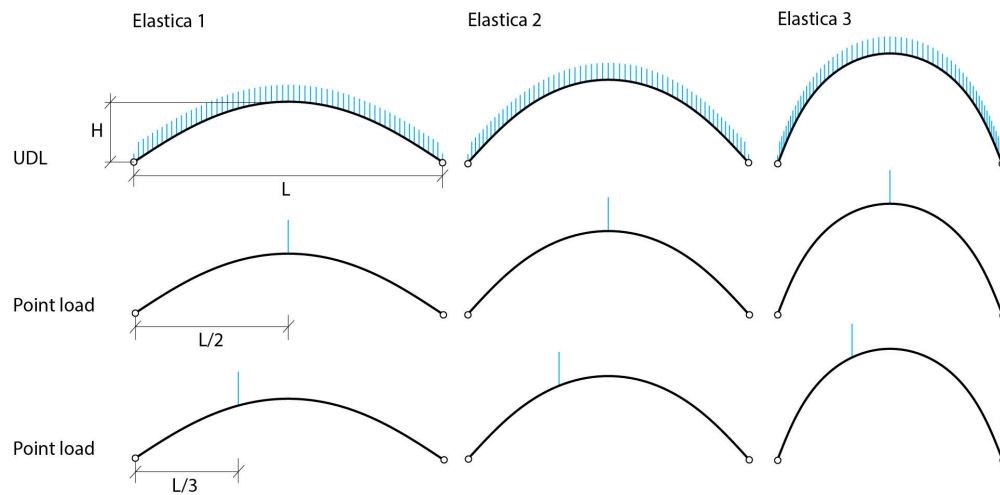
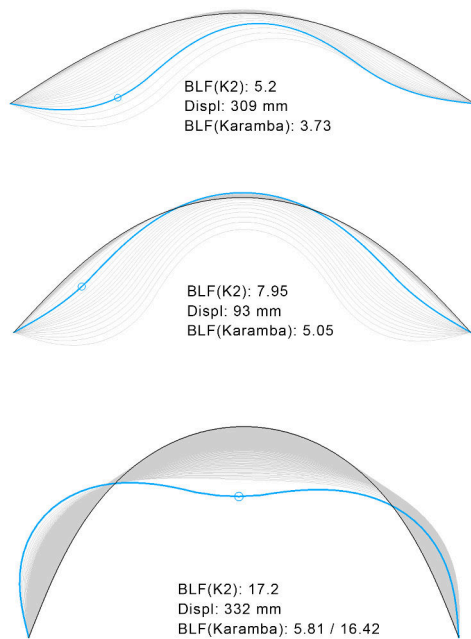


Figure 4.8.2 – Different configurations for the buckling analysis of elastica curves with and without prestress from bending

UNIFORM DISTRIBUTED LOAD

WITHOUT PRESTRESS



WITH PRESTRESS

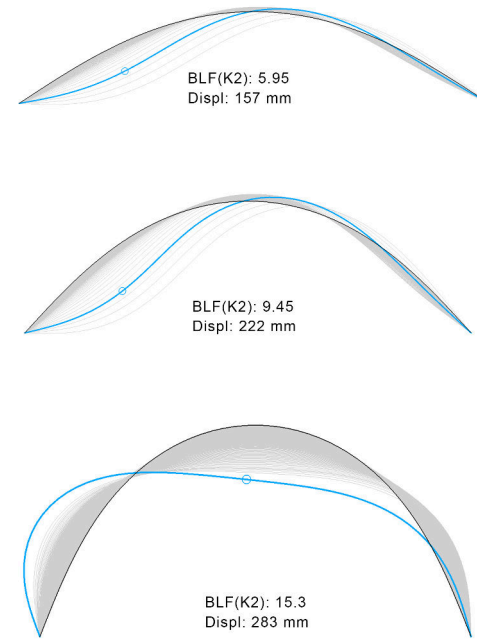
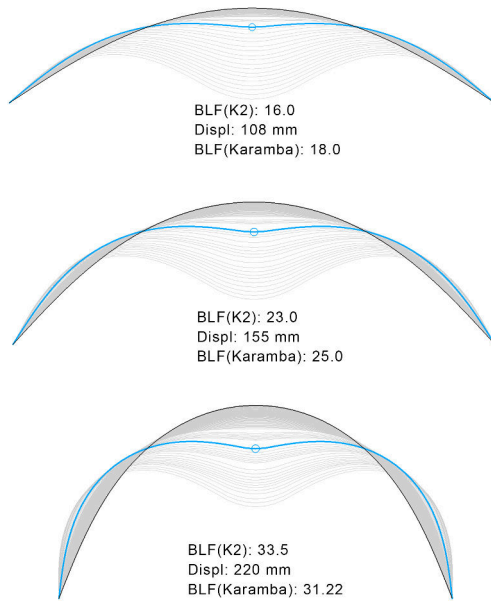


Figure 4.8.3 – Buckling of elastica curves without (left) and with (right) prestress from bending and subjected to a uniform distributed load

SYMMETRIC POINT LOAD

WITHOUT PRESTRESS



WITH PRESTRESS

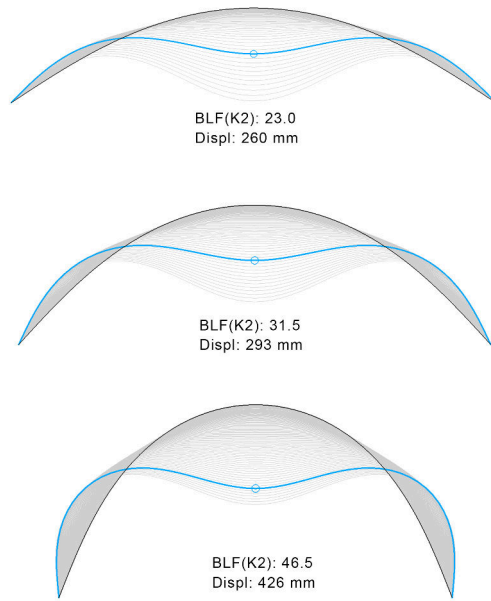
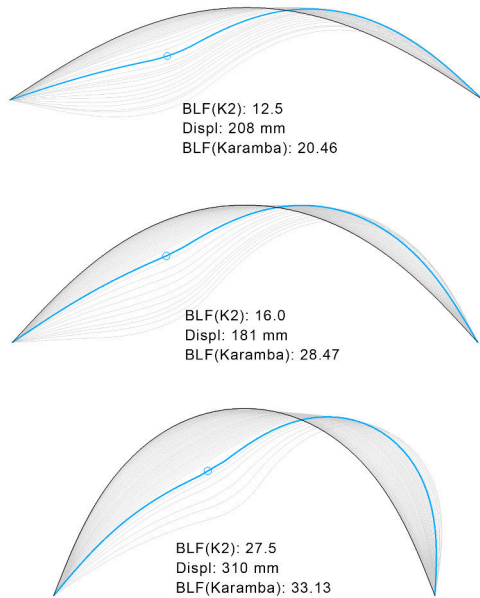


Figure 4.8.4 – Buckling of elastica curves without (left) and with (right) prestress from bending and subjected to a symmetric point load

ASYMMETRIC POINT LOAD

WITHOUT PRESTRESS



WITH PRESTRESS

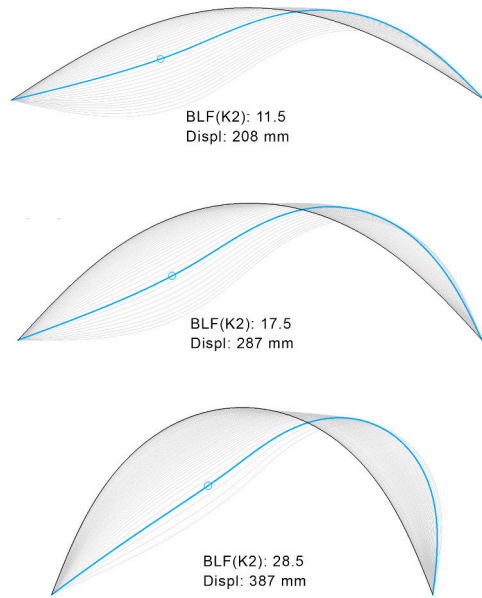


Figure 4.8.5 – Buckling of elastica curves without (left) and with (right) prestress from bending and subjected to an asymmetric point load

The following observations are made:

- **Karamba vs. Kangaroo2 (without prestress).** The results from the two analyses are very similar for the scenarios with a uniform distributed load and a symmetric point load. In the first case, Karamba underestimates the buckling load factor slightly whereas it is the opposite case for the latter. The deviation between the two analyses is larger for the asymmetric point load scenario, where Karamba overestimates the buckling load factor with a more significant amount. Even though the buckling analyses are based on two different approaches, it is useful to observe a similarity between the results (at least for the first two cases) to gain more confidence in the implementation.
- **Curvature.** It is clear that more curvature is beneficial for the buckling capacity hence elastica 3 performs better in all three load scenarios with and without prestress. For the uniform distributed load it is especially interesting that the increased curvature for elastica 3 eventually helps to stiffen the sides as they become more vertical, which turns the middle into the weakest part instead.
- **Load.** From this set-up it is not possible to compare the results between the uniform distributed load and the point loads. This is because the resultant force from the UDL is larger than the point loads and it is therefore natural the buckling load factors are smaller for this load scenario. However, the point loads have the same magnitude and the influence of the force location can therefore be evaluated. Interestingly, the buckling load factors from Karamba indicate that the elastica curve is most vulnerable to buckle under the symmetric point load whereas the results from Kangaroo2 suggest that the asymmetric point load is more critical. Since the UDL case highlights that the sides are the weakest part, it seems more intuitive that the elastica curve buckles earlier under the asymmetric point load.
- **Prestress effect.** As Karamba is not capable of modelling prestress from initial bending, the effect is evaluated from a comparison between the Kangaroo2 results only.
Uniform distributed load: For the uniform distributed load it is observed that the increasing prestress from curvature only has a beneficial effect on the buckling load factor for elastica 1 and 2. It therefore seems like the prestress has a stiffening effect when the curvature is low. However, for elastica 3 the prestress level becomes sufficiently high that it weakens the ability to withstand the applied load instead. This is an interesting balance and makes it hard to predict whether prestress has a positive or negative effect for the stability of a structure. Furthermore, it is observed that the buckling modes from the prestressed state emphasise the weakness of the sides and thus encourage a more asymmetric buckling behaviour.
Point loads: It is clear that the prestress introduces a much more ductile behaviour, which makes it very difficult to determine a buckling load factor. The displacements escalate slightly at a later stage but it is questionable whether this behaviour can be characterised as buckling at all. In this case, the maximum displacement is a more critical design criteria

for the structure. It makes sense that the prestress introduces a more ductile behaviour because the sudden change in displacements only occurs due to compressive forces and not bending action, which the prestressed elements are dominated by. These observations correspond well with the behaviour of the steel ribbon from Figure 4.8.1.

The reason why the ductile behaviour is more significant for the prestressed elements in the point load scenarios may be related to the similarity between the elastica curves and a catenary shape. A catenary shape is very effective when subjected to a uniform distributed load as it transfers the forces in pure compression. However, this makes it vulnerable to buckling. On the other hand, if the catenary is subjected to a point load instead, it becomes less efficient as it resists the applied load by a combination of axial and bending action. When bending action becomes more dominant, the risk of buckling decreases.

Chapter 5

Smart Geometry 2016

Smart Geometry is a yearly event, which focuses on the application of digital tools in the design process. It consists of a four days workshop followed by two conference days where the interaction between new technology and a broad variety of disciplines is explored. The aim is to create inspiring and informed architecture whilst simultaneously handling fabrication challenges. This year the event was hosted by Chalmers University of Technology in Gothenburg and the theme was “Hybrid Domains” (SmartGeometry, 2016).

5.1 Workshop proposal

The Smart Geometry workshop was divided into ten so-called “clusters” with different topics related to Hybrid Domains. These ten clusters were selected amongst more than forty workshop proposals by the Smart Geometry team. The author was part of the expert team behind one of the clusters called “Calibrated Modelling of Form-Active Hybrid Structures” together with Gregory Quinn (KET at University of Arts in Berlin), Anders Holden Deleuran (CITA), Daniel Piker (McNeel and Foster+Partners), Will Pearson (McNeel and Format) and Harri Lewis (Mule Studio). The expert team had a shared interest in using Kangaroo2 with calibrated structural properties from various different projects in the past and this formed the basis for the cluster proposal.

The specific research question was: *“How can we implement projection based dynamic relaxation methods to unlock the latent potential of form-active hybrid structures?”* (Deleuran et al., 2016)

A number of contributions by the cluster experts helped to support this research question:

- Extending Kangaroo2 to output and visualise structural properties (as described in Chapter 4)
- Validation of structural behaviour through comparisons with analytical solutions and other FEA packages (referring to Appendix A).

- Validation of structural behaviour through physical testing by means of a projection board
- A modelling pipeline that integrates coarse geometry modelling, form-finding, analysis and fabrication into one smooth workflow
- Continuous improvements of the Kangaroo2 solver and addition of new goals to support the initiatives mentioned above

It was decided to limit the type of elements to GFRP (glass fibre reinforced polymer) rods and plastic cables to ensure a reasonable outcome of the workshop within the very limited time frame.

5.2 Projection board

The projection board enabled the interaction between digital and physical and thus helped to verify identical behaviours. The set-up was developed by Gregory Quinn and consisted of a white board with pre-drilled holes in a grid arrangement, a projector which overlay a digital simulation using the same grid, different bolts for pinned and roller type supports (see Figure 5.2.1), GFRP rods with various radii, strings and 50 g weights. The first step was to select a specific set-up, which was modelled both digitally and physically with identical support conditions and weights. It was then possible to determine the Young's Modulus of the GFRP rods by adjusting the stiffness value in the digital model until the same deformed shape emerged. The set-up could then be adjusted e.g. by applying more weights and it was observed how the deformed shape in the digital and physical model remained identical to each other and thus confirmed the accuracy of the simulation.

In addition to plotting the deformed shape, the set-up also allowed to overlay other structural properties such as normal-, shear- and reaction forces as well as bending moments at interactive speed. This helped to explain structural principles and the participants were able to directly relate them to the physical experiences they were having at the same time. The projection board thus proved to be a very useful learning tool. It was furthermore extended to work with a physical marker, which was recognised by Grasshopper via a webcam and this allowed the deformation of the physical rod to be controlled by hand (rather than weights) whilst simultaneously overlaying the digital plots (Quinn, 2016).



Figure 5.2.1 – Projection board kit for different support types (credit to Gregory Quinn)



Figure 5.2.2 – Projection board setup (credit to Gregory Quinn)

5.3 Modelling pipeline

The modelling of coarse geometry, form-finding, analysis and fabrication details were integrated into one smooth workflow (modelling pipeline) developed by Anders Holden Deleuran. All functionalities were scripted with custom Python components in Grasshopper but used clever methods to only operate in the background such that all modelling activity happened in Rhino and key parameters from the Grasshopper script were accessible through a remote control panel.

The modelling pipeline was divided into the following subcategories:

ASSEMBLY GEOMETRY

The coarse geometry was modelled with points and polylines in Rhino and arranged in appropriate layers (cables, beams or anchors). The Grasshopper script then automatically referenced these polylines, subdivided them into smaller segments based on a defined length and sent them back into Rhino. The division into smaller segments made it easy to snap different elements together in order to quickly model different geometries.

FORM-FINDING

By pressing a “solve” button in the remote control panel, the elements in the different layers were assigned with fictitious stiffness properties according to their structural characteristics to initiate a form-finding process using Kangaroo2 with native goals. The form found shape was subsequently sent back to Rhino such that the effect on the shape from adding/removing elements could be immediately evaluated. The fictitious stiffness properties were useful for modelling at interactive speed and still reflected the structural behaviour.

STRUCTURAL

After the form-found shape was established, a “calibrate” button enabled these fictitious properties to be replaced by real material properties (specified by the user) such that stresses and deflections could be accessed. This step changed the original Kangaroo2 goals into K2Engineering goals and added self-weight to the structure. Visualisation of the forces and moments were subsequently activated by check-boxes in the remote control panel.

TOPOLOGY AND FABRICATION DATA

To make the construction of the hybrids easier, the modelling pipeline had an option to output an assembly graph (using the external libraries GraphViz and NetworkX). This graph showed the connectivity between all the elements, their type (beam or cable) and lengths.

Initially the participants were given the task of making small physical models to acquire an understanding of the materials and how they could be combined to produce hybrid action. Good hybrid behaviour was characterised by an efficient force distribution when the model was subjected to pushing/pulling at different locations, which was achieved by having more cables than rods (axial forces being more effective than transferring loads via bending action). One of the models, which performed particularly well is shown in Figure 5.3.1. It was made from

two “tear-drop” shaped rods of different lengths that met at the top, were bent backwards and held in place by a back tie. A fan-like cable system in front helped to distribute the loads in an efficient way and made the whole structure very stiff.



Figure 5.3.1 – Exploration of hybrid concepts

Subsequently, the participants had to recreate the shape and simulate the structural behaviour of their physical prototype using the modelling pipeline. It proved to be an important tool to explore many variations in a short time as the physical prototyping quickly revealed the difficulty of connecting elements together and cutting the elements in the right lengths to obtain a desired spatial shape.

Figure 5.3.2 shows the recreation of the physical model from Figure 5.3.1 in a digital environment and consisted of the following steps:

1. Model coarse beam geometry with polylines and place an anchor point at the bottom
2. Add a cable from the bottom to the midpoint of the beam
3. Move the two outer points of the triangle out of plane and increase the cable stiffness to obtain a desirable three dimensional tear-drop shape
4. Add cables in a fan arrangement to stiffen the structure
5. Calibrate the model to real material properties to verify that the beam can obtain the desired curvature without exceeding the material strength

6. Extract assembly graph to get an overview of the connectivity and element lengths

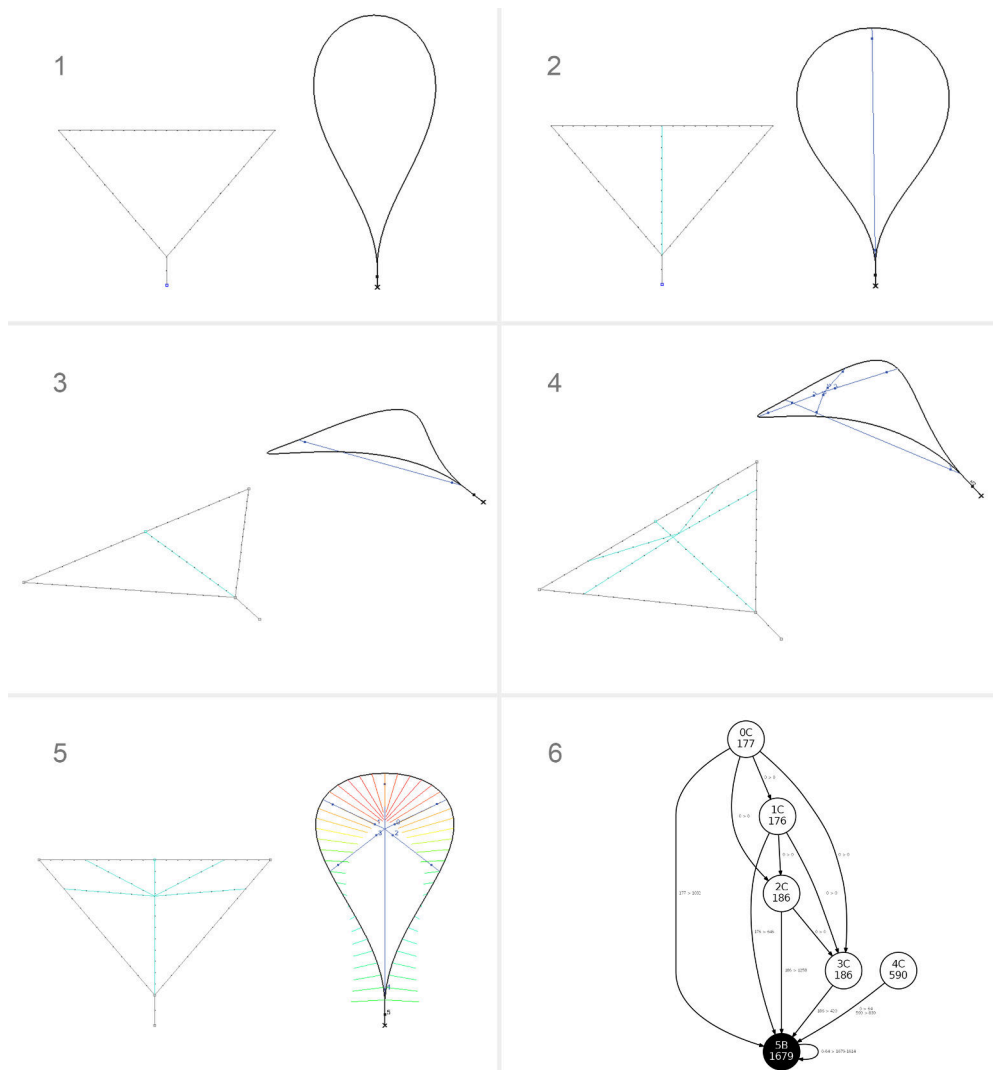


Figure 5.3.2 – Digital recreation of hybrid concept

5.4 Tower challenge

The participants were challenged to build a tall hybrid tower based on their experience from the smaller models. The strategy was to design a simple hybrid unit and then stack and/or rotate these to create the entire tower structure. The concept from above was further refined and rotated around a vertical axis to create one layer of the tower. Multiple layers were then added on top of each other (bottom of tear-drop beam to middle of cable fan) and by flipping the bent direction of the units in one layer, interesting curvature variations emerged as seen in Figure 5.4.1.

The structure was analysed under self-weight, which resulted in maximum bending stresses of 525 MPa and a maximum deflection of 50 mm at the middle of each beam in the second layer. The bending stresses were evaluated to be within an acceptable range as the flexural strength of the GFRP material was 900 MPa. The total weight of the 3 m tall tower was only 3 kg!

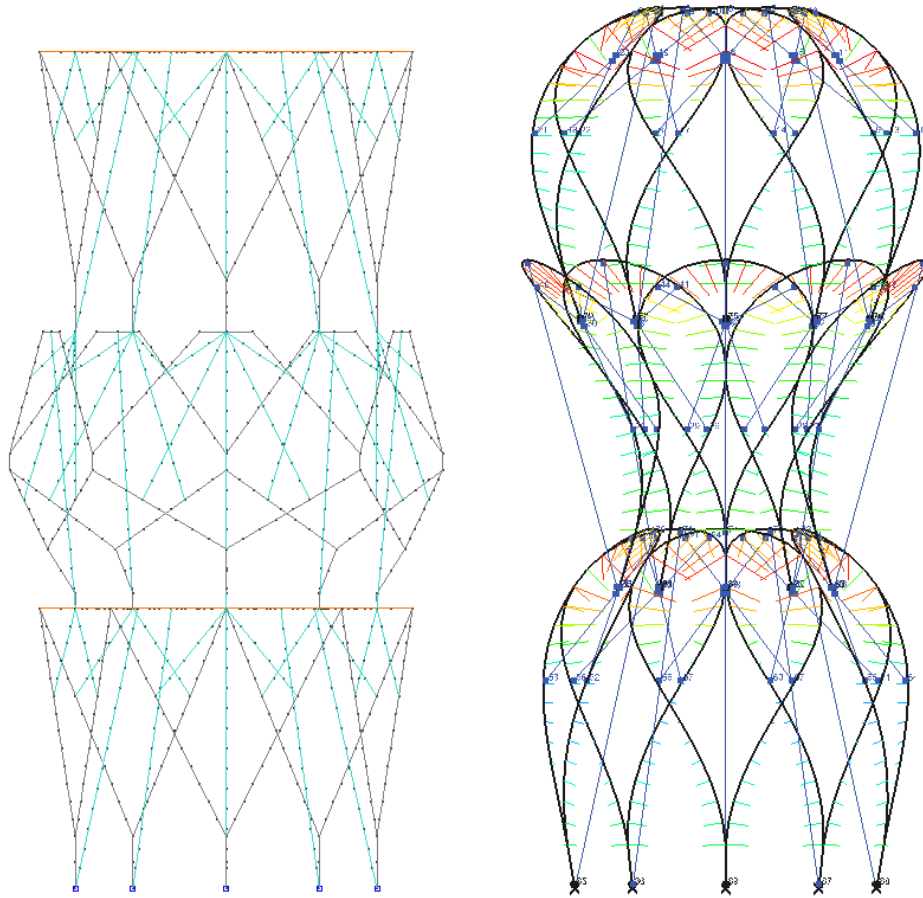


Figure 5.4.1 – Digital model of the tower design (credit to Léa Boulic)

The tower was originally designed as three stacked layers but due to a very smooth assembly process and the fact that the structure was so stiff, it was decided to add another two layers resulting in a 5 m tall hybrid tower as seen in Figure 5.4.2. A direct comparison between the physical and digital model was difficult to make because of the extra layers and the fact that the connection details were created from tape and staples and hence associated with inaccuracies. However, the built tower felt surprisingly stable when subjected to different kinds of loads and was part of the workshop exhibition on the last day.



Figure 5.4.2 – Final hybrid tower (5 m tall)

Chapter 6

GFRP gridshell design

This chapter describes the design of a bending-active gridshell made of GFRP rods. The purpose of the case study is to demonstrate the capabilities of the developed analysis tool and the advantages of having the entire workflow integrated in Rhino/Grasshopper.

6.1 Brief

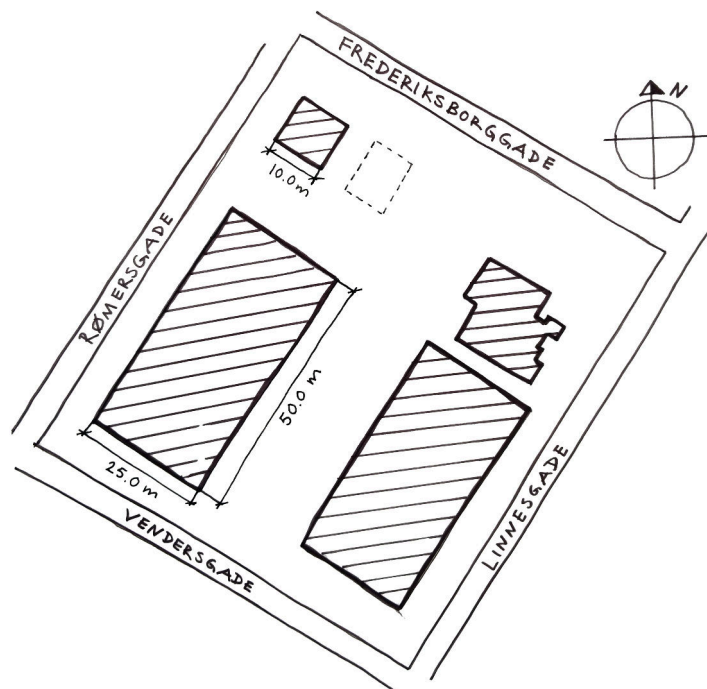


Figure 6.1.1 – Gridshell location at Torvehallerne (dotted line)

The gridshell is a design for an outdoor market stall at Torvehallerne in Copenhagen. The aim is to create an interesting and inspiring space, which adds quality to the outdoor area in-between the two main halls and encourages people passing by on Frederiksborggade to enter the market area (see Figure 6.1.1). The gridshell has to fit within a footprint of 10.0 x 8.0 m and the height is restricted to be within 2.1 - 3.5 m.

6.2 Shape generation

There are several different strategies for the design of gridshells, which are highly related to the intended construction method. One common technique is to lay out a flat grid in plan and then push parts of the boundary inwards to create the spatial shape. The key behind this strategy is that the grid has no shear rigidity during the form-finding process but when the desired spatial shape has been found, it is locked into its position by adding bracing elements. The largest design flexibility is obtained by having pinned connections between the laths such that wrinkles are avoided. This technique was used for e.g. the Multihalle Mannheim (Naicu et al., 2014).

If the desired spatial shape is already defined, the problem rather becomes how to lay out a grid on that surface. The so-called compass method addresses this problem by iteratively setting out equal distance point on a surface based on two input curves (Popov, 2002). The resulting geometry is also known as a Chebychev net and the advantage is that the grid arrangement can be defined in the plane and then lifted or draped into its spatial configuration. This technique was used for the Saville Garden gridshell (Naicu et al., 2014). The curve network may also be defined as geodesics lines on the input surface, which is particularly useful when timber laths with significant anisotropic properties are used for a bending-active gridshell (like the OnGreening Pavilion). Thereby bending about the strong axis is avoided and this reduces the stress level from initial bending of the elements. The choice of using geodesic lines necessitates a sequential erection method.

The shape generation method for this gridshell design is an adapted version of the “grid-to-shape” method, which defines the spatial shape from the bending of a number of parallel primary rods. The secondary rods and bracing rods are subsequently added. This makes the assembly process a bit more complicated compared to the alternative approach where the entire grid is defined in the plane and pushed into its spatial shape. The reason for this choice is related to the current state of the modelling technique that is used during the form-finding process; the rods are modelled with both axial and bending stiffness along their lengths but this makes the connections rigid because the two coincident nodes at an intersection are combined into one. This implies that the rods cannot rotate at the intersections and hence restrain the structure too much during the form-finding process such that wrinkles develop. To avoid this, it is necessary to separate the two layers and model the connections with e.g. zero length springs to provide the necessary rotational freedom. The visual effect of this strategy is that the primary rods are no longer perfectly vertical as the grid arrangement influences the overall shape. However, this modelling refinement has not been included in this thesis.

The overall shape of the gridshell is defined by two boundary curves in the plane as illustrated in Figure 6.2.1 (1). For simplicity it is chosen that all the primary rods have the same length such that the order becomes irrelevant during the assembly (Figure 6.2.1 (2)). The spatial shape is then generated by moving the end points of the rods towards the defined boundary curves (Figure 6.2.1 (3-4)). Lastly, the secondary and bracing rods are added (Figure 6.2.1 (5-6)). The equal length constraint introduces an interesting “reciprocal” relationship between the width of the boundary curves and the height of the gridshell. Furthermore, it has the advantage of making the surface area approximately constant when the boundary curves are modified, which makes it possible to directly compare the structural performance of different configurations.

The asymmetry of the boundary curves creates a more interesting and less predictable space and it fits well into the site context since one side of the gridshell faces the building next to it and the other side opens up towards the market place as seen in Figure 6.1.1.

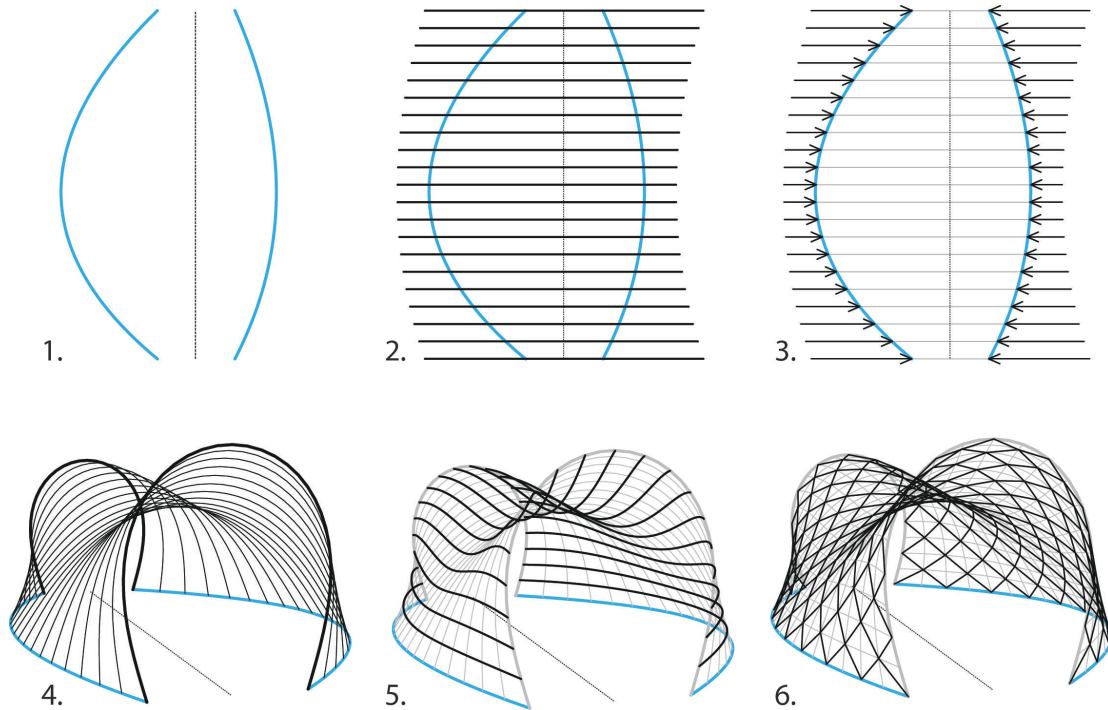


Figure 6.2.1 – Gridshell shape generation. 1. Definition of boundary curves. 2. Lay out primary rods of equal length in the plane. 3. Move endpoints inwards to fit the boundary curves. 4. The spatial shape defined by the boundary curves and equal length constraint. 5. Addition of secondary rods. 6. Addition of bracing rods

6.3 Material

It is chosen to build the gridshell from GFRP rods because the material is very light, flexible and has a high strength. All these properties are advantageous when designing bending-active structures. In comparison with timber, which is often used for bending-active gridshells, the strength/stiffness ratio is approximately 3 times larger and the strength/density at least 2 times

larger. Fibrolux (2016) is an example of a supplier of GFRP rods with the material properties as specified in Table 6.3.1 and with diameters available in the range between 1.2-80 mm.

The circular sections are well suited for the K2Engineering analysis and the size of the pavilion makes it possible to build it from a single layer of rods, which simplifies the modelling.

| | |
|-----------------------------|---------------|
| GFRP Young's modulus | 40 000 MPa |
| GFRP Density | 2000 kg/m^3 |
| GFRP strength | 900 MPa |

Table 6.3.1 – GFRP material properties (Fibrolux, 2016)

6.4 Design exploration

Initially five different variations within the defined design space are investigated. The structural performance of each variation is measured in terms of the axial forces, bending moments, reactions, stresses and deflections. Table 6.4.1 shows the design variables that are kept constant throughout this study.

| | |
|-----------------------------------|---------|
| Primary length | 8.0 m |
| Primary spacing | 0.5 m |
| Secondary spacing | 0.4 m |
| Free edge diameter | 24 mm |
| Primary diameter | 20 mm |
| Secondary/bracing diameter | 16 mm |
| Total weight | ~300 kg |

Table 6.4.1 – The predefined gridshell design parameters

Each gridshell design is subjected to a load case that includes the self-weight and a dominant wind load in the direction towards the side of the structure, where it is most vulnerable ($1.0 \cdot SW + 1.5 \cdot W$). The wind load is not easily defined for freeform shapes and it is not within the scope of this study to develop an accurate model for this. Instead the aim is to define a simplified version that reflects an approximate behaviour as described in the following:

- The wind load in the city for a height less than 10.0 m and a wind speed of 24 m/s is 0.4 kN/m^2 according to EN 1991-1-4 DK NA
- A perforation factor is included to account for the fact that the gridshell is not a solid surface. This factor is calculated as the ratio between the total length of all the rods multiplied with the largest section diameter and the total surface area. The wind load is subsequently reduced by this factor.
- A triangulated mesh that covers the entire gridshell surface is created and the normal vectors at the vertices (which coincides with the intersection points of the rods) are defined

from a weighted average of their neighbouring mesh face normals. The projection of each normal vector onto the specified wind direction determines the direction (inwards pressure if the dot product is negative and outwards suction if the dot product is positive) and the magnitude such that a variation over the surface is obtained (if the dot product is 1.0 i.e. parallel normal and wind direction then the magnitude is 1.0 and if the dot product is 0 i.e. perpendicular normal and wind direction then the magnitude is 0). The result of these operations is a number of scaled vectors in the vertex normal directions.

- The voronoi area around each vertex is calculated from a hybrid method as described by Meyer et al. (2002) and multiplied with the reduced wind load value (due to the perforations) to obtain the lumped nodal force.
- Lastly, the lumped nodal force in each vertex is multiplied with the scaled normal vectors and the load case wind coefficient, which gives the wind distribution as illustrated in Figure 6.4.1.

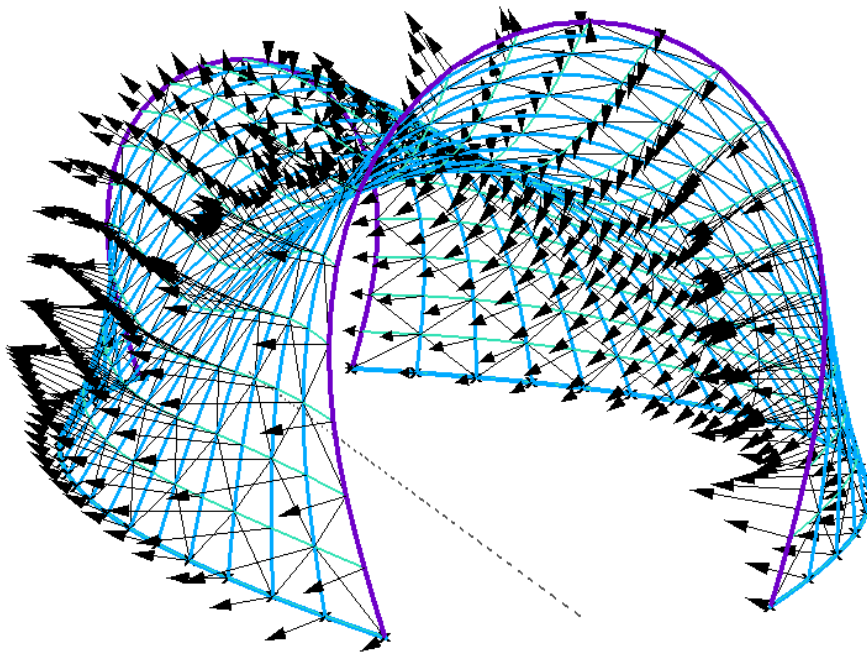


Figure 6.4.1 – The simplified wind load acting on the gridshell

6.4.1 Design 1

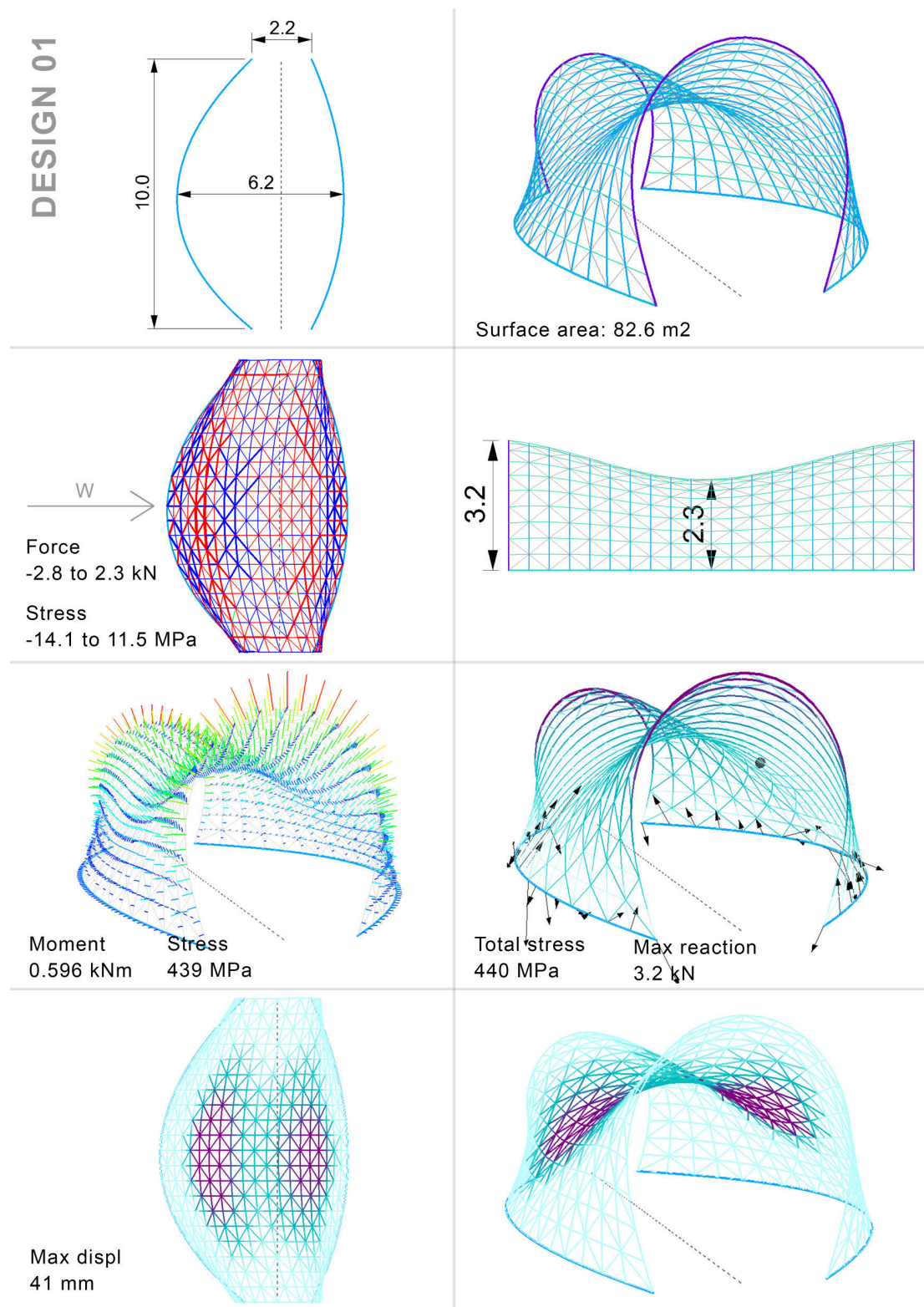


Figure 6.4.2 – Gridshell design 1: Geometry and structural performance

6.4.2 Design 2

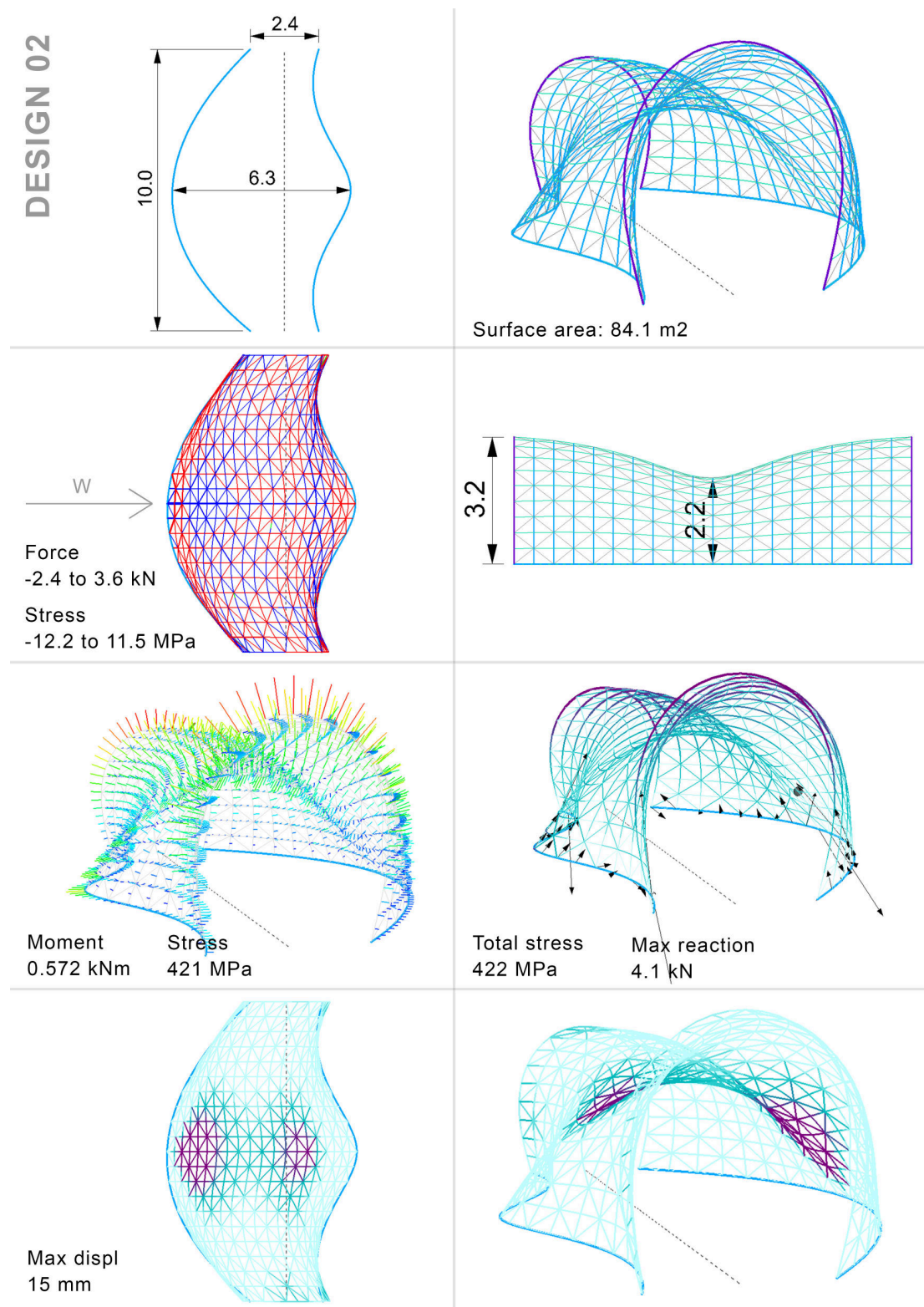


Figure 6.4.3 – Gridshell design 2: Geometry and structural performance

6.4.3 Design 3

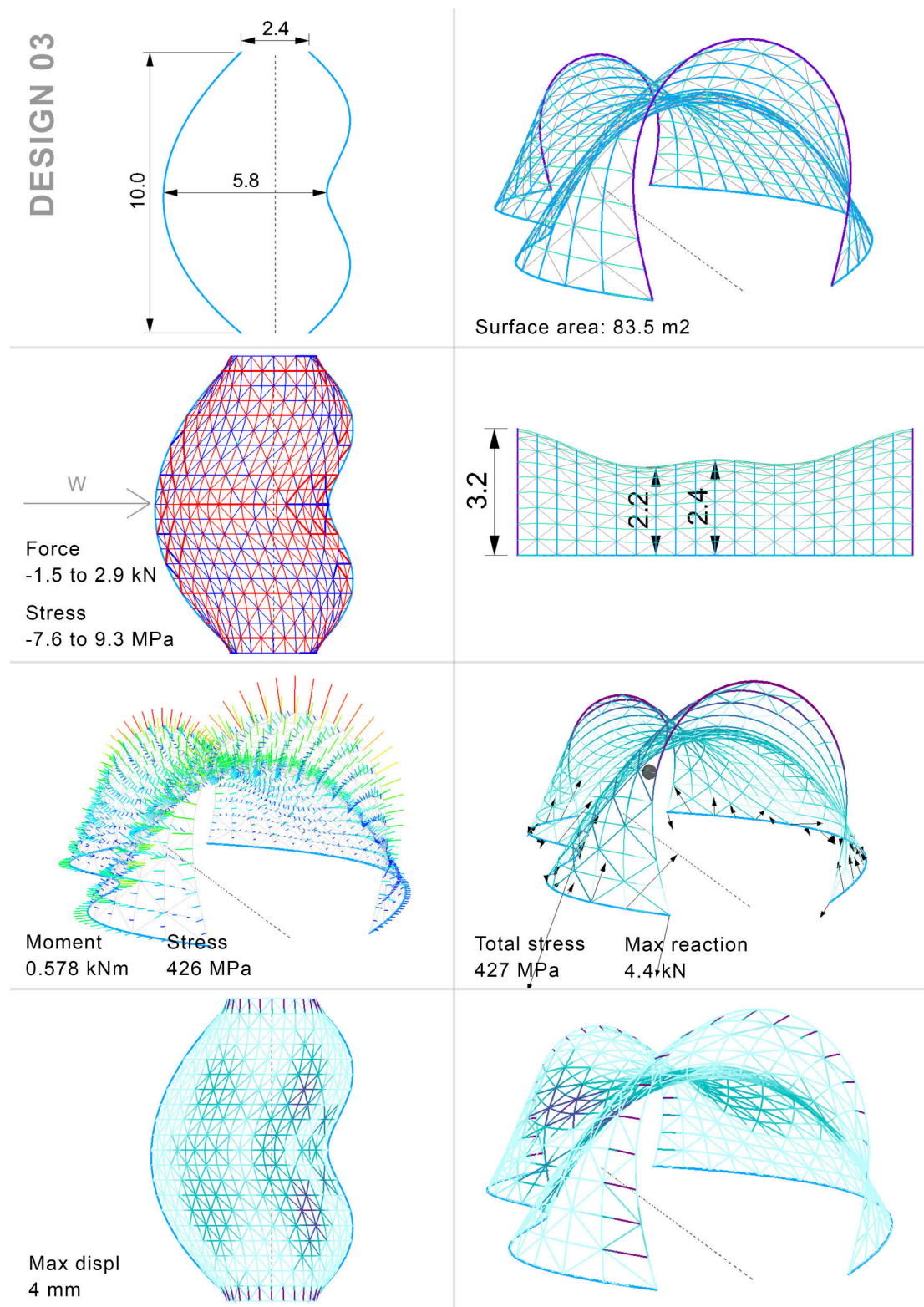


Figure 6.4.4 – Gridshell design 3: Geometry and structural performance

6.4.4 Design 4

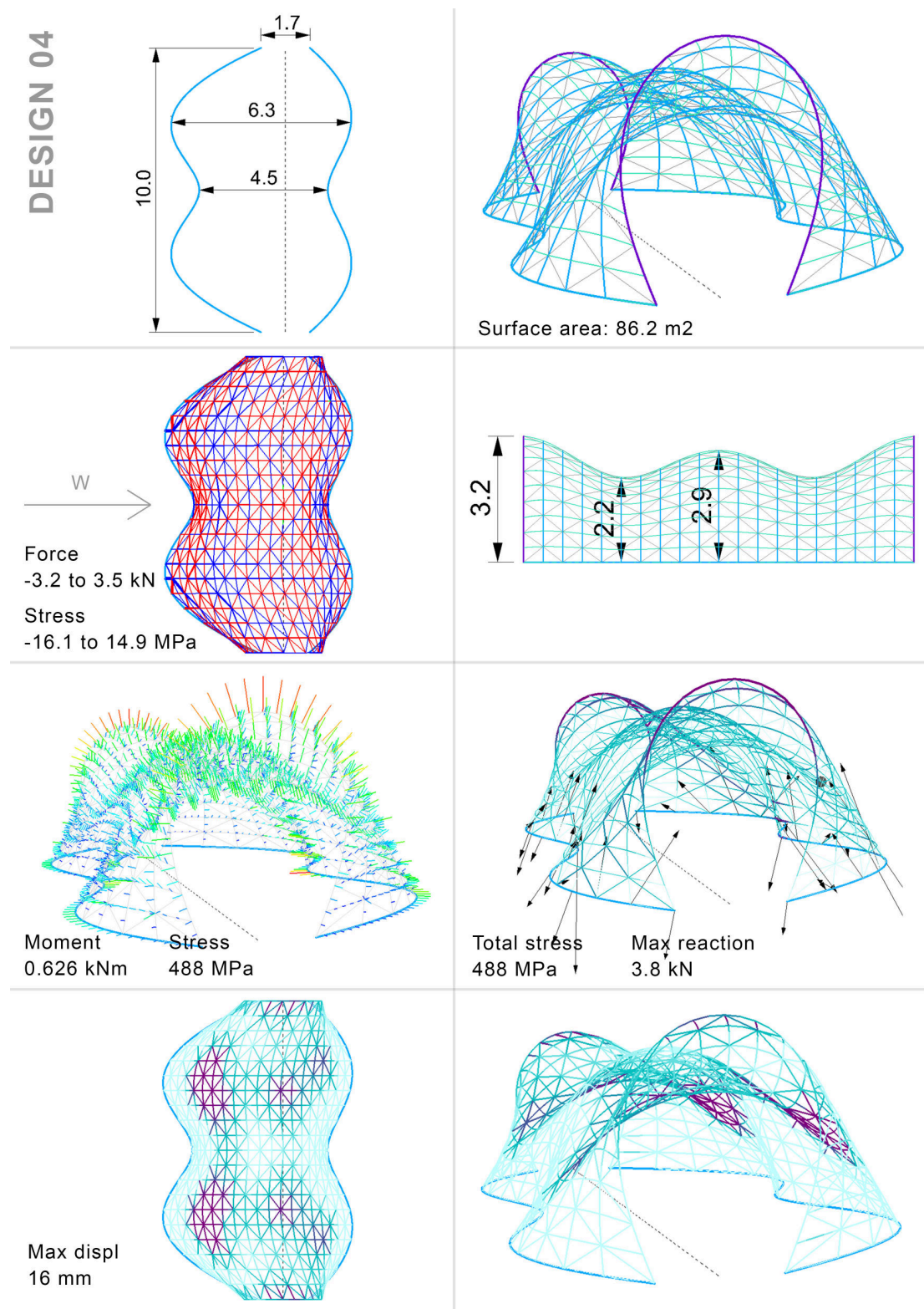


Figure 6.4.5 – Gridshell design 4: Geometry and structural performance

6.4.5 Design 5

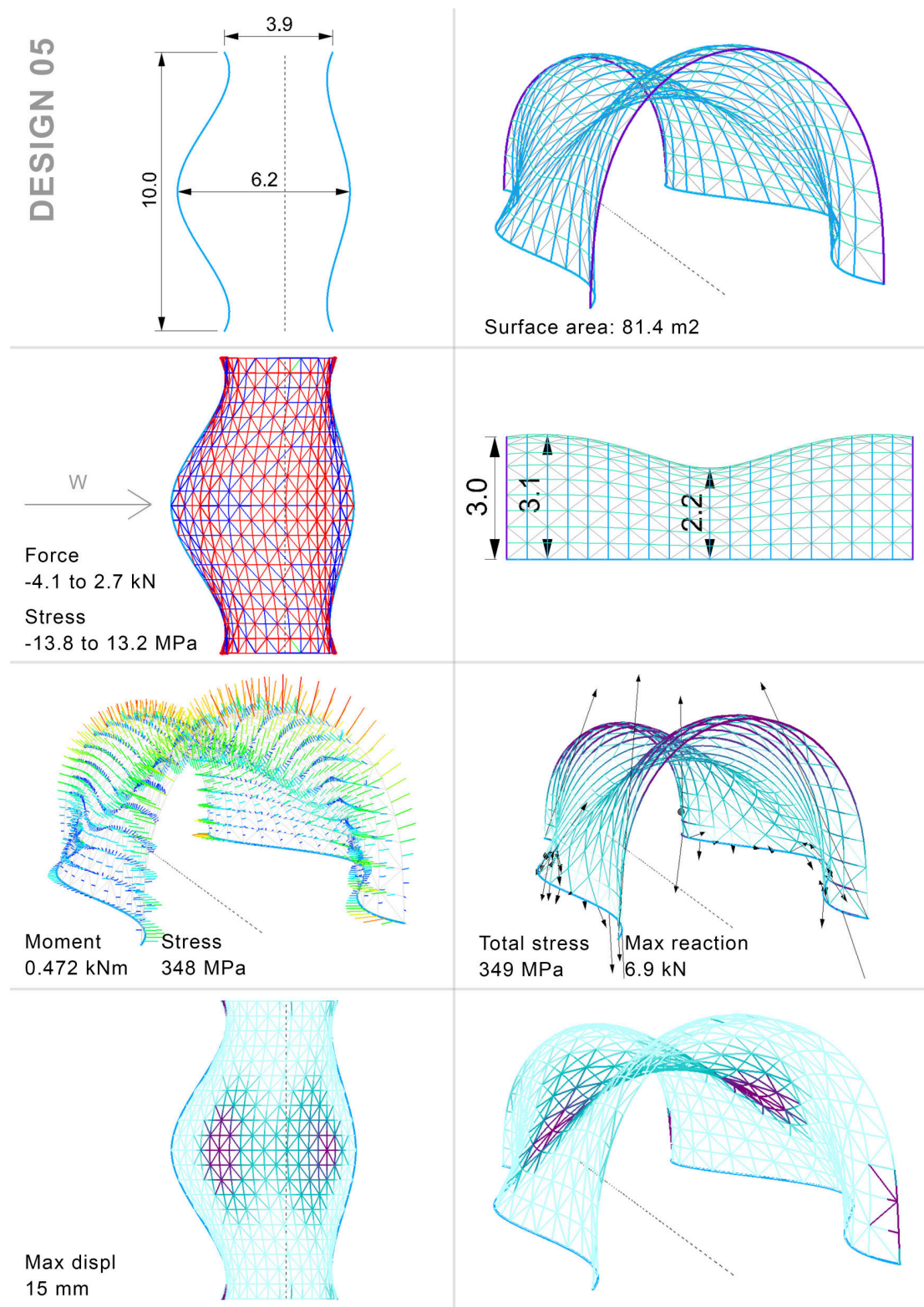


Figure 6.4.6 – Gridshell design 5: Geometry and structural performance

6.5 Design selection and refinement

The deflection is the most interesting of the structural performance parameters as the structure is very lightweight and flexible. It is furthermore observed that the maximum utilisation of the elements is approximately 0.5, which implies that the stresses are not a problem. Design 1 with the most simple boundary curves has the largest deflection of 41 mm whereas the other 4 design variations have a maximum deflection in the range between 4-16 mm. This significant reduction shows the importance of curvature as a means of stiffening the shell. Design 3 is particularly stiff and from a comparison with the three other design variations, it is observed that different frequencies of the boundary curves and a reduction of the width at the middle are beneficial design parameters to stiffen the shell.

When designing bending-active structures it is expected that the deflections are larger than in a normal structure. For a 10 m long gridshell it is evaluated that a maximum deflection of 50 mm is acceptable ($L/200$). All five design variations thus fulfil this requirement and the selection is therefore mainly influenced by the aesthetics of the pavilion. In this case, the most efficient design is also evaluated to be the most aesthetically pleasing design and hence Design 3 is selected for further refinements.

The first step is to reduce the diameters of the rods since the maximum deflection for this design is so small. The total weight is approximately halved compared to the initial design exploration with the rod diameters as shown in Table 6.5.1.

| | |
|-----------------------------------|--------|
| Free edge diameter | 16 mm |
| Primary diameter | 14 mm |
| Secondary/bracing diameter | 12 mm |
| Total weight | 161 kg |

Table 6.5.1 – Reduced diameters of the rods

The structural performance under three different load cases are subsequently investigated:

1. $1.0 \cdot SW$
2. $1.0 \cdot SW + 1.5 \cdot S + 0.45 \cdot W$
3. $1.0 \cdot SW + 1.5 \cdot W$

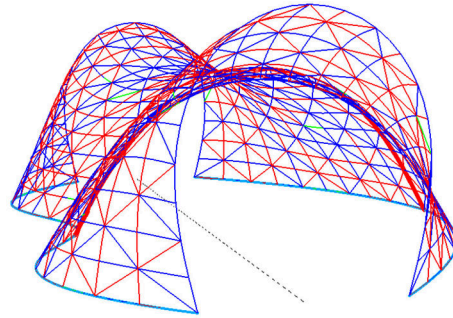
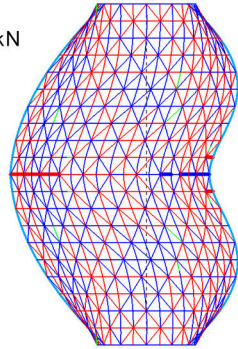
Here SW is the self-weight, S is the snow load and W is the wind load. All load cases include the prestress of the rods from initial bending. The perforation factor (as described previously) is now doubled to account for the additional area around the circular sections, which is influenced by the wind as well. The snow load is set to 0.9 kN/m^2 according to EN 1991-1-3 DK NA and multiplied with the horizontal voronoi area around each node in the mesh in order to calculate the lumped nodal forces. For simplicity, the same perforation factor is used.

6.5.1 Analysis results

LOAD CASE: 1.0SW

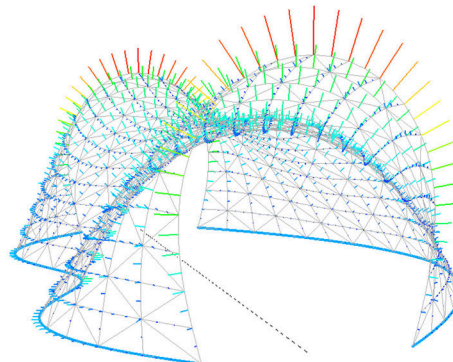
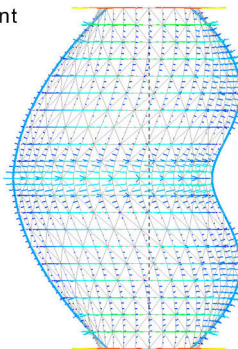
Axial force
-0.8 kN to 0.9 kN

Axial stress
-5 to 6 MPa



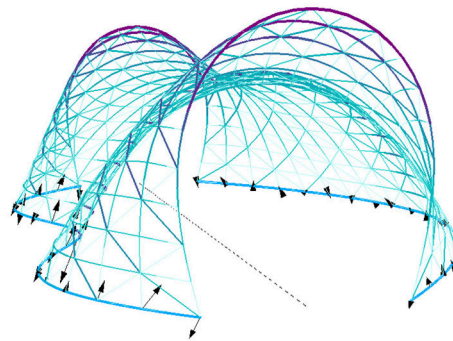
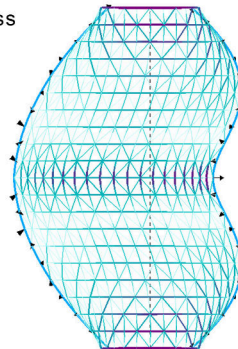
Bending moment
0.104 kNm

Bending stress
259 MPa



Max total stress
259 MPa

Max reaction
1.2 kN



Max displacement
6 mm

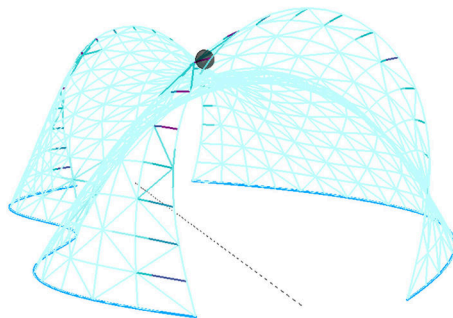
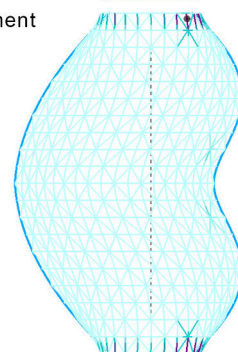
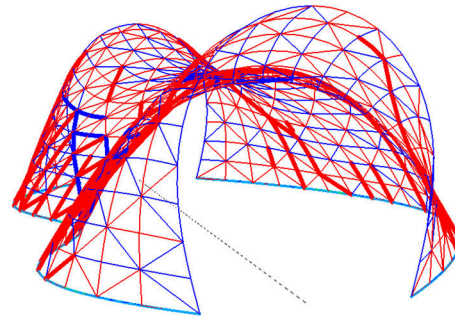
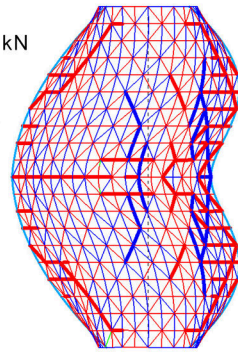


Figure 6.5.1 – Structural performance for load case 1

LOAD CASE: $1.0SW+0.45W+1.5S$

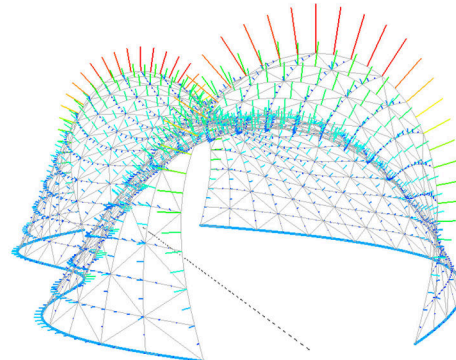
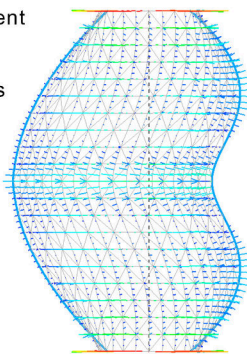
Axial force
-1.3 kN to 1.4 kN

Axial stress
-10 to 12 MPa



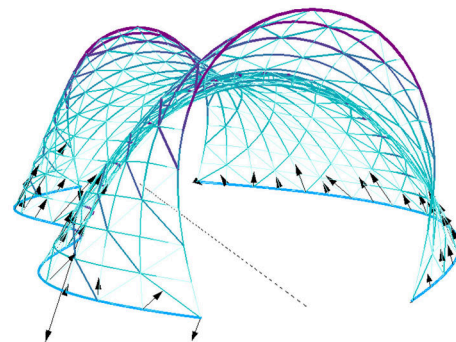
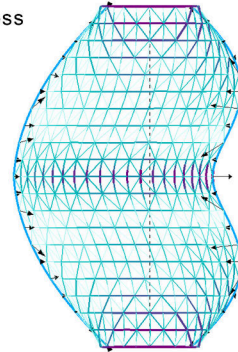
Bending moment
0.1 kNm

Bending stress
249 MPa



Max total stress
250 MPa

Max reaction
2.5 kN



Max displacement
19 mm

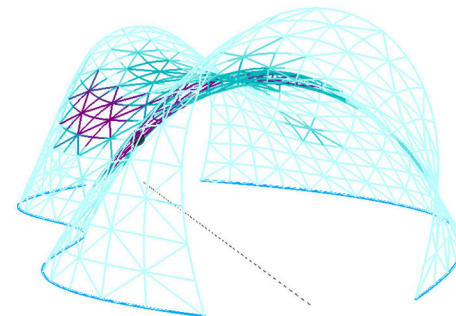
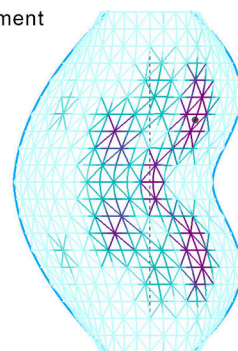
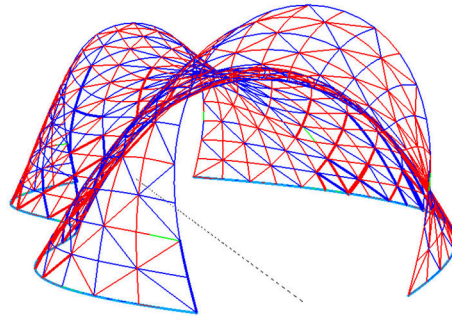
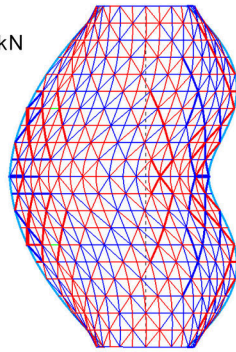


Figure 6.5.2 – Structural performance for load case 2

LOAD CASE: 1.0SW+1.5W

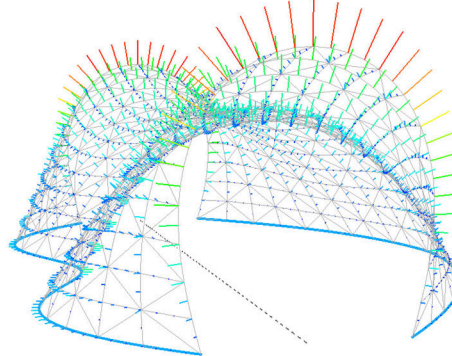
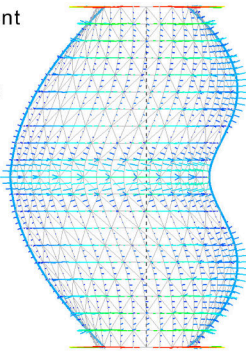
Axial force
-2.0 kN to 1.9 kN

Axial stress
-18 to 12 MPa



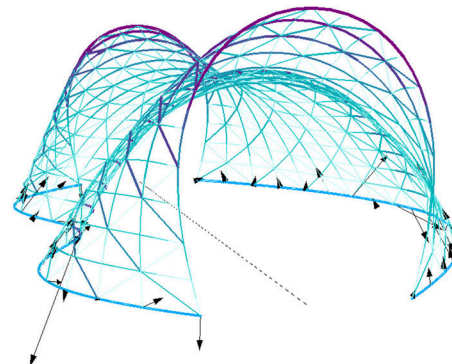
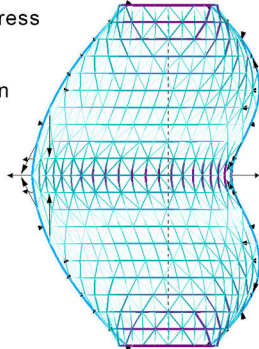
Bending moment
0.1 kNm

Bending stress
248 MPa



Max total stress
249 MPa

Max reaction
3.2 kN



Max displacement
17 mm

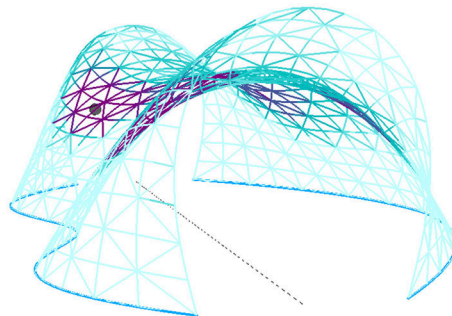
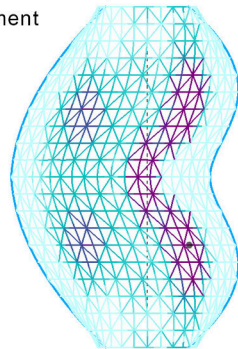


Figure 6.5.3 – Structural performance for load case 3

Load case 1 is useful to understand the behaviour from the initial prestress of the rods. Interestingly, the maximum stress level of all the load cases occurs here. This is because the snow load introduces bending in the opposite direction of the moment from prestress and the wind load vanishes along the top according to the chosen distribution i.e no suction to further increase the moment at the top. This highlights the significance of the initial prestress, which is the dominant contribution to the stress level. Baverel et al. (2012) describe the same behaviour for a built example of a GFRP gridshell (Soliday's Pavilion in Paris). The maximum bending stresses are located at the top of the free edges where the curvature is largest and a band with increased stresses across the middle is also noticed as a result of the curvature of one of the boundary curves. Whilst the primary curves are in equilibrium on their own, the displacements are caused by the added secondary and bracing rods, that also want to straighten out. The behaviour is most significant in the areas where the secondary rods meet the free edges. This may be a result of the connections being modelled as rigid everywhere except at the free edges where the connections are pinned. The movement at the middle of the gridshell is more restrained but the attempt of the secondary rods to straighten out is noticeable from the increased tension/compression forces at these locations.

Apart from the bending stresses, the other structural performance parameters increase for load case 2 and 3. The maximum displacement is approximately the same and occurs in the same curved band along the suction side of the gridshell. It is evident how the extra curvature at the middle helps to stiffen the shell and pushes the displacement band closer to the top. The difference between the two load cases is most clear from the two additional displacements "pockets", which for the snow load develop closer to the top and for the wind load develop closer to the boundary (both inwards). The high strength/stiffness ratio of the GFRP material means that the deflections become the critical parameter to design for. The maximum displacement value of 19 mm suggests that the diameters of the rods are further reduced or the spacing between the rods are increased. However, a reduction of only 2 mm of each rod's diameter increases the maximum deflection to approximately 65 mm. This is evaluated to be too much and buckling issues may occur before this point.

The largest reaction forces occur at the middle of the gridshell due to the suction from the wind. This means that the structure has to be properly tied down.

6.5.2 Buckling

A non-linear buckling analysis of the gridshell is performed for a load combination including pre-stress, self-weight and wind. Even though the stresses and deflections are within an acceptable range using factored loads, buckling may still be an issue. A load factor step size of 0.1 is used for the buckling simulation, which continues to run until a maximum deflection of 500 mm is reached. The deformed shape from each iteration is stored and subsequently analysed to detect when buckling occurs. Figure 6.5.4 and Figure 6.5.5 illustrate the deformed shape from a number of selected load factors ("LF"). Up until a load factor of 3.0 the displacements slowly increase with no sudden change for any of the elements. However, it is observed how a small displacement pocket (purple area) on the left side from the top view starts to develop and between LF=3.0 and LF=3.1 the maximum displacement suddenly changes from one side of the structure to the other due to local buckling of some elements. The buckling load factor is therefore estimated to a value of 3.1, which means that the structure is sufficiently stiff.

Sometimes these local buckling modes can be ignored if the deflections are within an acceptable range, the shape is still in a good condition and the structure finds a new stable configuration. Figure 6.5.5 illustrates the behaviour when the load increments continue. The structure finds a stable configuration again where the displacements slowly increase up until a load factor of 4.9. After this point, local buckling occurs first at another part in the same area as before (a bit further up) and shortly afterwards in a second displacement pocket.

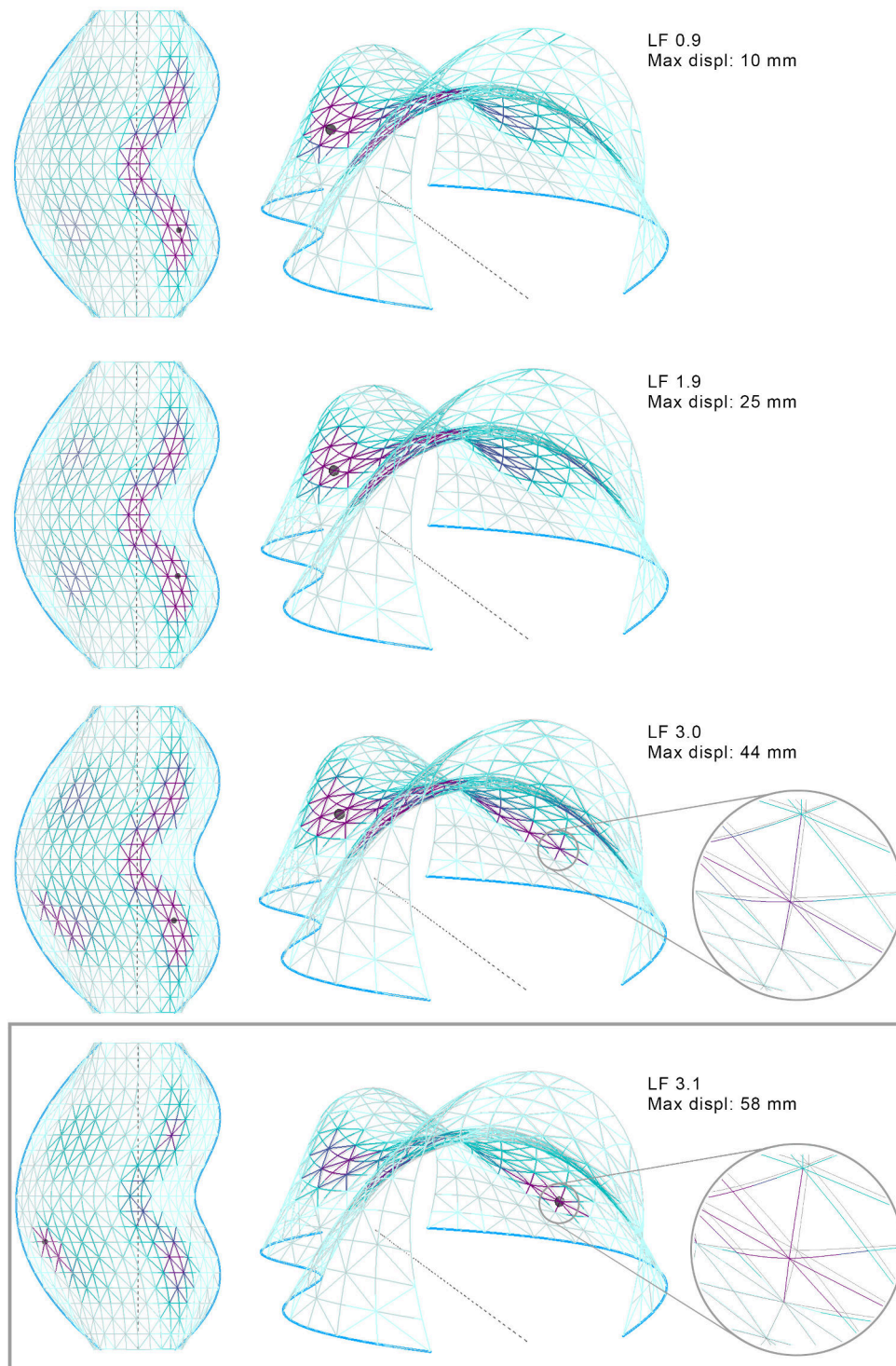


Figure 6.5.4 – Selected deformations of the gridshell during a buckling analysis (part I). “LF” is the load factor and the grey box indicates when buckling is detected

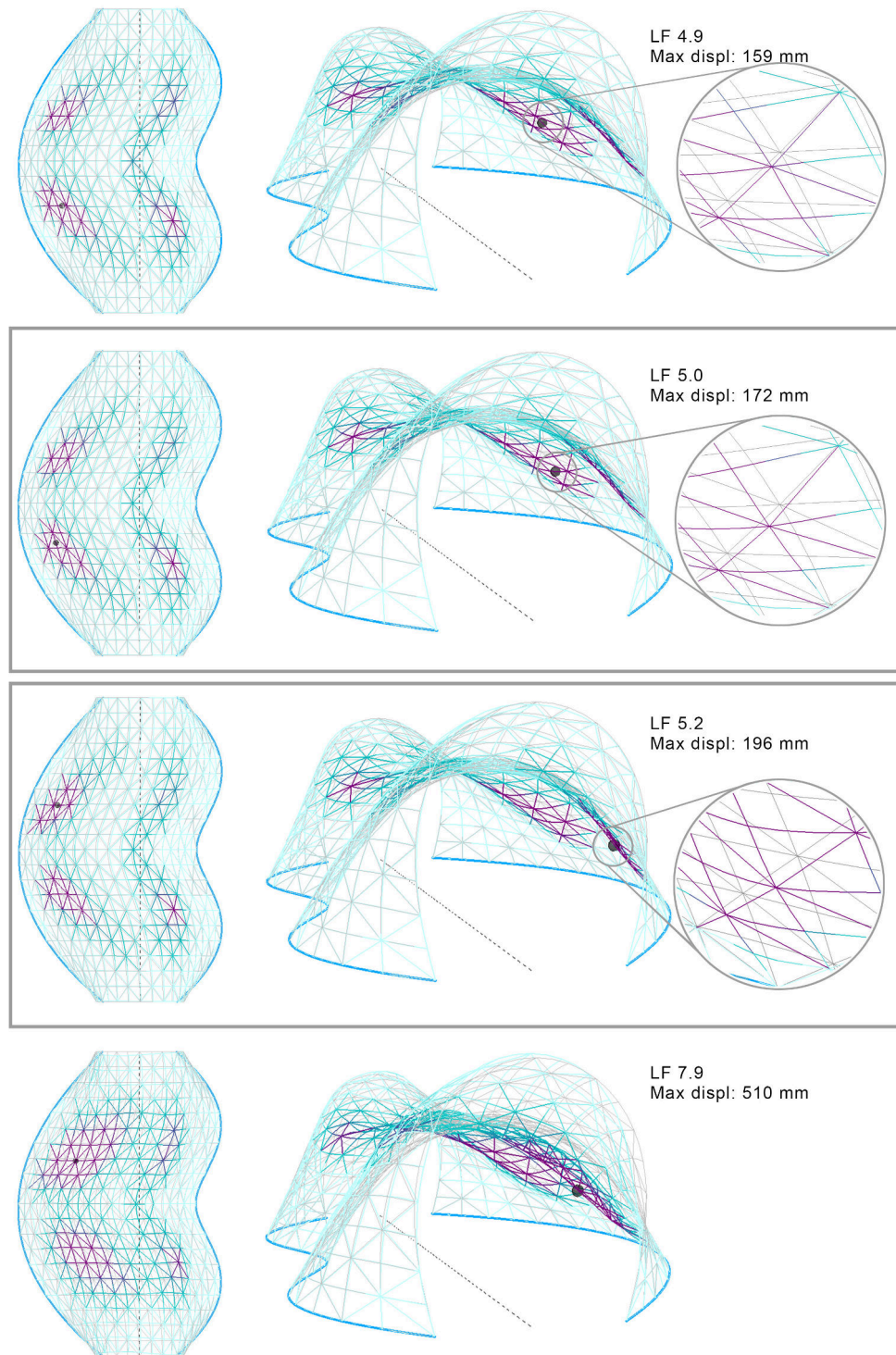


Figure 6.5.5 – Selected deformations of the gridshell during a buckling analysis (part II). “LF” is the load factor and the grey boxes indicate when buckling is detected

6.5.3 Connections and supports

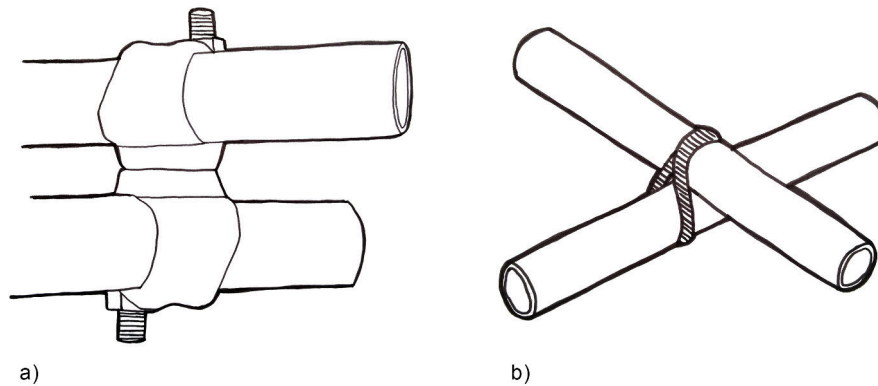


Figure 6.5.6 – Different options for the GFRP gridshell connection detail. a) Standard swivel scaffold connection. b) Wrap around connection of metal wire or fabric tape

For gridshells made from circular cross sections, the two most common connection types are a swivel scaffold connection or a wrap around solution (Quinn and Gengnagel, 2014) as shown in Figure 6.5.6. Both connectors allow rotation of the intersecting rods, which is the ideal property to enable sufficient freedom during the erection process. The bracing layer are then connected with a similar type of connection to the secondary layer. In this case study, the connections are analysed as rigid due to the simplified modelling technique. This implies that if the described connectors are used in practice, the gridshell becomes less stiff. Due to the small deflections from the analysis, this may be acceptable in this case.

The analysis shows that the supports have to be designed for an uplift force of maximum 3.2 kN corresponding to 320 kg. Figure 6.5.7, Figure 6.5.8 and Figure 6.5.9 illustrate different ideas for how this detail at the boundary may be designed. The challenge is that the primary rods meet the boundary with different angles. Option 1 (see Figure 6.5.7) is a curved edge beam made from two timber panels that meet at a right angle. The beam has triangular stiffeners along its length, which the primary rods are fastened to by simple brackets with appropriate inclinations. The edge beam is subsequently anchored to the ground. This solution gives a nice and clean expression from the outside but might be a bit tricky with regard to the bracing rods. Option 2 (see Figure 6.5.8) is based on the addition of an extra rod along the boundary of the gridshell. The connections between this rod and the primary rods are similar to the rest of the gridshell. This boundary rod is then fixed to a curved edge beam with simple brackets. The top part of the edge beam is split into segments in order to make space for the connectors between the intersecting rods. The edge beam is similarly anchored to the ground. Option 3 (see Figure 6.5.9) uses a metal sleeve with a flat plate welded to it, which is bolted to an angle bracket and fixed to a curved edge beam. However, the rod diameters are very small and it is questionable whether this solution is applicable at this scale.

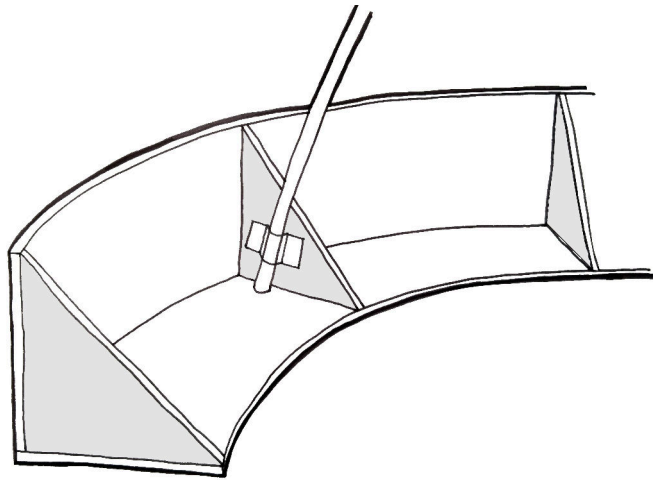


Figure 6.5.7 – Support option 1

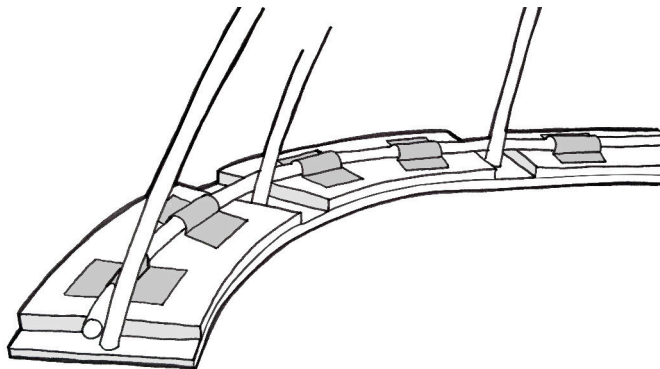


Figure 6.5.8 – Support option 2

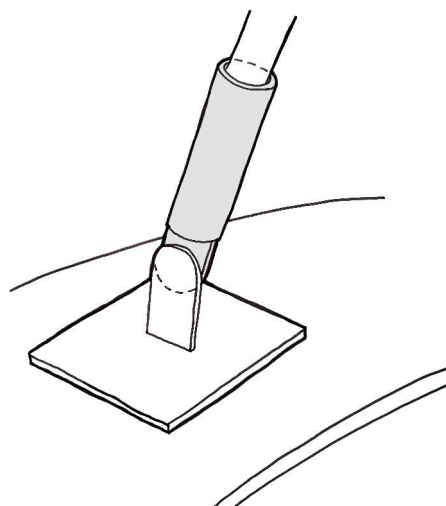


Figure 6.5.9 – Support option 3

6.6 Final design

The final design of the GFRP gridshell for a market stall at Torvehallerne is visualised in Figure 6.6.1.

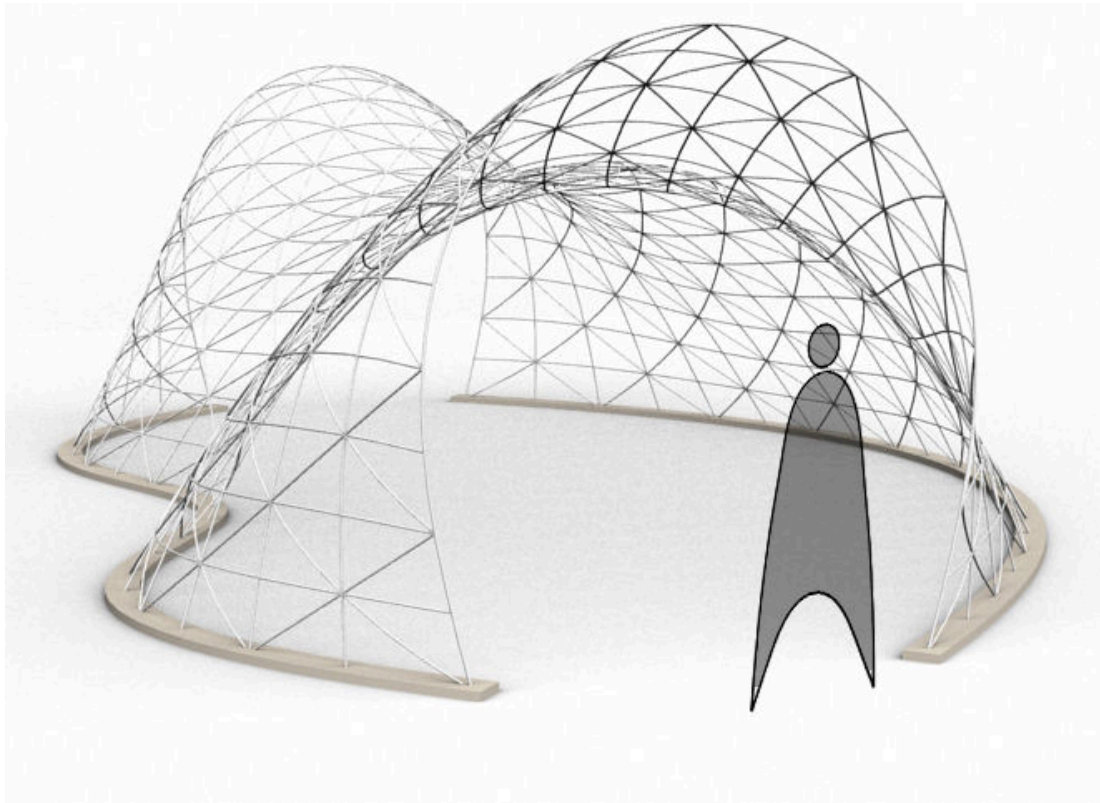


Figure 6.6.1 – The final gridshell design

The case study demonstrates how the developed tool helps to assist during the early design stage. Multiple configurations are easily explored and the visualisations help to build an intuitive understanding of the structural behaviour. The performance of the individual designs is quickly evaluated from the output of maximum axial forces, reaction forces, bending moments, total stresses and deflections. This helps to inform the design selection process and the subsequent analysis shows to which extent the design can be refined before a more detailed analysis with advanced finite element software is necessary for structural validation and design of connection details.

Chapter 7

Conclusions

7.1 Summary

The aim of this research was to improve the current workflow related to the design and analysis of form-active structures within the Rhino/Grasshopper environment. The literature and software review highlighted that the integrated finite element software Karamba had limited capabilities in this context due to the large deformations related to these structures. On the other hand, the Grasshopper plug-in Kangaroo2 showed great potential as it inherently dealt with these large deformations from its underlying dynamic relaxation solving technique and with improved stability such that real world material properties could be used. This made it the ideal platform to build this research upon and the ability to analyse form-active structures within the Rhino/Grasshopper environment was therefore accomplished by the development of a plug-in, which extended this Kangaroo2 framework with calibrated structural behaviour and output.

The implementation included a number of custom Kangaroo2 goals, which mimicked axial and bending behaviour based on a 3 DoF system in order to model cables, bars and beams. The output from the goals played an essential role for this development such that axial forces, shear forces and bending moments became accessible and hence made it possible to evaluate the structural performance. The output was also used to provide additional geometrical data and information about the system for visualisation purposes and stress summation. Comparisons with analytical solutions and results from Karamba (for appropriate kind of structures) showed excellent compliance and were an important factor to improve the reliability associated with the software such that the results for more complicated structures could be trusted.

The developed plug-in was tested during the workshop days at the Smart Geometry conference and integrated in a modelling pipeline to evaluate the structural performance of a 5 m tall hybrid tower of GFRP rods and plastic cables. The interactive environment with live feedback on the structural performance showed great potential for further exploration of form-active hybrid

structures in general. The case study of a GFRP gridshell also demonstrated the advantages of this workflow to quickly explore five design variations and use the structural performance output to guide the selection process. A subsequent refinement of the selected design illustrated the level of detail of the analysis, which helped to halve the material consumption.

7.2 Discussion and future work

The developed K2Engineering plug-in is only the early stage of a larger potential of analysing form-active structures based on the Kangaroo2 framework. The implementation so far only represents the simplest elements i.e. cables, bars, beams and simple supports in a 3 DoF system. Whilst this 3 DoF system reduces the complexity and increases the speed of the simulation it also means that e.g. modelling of bending behaviour necessitates a requirement for two consecutive line segments and is limited to a cross section with rotational symmetry. Similarly, it is only possible to model rigid supports by having two consecutive pinned supports. When working inside the boundary of those limitations, the tool has already proven useful as demonstrated by the Smart Geometry workshop and the gridshell case study. Breaking down the behaviour into simple goals also has a big value for educational purposes and helps to explain basic structural principles. However, most structures are not only built from tubular sections and hence require the consideration of e.g. biaxial bending and torsion as well. Membranes are also a key element in the design of form-active structures and whilst a cable-net abstraction can be used for the early stage design, it is desirable to develop a proper CST (constant strain triangle) element.

Methods have already been developed to simulate biaxial bending and torsion based on 4 degrees of freedom rather than the usual 6 and one of these can potentially be integrated with Kangaroo2. However, these methods become fairly complex and are naturally involved with some assumptions (like the rod goal) in order to achieve the same behaviour with less degrees of freedom. Instead of using these workarounds, a better approach is perhaps to implement the functionality in a 6 DoF system. According to Daniel Piker, it is only a matter of time before Kangaroo2 is extended to such a system, which supports the idea of that development path. The desire for more accuracy is however at the cost of computational speed and this balance questions the overall purpose of the structural calibration with Kangaroo2: Is the interactive level and ability to engage with the simulation a higher priority than the structural accuracy? The interactive level is what makes Kangaroo so powerful and is very useful in the early stage of the design development. On the other hand, the increased stability of the solver has allowed real world material properties to be used and with that opened up for the possibility to calibrate the simulation to accurate structural behaviour. Even with a decreased simulation speed from more advanced goals, it is still advantageous with an analysis that takes half an hour to perform compared to several hours in a standard finite element program. It thus seems like the more advanced goals still have a gap to fill out and if interactive speed is the priority then the more simple goals can be used.

7.3 Perspective

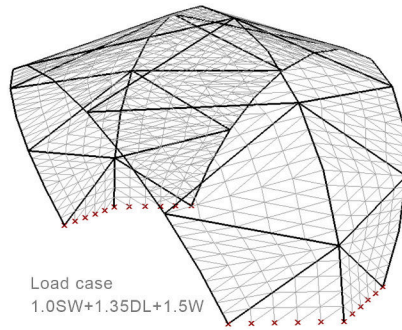
The developed K2Engineering plug-in has already proven to be useful for two projects at Format Engineers and this confirms that it fills out a gap in the existing software.

7.3.1 Clamshell Tent

The plug-in was used for the analysis of a tent-like structure made from tubular aluminium arch frames and a PTFE membrane as shown in Figure 7.3.1 (top). The arch frames were modelled as prefabricated curved elements with interconnected struts and the membrane as a cable-net. The purpose of the analysis was to investigate the effect of pretensioning the membrane on the deflections and bending moments. The structure was subjected to a load case that combined self-weight, dead load and wind load. The result from the analysis is illustrated in Figure 7.3.1. When the membrane was pretensioned, it had less slack areas (noticeable from the ratio between green and blue colours) and reduced the maximum deflection of the membrane significantly. At the same time, the axial forces in the membrane became larger and the bending moments and maximum deflection of the arch frames increased as well from the pulling. The problem with Karamba in this case was that the membrane could only be modelled as a shell working in both tension and compression and a reduction of the thickness caused buckling issues. The K2Engineering plug-in therefore proved to be very useful in order to model the structural behaviour of the Clamshell Tent and ensured that the right compromise were found before the final analysis were carried out in a more advanced finite element software.

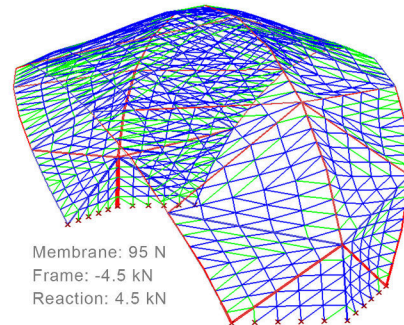
Chapter 7: Conclusions

INITIAL GEOMETRY

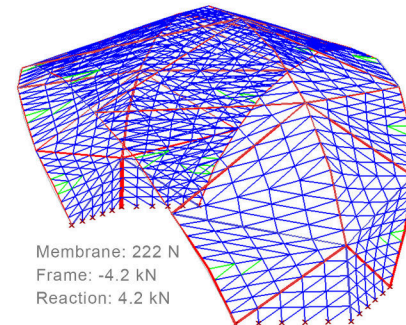


WITHOUT PRETENSION

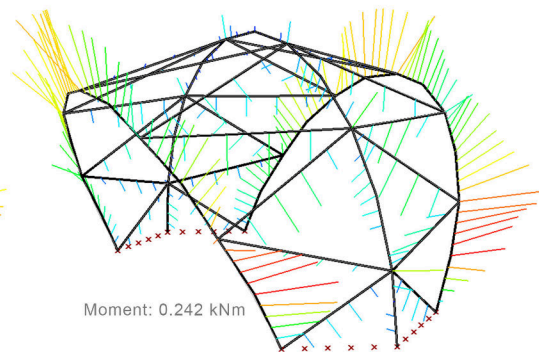
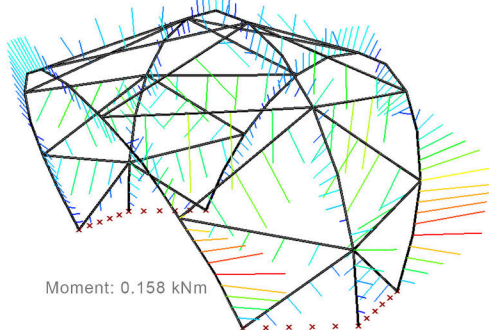
AXIAL FORCES



WITH PRETENSION (0.1 kN)



BENDING MOMENTS



DISPLACEMENTS

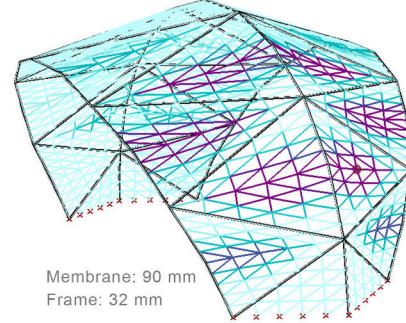
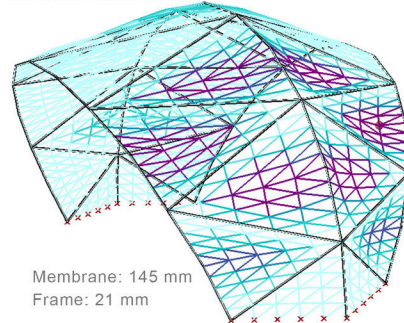


Figure 7.3.1 – Clamshell Tent analysis with K2Engineering. Client confidential

7.3.2 UWE Pavilion

A bending-active timber gridshell was designed for the graduate year show 2016 at the University of the West of England (UWE). The design was developed by the architecture students led by John Harding and with support from Format Engineers to validate the structural behaviour.

A Karamba model was initially set up to analyse the structure using the same approach from the OnGreening Pavilion (superimpose prestress from initial bending). However, the structure failed under Karamba's linear buckling analysis when subjected to wind load and it was unknown how critical the failure was and how much it would take to make it work. As a consequence, the overall shape of the pavilion was modified to incorporate more curvature at the boundaries as shown in Figure 7.3.2. This made the structure much stiffer but still not enough to avoid buckling failure according to Karamba and it was difficult to increase the thickness of the laths as that would cause failure from initial bending.

Therefore it was decided to analyse the structure with the K2Engineering software with an acknowledgement of the challenge related to the anisotropic cross section properties of the laths. Only 1/3 of the structure was considered due to the rotational symmetry and an assumption that the middle part would be very stiff. The issue of anisotropic cross section properties was addressed by the use of a weighted average of the moment of inertia about the weak and strong axis and the overall behaviour was compared to the Karamba model to ensure a reasonable coherence. The analysis helped to identify that the utilisation from prestress was too high for some laths close to the middle where the curvature radius was small (approximately 750 mm). The thickness of those laths was therefore reduced. Furthermore, a non-linear buckling analysis was carried out (referring to Figure 7.3.3), which showed a very ductile behaviour due to the prestress of the laths, no local buckling for load factors up to at least 1.5 and deflections within an acceptable range. The analysis also highlighted the vulnerable area close to the free edge, which as a consequence hereof was strengthened on site with additional laths. These observations helped to increase the confidence that buckling would not become an issue.

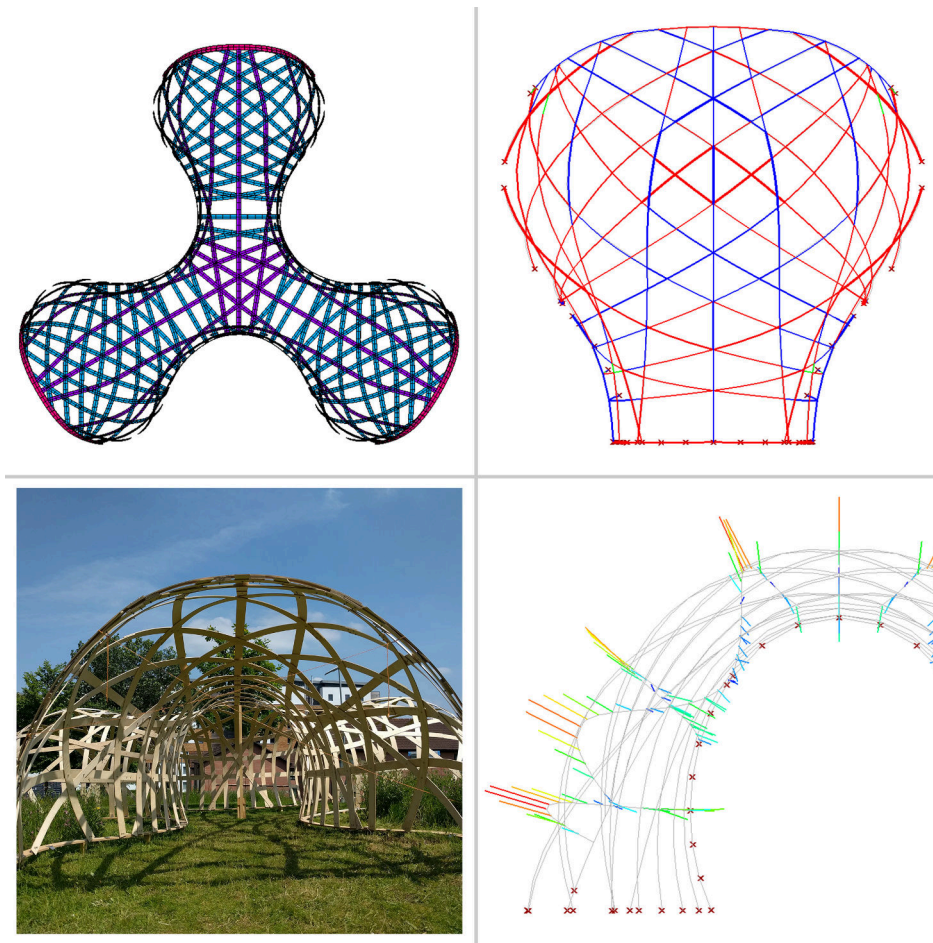


Figure 7.3.2 – Timber gridshell for the graduate year show 2016 at University of the West of England

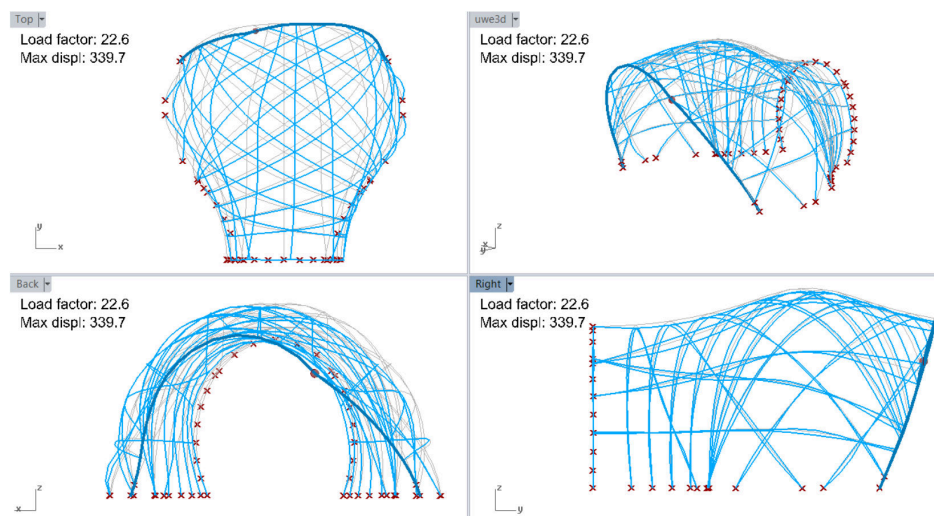


Figure 7.3.3 – Buckling analysis of the UWE gridshell with K2Engineering. The sphere highlights the critical area at the free edge

Bibliography

- Adriaenssens, S. and M. Barnes: 1999, ‘Tensegrity spline beam and grid shell structures’. *Engineering Structures* **23**, 29–36.
- Barnes, M.: 1999, ‘Form Finding and Analysis of Tension Structures by Dynamic Relaxation’. *International Journal of Space Structures* **14**(2).
- Barnes, M., S. Adriaenssens, and M. Krupka: 2013, ‘A novel torsion/bending element for dynamic relaxation modeling’. *Computers and Structures* **119**, 60–67.
- Baverel, O., J.-F. Caron, F. Tayeb, and L. Peloux: 2012, ‘Gridshells in Composite Materials: Construction of a 300 m² Forum for the Solidays Festival in Paris’. *Structural Engineering International* **3**.
- Block, P.: 2009, ‘Thrust Network Analysis - Exploring three-dimensional equilibrium’. Doctor of philosophy, Massachusetts Institute of Technology.
- Brandt-Olsen, C.: 2014, ‘Dynamic Relaxation [Online]’. Available from: www.ceciliebrandt.com/portfolio/dynamic-relaxation/ [Accessed 24 May 2016].
- Brandt-Olsen, C.: 2015, ‘Harmonic Form-Finding for the Design of Curvature-Stiffened Shells’. Master of philosophy, University of Bath.
- Brandt-Olsen, C.: 2016, ‘K2Engineering [Online]’. Available from: www.github.com/CecilieBrandt/K2Engineering [Accessed 24 May 2016].
- Cook, R. D.: 1995, *Finite element modeling for stress analysis*. John Wiley & Sons, Inc.
- Deleuran, A., G. Quinn, D. Piker, W. Pearson, C. Brandt-Olsen, D. Naicu, and H. Lewis: 2016, ‘Calibrated Modelling of Form-Active Hybrid Structures [Online]’. Available from: www.andersholdendeleuran.com/sg2016_HybridStructures.pdf [Accessed 24 May 2016].
- Engel, H.: 1997, *Structure systems*. Hatje Cantz Verlag.
- Fibrolux: 2016, ‘Reinforced plastics [Online]’. Available from: www.fibrolux.com [Accessed 24 May 2016].
- Grinspun, E., M. Bergou, M. Wardetzky, S. Robinson, and B. Audoly: 2008, ‘Discrete Elastic Rods’. *ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH 2008* **27**(63).
- Harding, J., W. Pearson, H. Lewis, and S. Melville: 2014, *Advances in Architectural Geometry 2014: The OnGreening Pavilion*. Springer.
- Karamba3D: 2016, ‘Karamba: About [Online]’. Available from: www.karamba3d.com/about [Accessed 29 April 2016].

BIBLIOGRAPHY

- Kroll, A.: 2011, 'AD Classics: Munich Olympic Stadium / Frei Otto and Gunther Behnisch [Online]'. Available from: www.archdaily.com/109136/ad-classics-munich-olympic-stadium-frei-otto-gunther-behnisch [Accessed 24 May 2016].
- Lienhard, J., H. Alpermann, C. Gengnagel, and J. Knippers: 2013, 'Active Bending, A Review on Structures where Bending is used as a Self-Formation Process'. *International Journal of Space Structures* **28**(3), 187–196.
- Loop.pH: 2014, 'Atmeture [Online]'. Available from: www.loop.ph/portfolio/atmeture/ [Accessed 4 May 2016].
- Malek, S. R.: 2012, 'The Effect of Geometry and Topology on the Mechanics of Grid Shells'. Doctor of philosophy in the field of structures and materials, Massachusetts Institute of Technology.
- McNeel: 2016a, 'Grasshopper - Algorithmic Modeling for Rhino [Online]'. Available from: www.grasshopper3d.com [Accessed 29 April 2016].
- McNeel: 2016b, 'Rhinoceros 5 Features [Online]'. Available from: www.rhino3d.com/features [Accessed 29 April 2016].
- Melville, S. and K. P. Nielsen: 2014, 'Archilace Letchworth - Structural Engineering Conceptual Design Report'. Technical report, Ramboll Computational Design.
- Menges, A.: 2010, 'ICD/ITKE Research Pavilion 2010 [Online]'. Available from: www.icd.uni-stuttgart.de/?p=4458 [Accessed 24 May 2016].
- Meyer, M., M. Desbrun, P. Schr, and A. H. Barr: 2002, 'Discrete Differential Geometry Operators for Triangulated 2 Manifolds'. *Visualization and Mathematics III* pp. 113–134.
- Murs, M.: 2014, 'Chevres en equilibre - goats balancing on a flexible steel ribbon [Online]'. Available from: www.youtube.com/watch?v=58-atNakMWw&noredirect=1 [Accessed 24 May 2016].
- Naicu, D., R. Harris, and C. Williams: 2014, 'Timber Gridshells: Design methods and their application to a temporary pavilion'. *World Conference on Timber Engineering (WCTE)*.
- Olsson, J.: 2012, 'Form finding and size optimization: Implementation of beam elements and size optimization in real time form finding using dynamic relaxation'. Master's thesis, Chalmers University of Technology.
- OSI: 2004, 'Apache License, Version 2.0 [Online]'. Available from: www.opensource.org/licenses/Apache-2.0 [Accessed 24 May 2016].
- Peloux, L., B. Lefevre, O. Baverel, and J.-F. Caron: 2015, 'Formulation of a 4-DOF torsion/bending element for the formfinding of elastic gridshells'. *Proceedings of the International Association for Shell and Spatial Structures Symposium 2015*.
- Piker, D.: 2015a, 'A "pull point to point" goal in Kangaroo 2 [Online]'. Available from: www.grasshopper3d.com/group/kangaroo/forum/topics/a-goal-that-acts-like-a-springs-but-gives-more-strength-to-one-of [Accessed 15 April 2016].
- Piker, D.: 2015b, 'Kangaroo 2.0 - Now out of the pouch and available for testing! [Online]'. Available from: www.grasshopper3d.com/profiles/blogs/kangaroo-2-0-now-out-of-the-pouch-and-available-for-testing [Accessed 10 March 2016].
- Piker, D.: 2015c, 'White paper: Goal-Driven Design'. Technical report, Robert McNeel & Associates.

BIBLIOGRAPHY

- Piker, D.: 2016a, ‘K2Goals [Online]’. Available from: www.github.com/Dan-Piker/K2Goals [Accessed 24 May 2016].
- Piker, D.: 2016b, ‘Kangaroo3d [Online]’. Available from: www.kangaroo3d.com/ [Accessed 24 May 2016].
- Piker, D. and W. Pearson: 2013, ‘Plankton [Online]’. Available from: www.github.com/meshmash/Plankton [Accessed 24 May 2016].
- Popov, E. V.: 2002, ‘Geometric Approach to Chebyshev Net Generation Along an Arbitrary Surface Represented by NURBS’. Nizhny Novgorod, Russia, International Conference Graphicon.
- Preisinger, C.: 2012, ‘Beam deflection stresses [Online]’. Available from: www.grasshopper3d.com/forum/topics/beam-deflection-stresses [Accessed 24 May 2016].
- Quinn, G.: 2016, ‘Projection board first test [Online]’. Available from: www.youtube.com/watch?v=t0ADPGJygr4&feature=em-upload_owner [Accessed 24 May 2016].
- Quinn, G. and C. Gengnagel: 2014, ‘A review of elastic grid shells, their erection methods and the potential use of pneumatic formwork’. *WIT Transactions on The Built Environment* **136**.
- SmartGeometry: 2016, ‘SG 2016 Hybrid Domains [Online]’. Available from: www.smartgeometry.org [Accessed 24 May 2016].
- SOFiSTiK: 2016, ‘SOFiSTiK [Online]’. Available from: www.sofistik.com/en/sofistik/ [Accessed 24 May 2016].
- Thomsen, M. R., M. Tamke, A. H. Deleuran, I. Tinning, H. Evers, C. Gengnagel, and M. Schmeck: 2015, *Modelling Behaviour: Hybrid Tower, Designing Soft Structures*. Springer.
- Umetani, N., R. Schmidt, and J. Stam: 2014, ‘Position-based Elastic Rods’. *Eurographics/ACM SIGGRAPH Symposium on Computer Animation* pp. 1–10.
- Weinstein, E. W.: 2016, ‘Runge-Kutta Method [Online]’. Available from: www.mathworld.wolfram.com/Runge-KuttaMethod.html [Accessed 27 May 2016].
- Wikipedia: 2016, ‘Directed acyclic graph [Online]’. Available from: www.en.wikipedia.org/wiki/Directed_acyclic_graph [Accessed 3 June 2016].
- Williams, C., S. Adriaenssens, M. Barnes, and R. Harris: 2014, *Shell structures for architecture: Dynamic relaxation*. Routledge.

BIBLIOGRAPHY

Appendix A

Structural validation

This appendix provides additional examples that serve to increase the reliability of the developed K2Engineering plug-in. In each test case, the results from the K2 analysis are compared with either analytical solutions or results obtained from the finite element program Karamba. The displacements are kept small to make the comparisons fair.

A.1 Axial behaviour

A.1.1 Tension

A steel cable is pinned in one end and subjected to an outwards force in the other end. Table A.1.1 shows the different properties for the set-up.

| Length | Segments | Diameter | Area | Young's modulus | Load |
|--------|----------|----------|----------------------|------------------------------|--------|
| 1.0 m | 10 | 10 mm | 78.54 mm^2 | $2.1 \cdot 10^5 \text{ MPa}$ | 500 kN |

Table A.1.1 – Properties used for the tension example

ANALYTICAL SOLUTION

The tension force in the cable is constant and equal to the applied load of 500 kN. The reaction force also has the same magnitude but in the opposite direction of the load. The extension of the cable can be calculated from Hooke's Law (as described in Chapter 4)

$$x = \frac{500 \cdot 10^3 \cdot 1000}{2.1 \cdot 10^5 \cdot 78.54} = 30.3 \text{ mm}$$

K2ENGINEERING RESULTS

The result obtained from the K2Engineering analysis is shown in Figure A.1.1 and is consistent with the analytical solution.

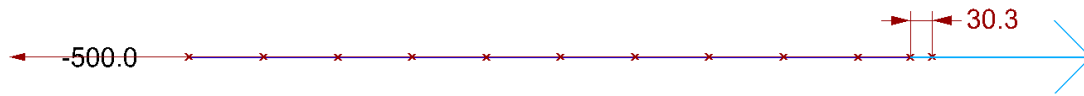


Figure A.1.1 – K2Engineering tension example. Reaction force in [kN] and deflection in [mm].

A.1.2 Pretension

A cable is pinned in both ends and pretensioned. No additional loads are applied. Table A.1.2 shows the properties that are used for the set-up. A constant tension force in each element equal to the specified pretension is expected as well as outwards pointing reaction forces (since the cable tries to shrink) of the same magnitude.

| Length | Segments | Diameter | Area | Young's modulus | Pretension |
|--------|----------|----------|----------------------|------------------------------|------------|
| 1.0 m | 10 | 10 mm | 78.54 mm^2 | $2.1 \cdot 10^5 \text{ MPa}$ | 200 kN |

Table A.1.2 – Properties used for the pretension example

K2ENGINEERING

The result from the K2Engineering analysis is shown in Figure A.1.2. This is in accordance with the expected behaviour as described above.

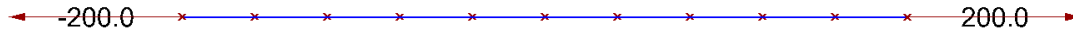


Figure A.1.2 – K2Engineering pretension example. Reaction forces in [kN].

A.1.3 Catenary

A catenary shape with 1.0 m between its end points is used as initial geometry for a steel cable that is subjected to a point load in each node. The following properties are used:

| Diameter | Area | Young's modulus | Point load |
|----------|----------------------|------------------------------|------------|
| 10 mm | 78.54 mm^2 | $2.1 \cdot 10^5 \text{ MPa}$ | 100 kN |

Table A.1.3 – Properties used for the catenary example

K2ENGINEERING

The axial- and reaction forces from the K2Engineering output are shown in Figure A.1.3. The maximum displacement at the middle is 21.3 mm.

ANALYTICAL

The forces are also calculated from graphic statics methods in order to validate a correct behaviour. The deformed shape from the analysis above (blue line in Figure A.1.3) is used as form

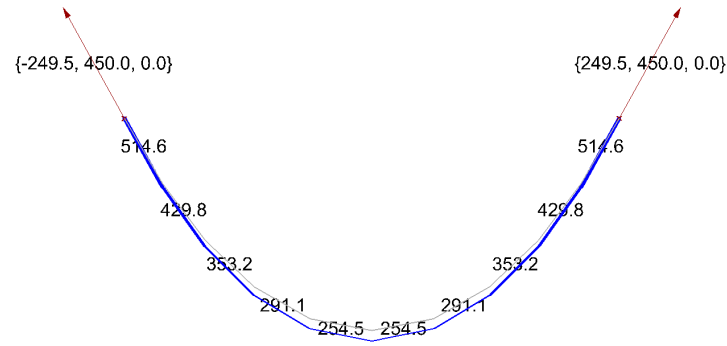


Figure A.1.3 – Axial- and reaction forces in [kN] for the catenary obtained from K2Engineering

diagram and the corresponding force diagram is drawn from the size of the applied load as seen in Figure A.1.4. The necessary forces in the bars to ensure equilibrium are extracted from the length of the lines in the force diagram and it is observed that they match the values from the K2Engineering plug-in. The reaction forces are extracted from the force diagram as well (the horizontal reaction force component is equal to the width and the vertical component to half of the height) and similarly show a nice correspondence.

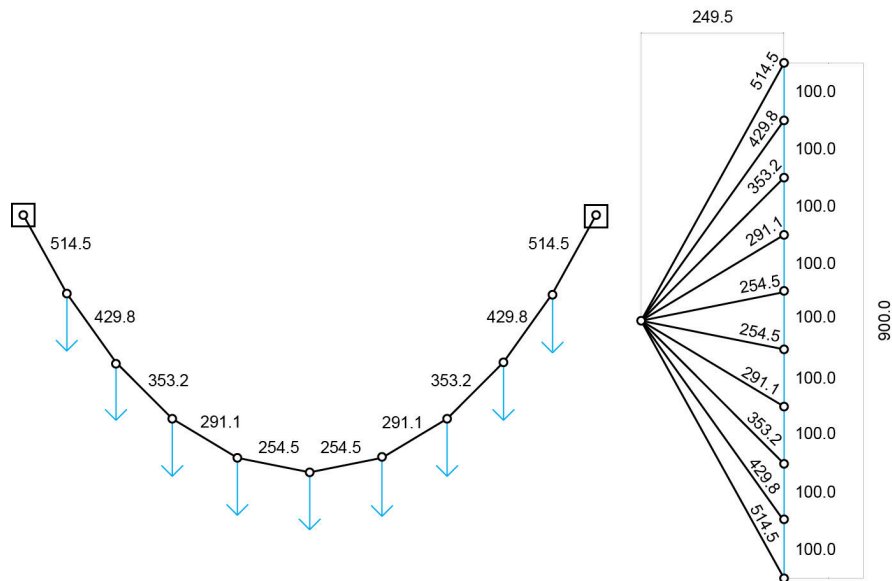


Figure A.1.4 – Analytical solution for the catenary using graphic statics (left: form, right: force polygon). The forces are in [kN].

A.1.4 Catenary with prestress

The same catenary shape and properties from Table A.1.3 are used in this example but in this case the cable is pretensioned with a force of $P = 200 \text{ kN}$. The pretensioning is expected to increase the axial- and reaction forces but reduce the deflection.

K2ENGINEERING

The result from the K2Engineering plug-in is shown in Figure A.1.5. The maximum displacement at the middle is 10.4 mm.

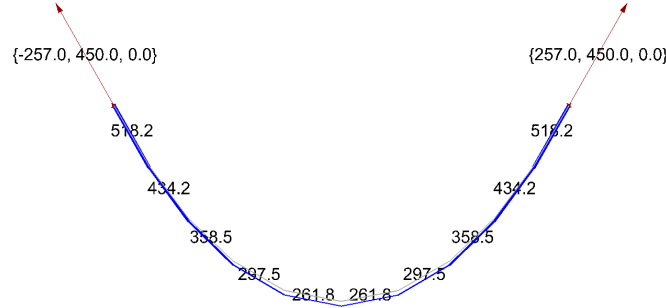


Figure A.1.5 – Axial- and reaction forces in [kN] for the prestressed catenary obtained from K2Engineering

KARAMBA

This is compared to the results obtained from a Karamba analysis using the same material properties, point loads and a pretension load. Karamba specifies the pretension load as a strain value [mm/m] for each element, which in this case is equal to (referring to Equation 4.3.1)

$$\epsilon = \frac{200 \cdot 10^3 \cdot 1 \cdot 10^3}{2.1 \cdot 10^5 \cdot 78.54} = 12.126 \text{ mm/m}$$

The result from a second order analysis in Karamba is shown in Figure A.1.6 and the maximum displacement at the middle is 10.4 mm. Again it is observed that the results from the two analysis are consistent and in accordance with the expected behaviour from prestress.

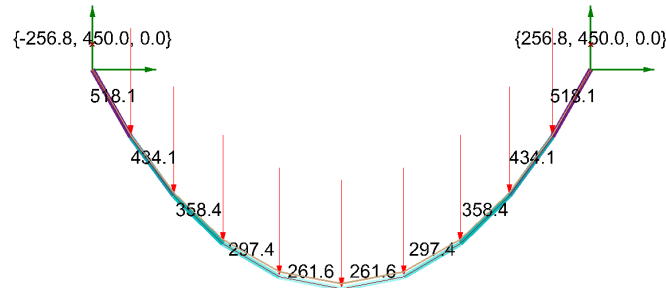


Figure A.1.6 – Axial and reaction forces in [kN] for the prestressed catenary obtained from Karamba

A.1.5 Truss

A truss of height 0.5 m and a span of 5.0 m is simply supported at its ends and subjected to its self-weight and a point load at the middle. Table A.1.4 shows the properties that are used for the set-up.

| Span | Cross section | Area | Young's modulus | Point load | Density |
|-------|---------------|---------------------|---------------------------|------------|------------------------|
| 5.0 m | 50 x 3 mm | 443 mm ² | 2.1 · 10 ⁵ MPa | 100 kN | 7850 kg/m ³ |

Table A.1.4 – Properties used for the truss example

K2ENGINEERING

The axial forces and reaction forces from the K2Engineering analysis are shown in Figure A.1.7. The maximum deflection at the middle is 28.8 mm.

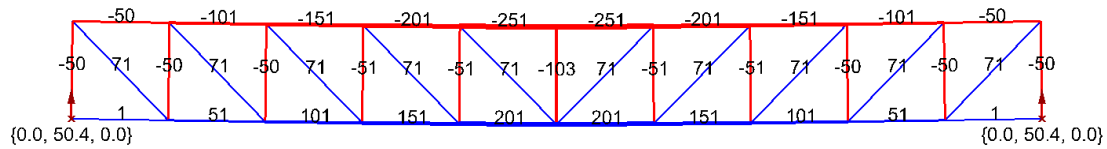


Figure A.1.7 – K2Engineering truss example. Axial and reaction forces in [kN]. Red = compression and blue = tension

KARAMBA

The forces are similarly shown for an analysis performed with Karamba as seen in Figure A.1.8. The maximum deflection at the middle is 28.9 mm.

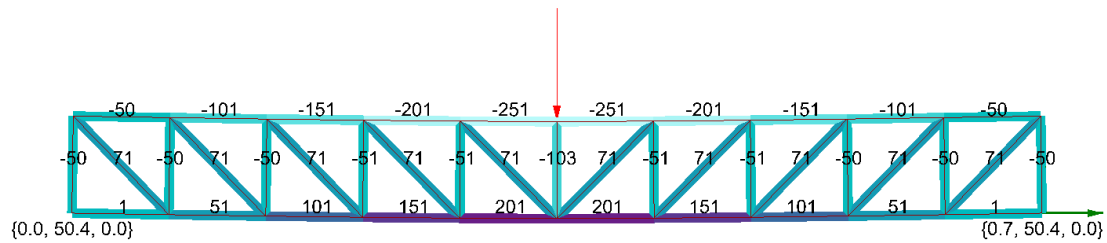


Figure A.1.8 – Karamba truss example where the colours express the utilisation of the elements. The axial and reaction forces are in [kN]

The same result is obtained from the two different tools.

A.2 Bending behaviour

A.2.1 Simple supported beam

A 1.0 m beam is simple supported and subjected to a uniform distributed load. Table A.2.1 shows the properties that are used for the set-up.

| Length | Segment length | Profile | Young's modulus | UDL |
|--------|----------------|---------|---------------------------|-----------|
| 1.0 m | 0.05 m | CHS 50 | 2.1 · 10 ⁵ MPa | 15.0 kN/m |

Table A.2.1 – Properties used for the simple supported beam example

K2ENGINEERING

A moment and shear plot of the simple supported beam from the K2Engineering analysis are shown in Figure A.2.1 and Figure A.2.2. The maximum displacement at the middle is 3.6 mm.

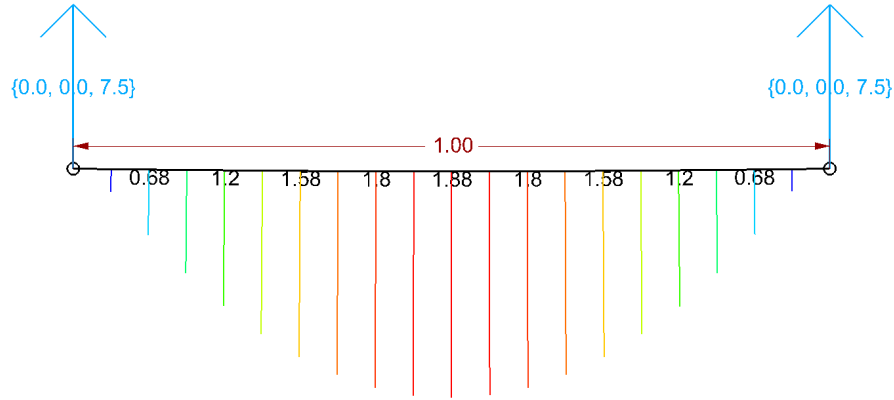


Figure A.2.1 – Simple supported beam with moment plot from K2Engineering. The moments are given in [kNm], the length measurement in [m] and reaction forces in [kN]

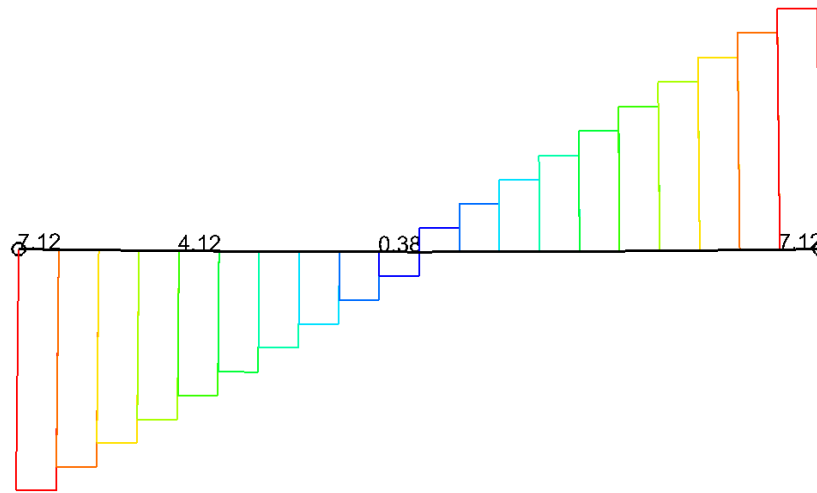


Figure A.2.2 – Simple supported beam with shear plot from K2Engineering. The shear forces are given in [kN]

ANALYTICAL

The moment distribution along the beam is calculated by hand from $M(x) = \frac{1}{2} \cdot q \cdot x \cdot (L - x)$, where q is the uniform distributed load, L is the beam length and x is the distance along the beam.

The moments along the beam from the left to the middle are as follows:

$$M(0) = 0 \text{ kNm}$$

$$M(100) = \frac{1}{2} \cdot 15 \cdot 100 \cdot (1000 - 100) = 0.675 \text{ kNm}$$

$$M(200) = \frac{1}{2} \cdot 15 \cdot 200 \cdot (1000 - 200) = 1.20 \text{ kNm}$$

$$M(300) = \frac{1}{2} \cdot 15 \cdot 300 \cdot (1000 - 300) = 1.575 \text{ kNm}$$

$$M(400) = \frac{1}{2} \cdot 15 \cdot 400 \cdot (1000 - 400) = 1.80 \text{ kNm}$$

$$M(500) = \frac{1}{2} \cdot 15 \cdot 500 \cdot (1000 - 500) = 1.875 \text{ kNm}$$

The vertical reaction forces: $R = \frac{1}{2} \cdot q \cdot L = \frac{1}{2} \cdot 15 \cdot 1 = 7.5 \text{ kN}$

The shear force is calculated from $V(x) = R - q \cdot x$, where x is measured to the midpoint of a segment. This gives the following:

$$V(0.025) = 7.5 - 15 \cdot 0.025 = 7.13 \text{ kN}$$

$$V(0.225) = 7.5 - 15 \cdot 0.225 = 4.13 \text{ kN}$$

$$V(0.475) = 7.5 - 15 \cdot 0.475 = 0.38 \text{ kN}$$

The maximum deflection is calculated from

$$u_{max} = \frac{5}{384} \cdot \frac{q \cdot L^4}{E \cdot I} = \frac{5}{384} \cdot \frac{15 \cdot 1000^4}{2.1 \cdot 10^5 \cdot 0.262 \cdot 10^6} = 3.5 \text{ mm}$$

It is observed that there is a good correspondence between the results obtained from the K2Engineering analysis and the analytical solution.

A.2.2 Cantilever beam

A 1.0 m beam cantilevers from one end. At this location, two consecutive points are pinned in order to mimic a fixed support in a 3 DoF system. This has the effect of creating a reaction force couple, which is equivalent to the reaction moment that would have otherwise occurred from a fixed support. A vertical point load is applied to the tip of the beam. Table A.2.2 shows the properties that are used for the set-up.

| Length | Segment length | Profile | Young's modulus | Point load |
|--------|----------------|---------|------------------------------|------------|
| 1.0 m | 0.05 m | CHS 50 | $2.1 \cdot 10^5 \text{ MPa}$ | 1.0 kN |

Table A.2.2 – Properties used for the cantilever example

K2ENGINEERING

A moment and shear plot of the cantilever beam from the K2Engineering analysis are shown in Figure A.2.3 and Figure A.2.4. The maximum displacement at the free end is 6.5 mm.

The right reaction force is 1 kN larger than the left reaction force and corresponds to the vertical reaction force from the point load. The reaction moment is calculated from the force couple and the segment length as $R_m = 20.0 \text{ kN} \cdot 0.05 \text{ m} = 1 \text{ kNm}$.

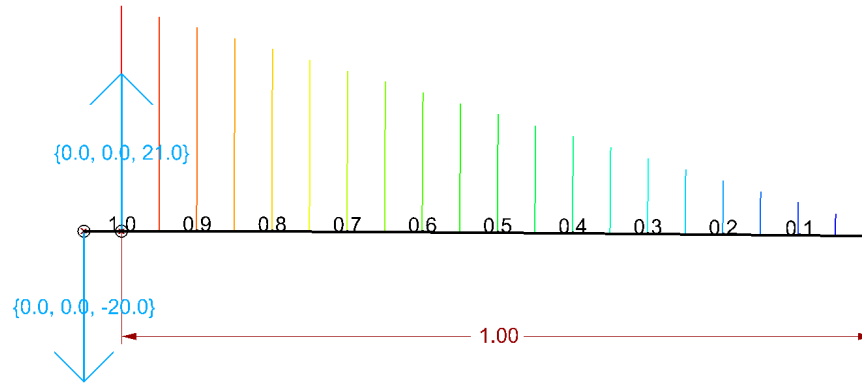


Figure A.2.3 – Cantilever beam with moment plot from K2Engineering. The moments are given in [kNm], the length measurement in [m] and reaction forces in [kN]

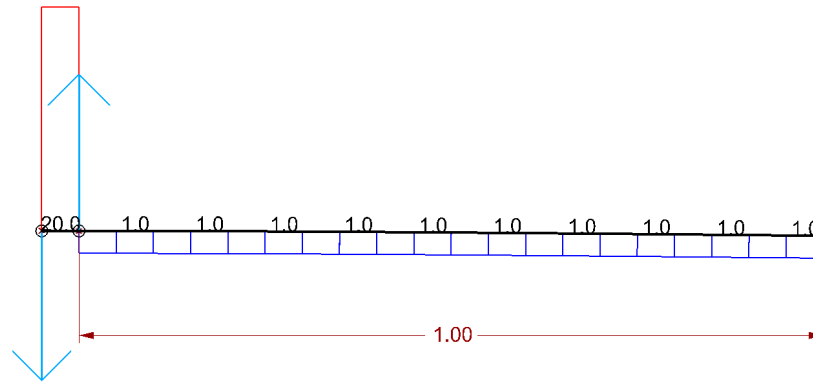


Figure A.2.4 – Cantilever beam with shear plot from K2Engineering. The shear forces are given in [kN] and the length measurement in [m]

ANALYTICAL

The moment distribution along the beam is calculated by hand from $M(x) = -F \cdot x$, where F is the point load and x is the distance along the beam (away from the load).

The moments along the beam from left to right are as follows:

$$M(1000) = 1 \cdot 10^3 \cdot 1000 = 1.0 \text{ kNm}$$

$$M(900) = 1 \cdot 10^3 \cdot 900 = 0.9 \text{ kNm}$$

$$M(800) = 1 \cdot 10^3 \cdot 800 = 0.8 \text{ kNm}$$

⋮

$$M(100) = 1 \cdot 10^3 \cdot 100 = 0.1 \text{ kNm}$$

The shear force is constant along the beam length and is equal to the applied point load of 1 kN. The vertical reaction force at the support is similarly equal to the magnitude of the point load and the reaction moment equal to $M(1000)$.

The maximum deflection is calculated from

$$d_{max} = \frac{1}{3} \cdot \frac{F \cdot L^3}{E \cdot I} = \frac{1}{3} \cdot \frac{1 \cdot 10^3 \cdot 1000^3}{2.1 \cdot 10^5 \cdot 0.262 \cdot 10^6} = 6.1 \text{ mm}$$

It is observed that there is a good correspondence between the results obtained from the K2Engineering analysis and the analytical solution. Only a small deviation with regard to the maximum displacement at the end is noticed.

A.2.3 Frame

A rigid frame of height 2.0 m and a span of 5.0 m is pinned at its ends and subjected to self-weight and a point load at the middle. Table A.2.3 shows the properties that are used for the set-up.

| Length | Height | Profile | Young's modulus | Point load | Density |
|--------|--------|---------|------------------------------|------------|-----------------------|
| 5.0 m | 2.0 m | CHS 50 | $2.1 \cdot 10^5 \text{ MPa}$ | 1.5 kN | 7850 kg/m^3 |

Table A.2.3 – Properties used for the frame example

K2ENGINEERING

A moment and shear plot of the frame from the K2Engineering analysis are shown in Figure A.2.5 and Figure A.2.6. The maximum moment is located at the middle of the frame where it is also observed that the shear force changes sign. The maximum displacement at the middle is 31.7 mm and the biggest compression force is -1.0 kN at the bottom of the frame.

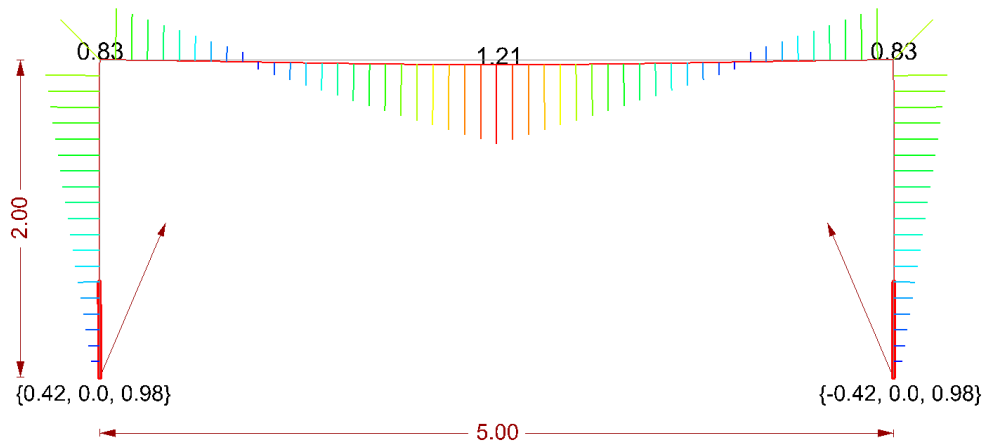


Figure A.2.5 – Frame with moment plot from K2Engineering. The moments are given in [kNm], the length measurements in [m] and reaction forces in [kN]

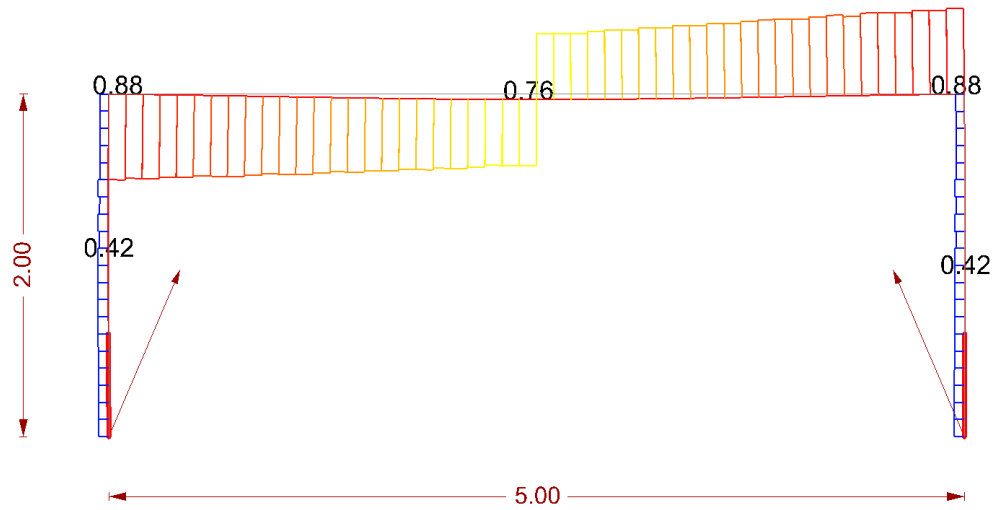


Figure A.2.6 – Frame with shear plot from K2Engineering. The shear forces are given in [kN] and the length measurements in [m]

KARAMBA

A moment and shear plot of the frame from the Karamba analysis are similarly shown in Figure A.2.7 and Figure A.2.8. The maximum displacement at the middle is 32.4 mm and the maximum compression force at the bottom of the frame is -1.0 kN.

By comparing the results from the two analysis, it is observed that the normal-, shear- and reaction forces as well as the moments are identical and only a small difference with regard to the maximum displacement at the middle exists.

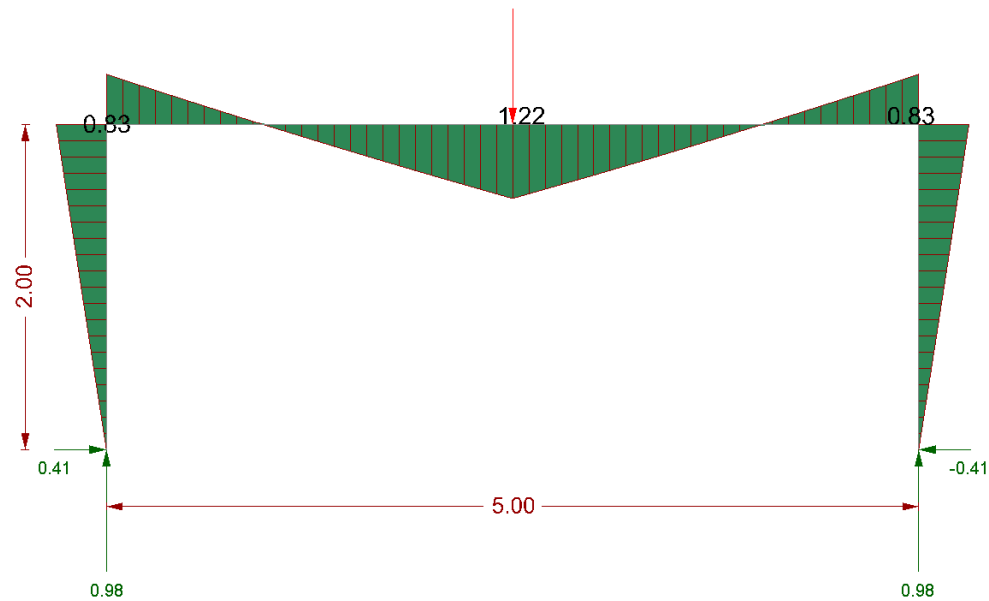


Figure A.2.7 – Frame with moment plot from Karamba. The moments are given in [kNm], the length measurements in [m] and reaction forces in [kN]

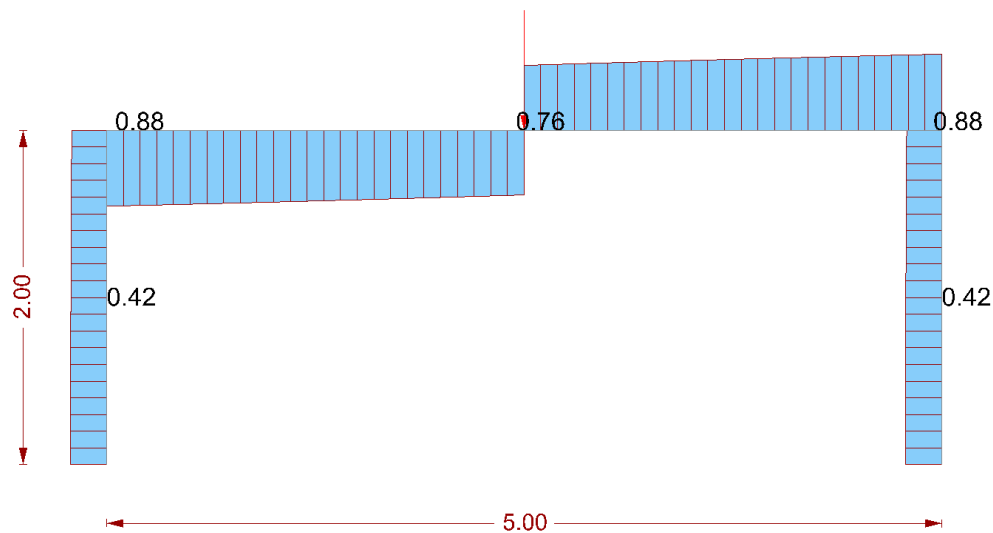


Figure A.2.8 – Frame with shear plot from Karamba. The shear forces are given in [kN] and the length measurements in [m]

A.2.4 Arch

An arch with a span of 5.0 m is pinned at its ends and subjected to self-weight and an uniform distributed load. Table A.2.4 shows the properties that are used for the set-up.

| Length | Profile | Young's modulus | UDL | Density |
|--------|---------|------------------------------|----------|-----------------------|
| 5.0 m | CHS 50 | $2.1 \cdot 10^5 \text{ MPa}$ | 2.5 kN/m | 7850 kg/m^3 |

Table A.2.4 – Properties used for the arch example

K2ENGINEERING

A moment and shear plot of the arch from the K2Engineering analysis are shown in Figure A.2.9 and Figure A.2.12. The maximum displacement at the middle is 27.2 mm and the biggest compression force is -9.8 kN at the bottom of the arch.

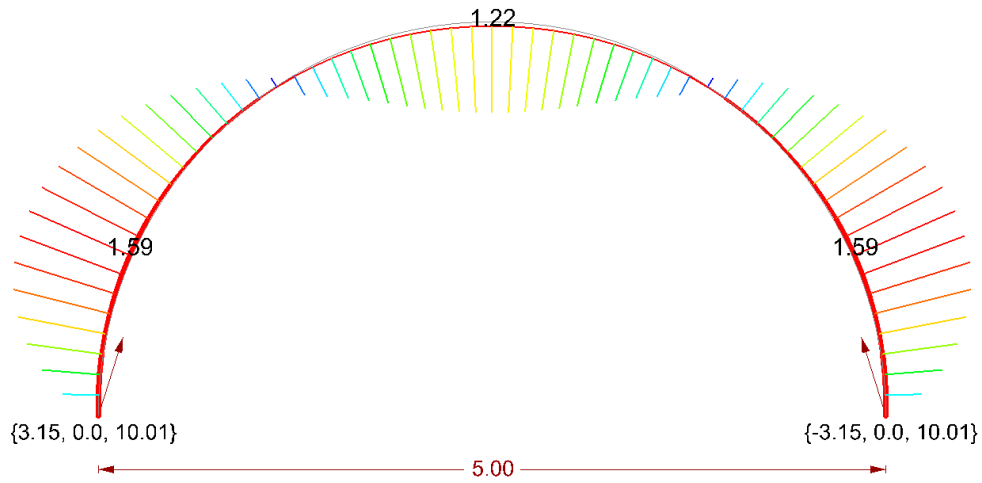


Figure A.2.9 – Arch with moment plot from K2Engineering. The moments are given in [kNm], the length measurement in [m] and reaction forces in [kN]

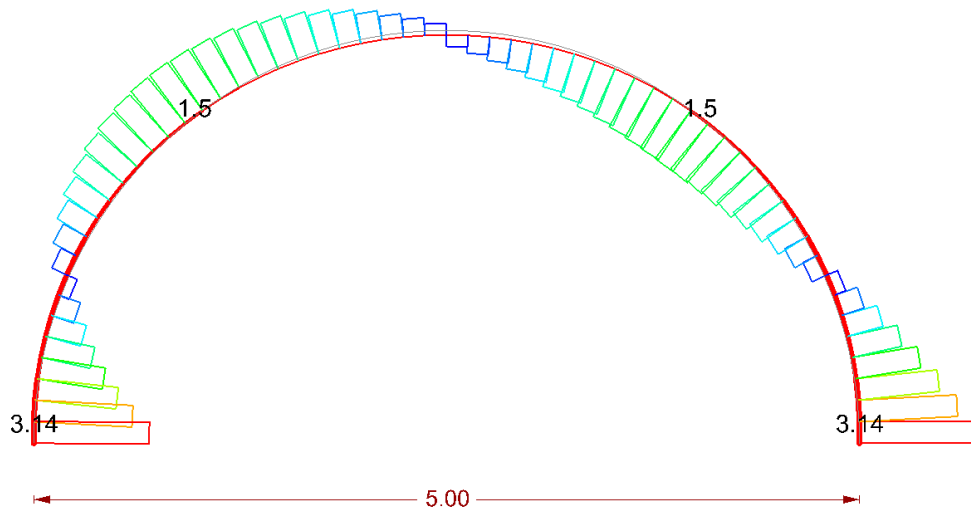


Figure A.2.10 – Arch with shear plot from K2Engineering. The shear forces are given in [kN] and the length measurement in [m]

KARAMBA

A moment and shear plot of the arch from the Karamba analysis are similarly shown in Figure A.2.11 and Figure A.2.12. The maximum displacement at the middle is 27.0 mm and the maximum compression force at the bottom of the arch is -9.9 kN.

Again, it is observed that the results from the two analysis are identical.

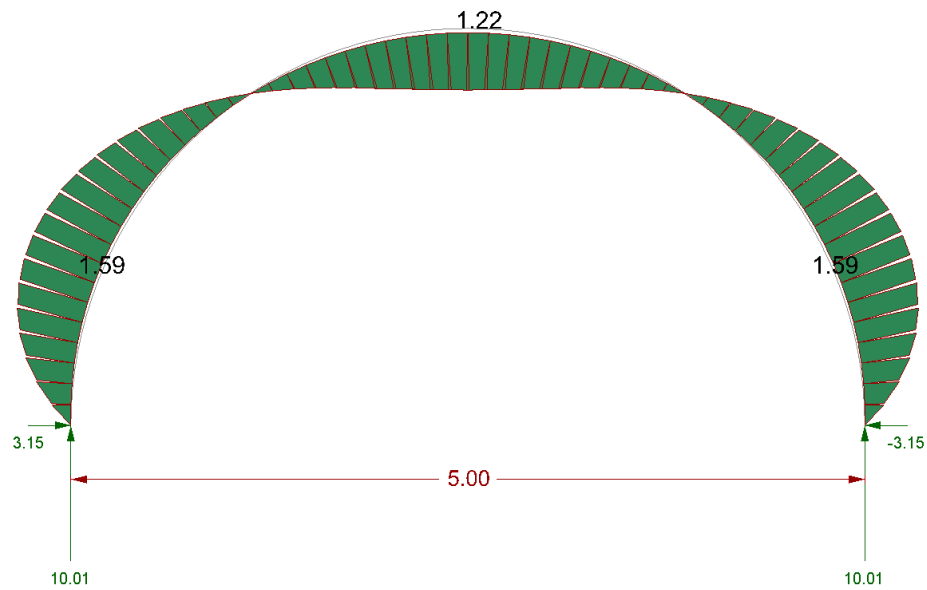


Figure A.2.11 – Arch with moment plot from Karamba. The moments are given in [kNm], the length measurement in [m] and reaction forces in [kN]

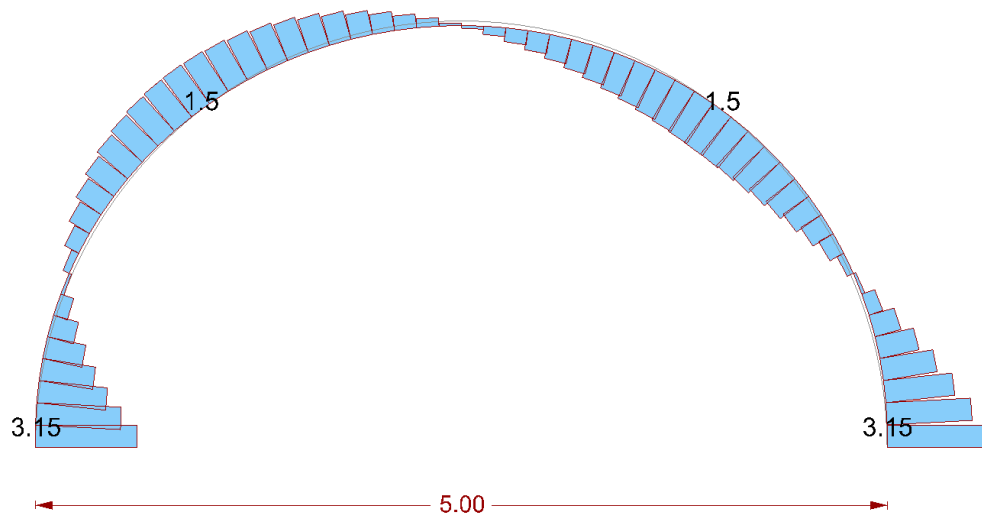


Figure A.2.12 – Arch with shear plot from Karamba. The shear forces are given in [kN] and the length measurement in [m]

A.3 Buckling

A.3.1 Column

A 1.0 m tall column is pinned at the bottom and free to move in the vertical direction at the top and subjected to a normal force. The properties used for the set-up is shown in Table A.3.1.

| Length | Buckling length | Diameter | Young's modulus | Vertical load | Horizontal load |
|--------|-----------------|----------|-----------------|---------------|-----------------|
| 1.0 m | 1.0 m | 6 mm | 45 000 MPa | 1 N | 0.01 N |

Table A.3.1 – Properties used for the column buckling example

K2ENGINEERING

A small horizontal load is applied at the middle of the column in order to cause an out-of-plane buckling behaviour. The magnitude of this load is constant throughout the buckling analysis. The result is illustrated in Figure A.3.1 where the black line indicates the initial configuration and the blue curve is the shape after the first sudden change in displacements. The buckling load factor is estimated to 33.0 with a maximum displacement at the middle of 25 mm.

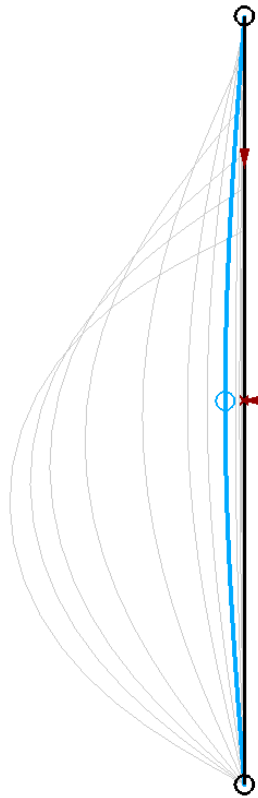


Figure A.3.1 – Column buckling behaviour from K2Engineering

ANALYTICAL

The critical load is determined by:

$$N_{cr} = \frac{\pi^2 \cdot E \cdot I}{L_s^2} = \frac{\pi^2 \cdot 45000 \cdot 63.62}{1000^2} = 28.2 \text{ N}$$

Since the applied load equals 1.0 N, this corresponds to a buckling load factor of 28.2.

The buckling load factor from the non-linear analysis is observed to be on the unsafe side of the analytical solution but within an acceptable range.

Appendix B

K2Engineering overview

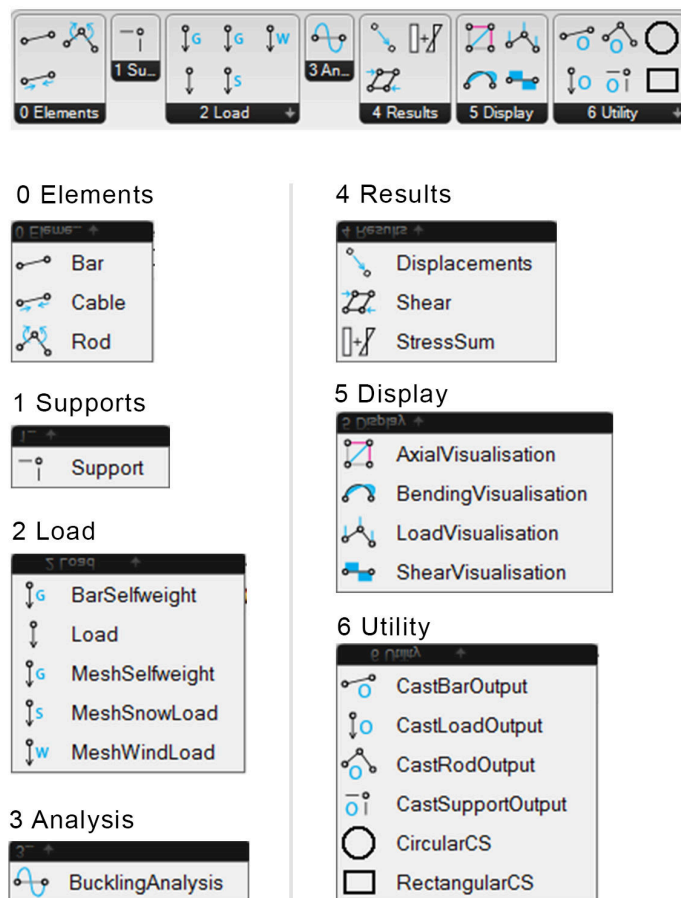


Figure B.0.1 – K2Engineering components

This appendix provides an overview of the developed K2Engineering plug-in with all its components.

| Component name | Functionality |
|----------------|--|
| Bar | Creates a bar goal, which provides axial stiffness in both tension and compression |
| Cable | Creates a cable goal, which provides axial stiffness in tension only |
| Rod | Creates a rod goal, which provides bending stiffness to resist out-of-plane forces |

Table B.0.1 – Element components

| Component name | Functionality |
|----------------|--|
| Support | Creates a support goal to restrict the movement in the X, Y and Z directions of a selected point |

Table B.0.2 – Support component

| Component name | Functionality |
|----------------|---|
| BarSelfweight | Calculates the lumped nodal forces from a list of lines, cross section area and the material density |
| MeshSelfweight | Calculates the lumped nodal forces from a triangulated mesh based on the vertex voronoi area, thickness and material density |
| MeshSnowLoad | Calculates the lumped nodal forces from a triangulated mesh based on the projected vertex voronoi area and a snow load |
| MeshWindLoad | Calculates the lumped nodal forces from a triangulated mesh based on the vertex voronoi area, a wind load and direction. The vertex normals are projected onto the wind direction to distinguish between pressure and suction |
| Load | Creates a load goal to apply the forces to the K2 simulation |

Table B.0.3 – Load components

| Component name | Functionality |
|------------------|--|
| BucklingAnalysis | Performs a non-linear buckling analysis from the specification of permanent goals and load goals. The load goals are incrementally increased in between the equilibrium iterations and the displacements simultaneously traced |

Table B.0.4 – Analysis component

| Component name | Functionality |
|----------------|--|
| Displacements | Measures the distance between the points in the initial configuration and the updated configuration from the simulation. The existing K2 goal "show" is useful in this context to maintain a consistent particle order |
| Shear | Calculates the shear forces based on the moments that occur in each end of a line segment. The magnitude of the vector corresponds to the shear value |
| StressSum | Calculates the total stress per line segment as a summation of the axial and bending stresses |

Table B.0.5 – Results components

| Component name | Functionality |
|----------------------|---|
| AxialVisualisation | Displays the axial forces in the structure by colouring the lines (red = compression, blue = tension, green = neutral) and scaling the line widths according to the magnitude of the forces |
| BendingVisualisation | Displays the bending moments as coloured lines (blue = low, green = medium, red = high) in the bending planes with scaled lengths according to the magnitude of the moments |
| LoadVisualisation | Displays the loads at the updated particle positions |
| ShearVisualisation | Displays the shear forces as coloured lines (blue = low, green = medium, red = high) perpendicular to each line segment and scaled according to the magnitude of the shear forces |

Table B.0.6 – Display components

| Component name | Functionality |
|-------------------|--|
| CastBarOutput | Extracts the output from the bar/cable goal: start and end particle indexes, updated line, axial force and stress |
| CastRodOutput | Extracts the output from the rod goal: particle index of the shared vertex between the two consecutive lines, the bending plane, moment and stress |
| CastSupportOutput | Extracts the output from the support goal: the position and reaction force |
| CastLoadOutput | Extracts the output from the load goal: updated position and the load vector itself |
| CircularCS | Calculates the area, moment of inertia and distance to the outer fibre for a circular cross section |
| RectangularCS | Calculates the area, moment of inertia and distance to the outer fibre for a rectangular cross section |

Table B.0.7 – Utility components