

Form Finding of Grid Shells

a Parametric Approach using Dynamic Relaxation

Henrik Green & Daniel Lauri

MSc Thesis in Mechanics
Department of Mechanics, KTH
Royal Institute of Technology

Stockholm 2017

Abstract

The rapid development of computational capacity in recent years has expanded the possibilities of digital modelling in architectural design. Parametric design has emerged from these possibilities with a capacity to generate complex geometries which call for advanced structural systems. Especially for form found structures, where the geometry is determined by structural mechanics, collaboration between architects and structural engineers is crucial in early design.

In this master thesis a form finding algorithm for grid shells has been developed. The algorithm is based on dynamic relaxation with kinetic damping coupled with a structural evaluation by the finite element method. The algorithm is applied to steel and glass grid shells, first with arbitrary boundary geometry and then in case studies where two shells are analysed in greater detail.

The algorithm is capable to form find grid shells with arbitrary boundary geometry. Convergence of dynamic relaxation is studied to ensure that a structure with a high degree of membrane action is found. Verification of the case study forms show that the algorithm produces grid shells with sufficient structural performance as long as the grid is sparse. However, the algorithm fails to provide structural stability for denser grid patterns, as shown by geometrically nonlinear analysis. Deviations from the conditions used to form find the grid shells are shown to adversely affect structural performance. Finally, an attempt to reduce the number of unique elements in grid shells is made using prescriptive dynamic relaxation. This is shown to be a difficult task for the case of complex geometry.

Structurally meaningful geometry can be achieved by a parametric modelling approach where constraints limit the solution space even if multiple parameters are included. The direct link between computational procedures and software commonly used in architectural design enables successful collaboration between architect and structural engineer.

Sammanfattning

De senaste årens snabba utveckling av beräkningskapacitet har utökat möjligheterna för digital modellering inom arkitektur. Parametrisk design har som följd av detta skapat möjligheter att generera komplexa geometrier vilka ställer krav på avancerade konstruktionslösningar. Speciellt för formfunna strukturer, där geometrin bestäms av strukturmekanik, är samarbetet mellan ingenjör och arkitekt avgörande i tidiga designskeden.

I detta masterarbete har en algoritm för formsökning av s.k. grid shell-strukturer utvecklats. Algoritmen baseras på dynamisk relaxering med kinetisk dämpning kopplad till utvärdering av strukturen med finita elementmetoden. Algoritmen appliceras på stål-glasskal, först för godtyckliga randgeometrier och sedan i två mer detaljerade fallstudier.

Algoritmen är kapabel att generera formfunna skal, uppbyggda av balkar, med godtycklig randgeometri. För att säkerställa hög membranverkan genomförs en konvergensanalys av resultaten från dynamisk relaxering. Verifiering av fallstudieformerna visar att algoritmen genererar skal med goda strukturella egenskaper så länge som mönsterindelningen är tillräckligt grov. För tätare mönsterindelningar misslyckas algoritmen med att säkerställa tillräcklig strukturell stabilitet, vilket visas i ickelinjär analys. Avvikelser från de förutsättningar som används vid generering av formerna visas ha kraftig påverkan på strukturernas beteende. Slutligen används dynamisk relaxering med föreskrivna elementlängder i ett försök att reducera antalet unika element. Detta visas vara ett svårt problem för komplexa geometrier.

Strukturellt meningsfulla geometrier kan uppnås genom parametrisk modellering där restriktioner begränsar lösningsmängden även om modellen innehåller ett flertal parametrar. Den direkta länken mellan numeriska beräkningar och mjukvara som ofta används inom arkitektur möjliggör framgångsrika samarbeten mellan arkitekt och ingenjör.

Preface

This master thesis was written at the Department of Mechanics at the Royal Institute of Technology in Stockholm in collaboration with the structural engineering department at consultancy firm Tyréns. The thesis is part of the masters program Civil and Architectural Engineering at the school of Architecture and the Built Environment, KTH.

The authors wish to thank a number of people whom have made this master thesis possible. First and foremost, supervisor and examiner Prof. Anders Eriksson at the Department of Mechanics, KTH, for his patience and invaluable expertise. Anders' excellent teaching and mentorship has been a great inspiration during the time at KTH and in the work with this project.

Further, we wish to thank co-supervisor Karl Graah-Hagelbäck, Head of Structural Engineering Department at Tyréns, for his support and encouragement. Per Backlund and Jonas Lejon at Comsol AB for providing us with software licences and modelling support. Finally, we wish to thank Tyréns for providing us with office space and additional software.

Contents

1	Introduction	1
2	Background	3
2.1	Structural Design Collaboration	3
2.2	Parametric Design	3
2.2.1	Computational Design	4
2.2.2	Architectural Geometry	4
2.3	Form Finding	5
2.3.1	Shell Structures	5
2.3.2	Grid Shell Structures	5
2.4	Early Work	7
2.4.1	Bundesgartenschau 1975 in Mannheim	7
2.4.2	Museum of Hamburg History	7
2.5	Recent Work	9
2.5.1	British Museum	9
2.5.2	Dutch Maritime Museum	9
2.6	Definitions	11
2.6.1	Free Form Shells	11
2.6.2	Form Found Shells	11
2.6.3	Form Finding	11
2.6.4	Structural Optimization	11
2.6.5	Parameter	11

2.7	Reference Project: New Town Hall Uppsala	12
3	Method	15
3.1	Coupled Form Finding Overview	15
3.2	Parametric Approach	16
3.3	Geometry	17
3.3.1	Grid Pattern	18
3.3.2	Case Study 1	18
3.3.3	Case Study 2	18
3.3.4	Rhino and Grasshopper	19
3.4	Notations	20
3.5	Form Finding	21
3.5.1	Script	22
3.5.2	Dynamic Relaxation	22
3.5.3	Data Structure	28
3.5.4	Kinetic Damping	29
3.5.5	Loaded Network	30
3.5.6	Voronoi Distribution	32
3.5.7	Self-stressed Network	32
3.5.8	Algorithm Convergence	32
3.6	Prescriptive Dynamic Relaxation	33
3.7	Finite Element Method	34
3.7.1	FEM Theory	34
3.7.2	Beam Orientation	37
3.7.3	Commercial Software	38
3.8	Optimization	38
3.8.1	Optimization Procedure	40
3.8.2	Cost Function	41

3.9	Verification and Structural Stability	43
3.9.1	Load Cases	43
3.9.2	Boundary Conditions	45
3.9.3	Global Stability	47
3.9.4	Dynamic Analysis	48
3.10	Method Summary	49
4	Results	51
4.1	General Form Finding Results	51
4.1.1	Form	51
4.1.2	Case Study Form	56
4.1.3	Preliminary Studies	58
4.1.4	Boundary Conditions	60
4.2	FEM script Verification	61
4.3	Convergence Analysis of Kinetic Energy Tolerance	62
4.4	Convergence Analysis of Geometric Exportation Tolerance	65
4.5	Grid Shell with Single Boundary	66
4.5.1	Optimization Data	66
4.5.2	Verification with Pattern Division 20	67
4.5.3	Verification with Pattern Division 40	68
4.6	Grid Shell with Interior Boundary	70
4.6.1	Optimization Data	70
4.6.2	Verification with Pattern Division 20	71
4.6.3	Verification with Pattern Division 40	72
4.7	Global Stability and Imperfections	74
4.8	Prescriptive Dynamic Relaxation	77
4.8.1	Continuous Arch	77
4.8.2	Grid Shell on Single Boundary	78

5	Discussion	83
5.1	Form and Geometry	83
5.2	Convergence Analysis	84
5.3	Case Studies	85
5.4	Global Stability	86
5.5	Prescriptive Dynamic Relaxation	87
5.6	Parametric Approach	87
6	Conclusions	89
6.1	Further Studies	90
A	Dynamic Relaxation Script	93

Chapter 1

Introduction

At the time of the industrial revolution, development of new building materials along with high demands on infrastructure and transportation systems required highly specialised engineers with a profound knowledge in material and mechanics [1]. The role of the structural engineer was very important in the development of new calculation methods and construction techniques. The development of steel, cross laminated timber and reinforced concrete gave rise to impressive bridge structures which utilised the material techniques available at the time. The introduction of glass, plastics and composite materials in the 20th century enabled new architectural forms which resulted in new complex structures. Today, the introduction of computer aided design along with increased computational capacity has been described as the new paradigm in structural engineering and architecture [1]. Computer power enables designers to develop complex forms which require advanced structural systems. However, complex designs requires close collaborations between architects and engineers when aiming for material efficiency, which is not always the case in architectural design.

In recent years a renewed interest in materialisation and fabrication in architecture and engineering has opened up new possibilities for collaborative design processes [2]. Digital modelling in general, and parametric design in particular has enabled architects and engineers to continuously develop design ideas throughout the design process. The increased computational capacity and high precision modelling give structural engineers the opportunity to guide the design by structural evaluation. The outcome would be unattainable without a collaborative process.

An area where this is evident is form-finding, where the global geometry is governed by structural mechanics. The concept of form finding is to use the load that the structure is subjected to in order to generate the shape. The form found shape then carries the load efficiently with minimal material consumption. The subject of study in this report is form finding of grid shell structures.

Previous research suggests several numerical form-finding techniques for shell structures. Here, the method of dynamic relaxation is used. The method was first employed as a form finding technique by Barnes [3], but also used in the solution of non-linear mechanical equilibrium problems, in e.g. [4]. The method has been employed by architects and structural engineers to mimic the physical hanging chain model. For grid shell structures the discretization of a smooth shell surface raises issues on panel planarity, optimal grid pattern configuration and structural optimization [5].

In this report, the method of dynamic relaxation is employed for grid shell structures. The aim is to develop a form-finding algorithm coupled with a finite element solver in order to generate structurally efficient forms. The algorithm is based on a parametric model in order to handle a wide range of shell forms.

The report begins with a short background on parametric design, form-finding, previous projects and definitions in section 2. Following the background the method employed in the study is presented. The methodology of generating initial geometry, form-finding, optimization and verification is described along with a detailed description of dynamic relaxation in section 3. General form-finding results are presented along with a detailed study of two cases in section 4. The developed algorithm and the results are discussed in section 5 and conclusions are presented in section 6.

Chapter 2

Background

2.1 Structural Design Collaboration

The built environment of today's societies is shaped by a broad variety of disciplines that form a complex network of relations and trade-offs between aesthetics, technical, cultural and economic concerns. This network has crystallised into distinct and highly specialised entities where each entity deals only with its limited field.

More specifically, the form of collaboration between architect and structural engineer has been a process where the architect designs before handing over to the engineer with the task to make it work. Recently however, engineers have begun to influence design at a much earlier stage in the design process. As highlighted by Oxman and Oxman [2] this shift has been made possible by engineers working with the same technological media as architects and by a renewed interest in materialisation within architecture in the tracks of digital fabrication. Engineering which mediates material and geometric concern with design intent are termed "design engineering" by Oxman.

This design engineering is closely connected to the field of parametric design where geometry is treated and manipulated through variables. The introduction of these variables and the rules of how to process them enables incorporation of engineering concerns in early design.

2.2 Parametric Design

Parametric design is a design work flow where algorithmic scripting logic is applied to generate geometry in a digital environment. It emerged in the late 1980s as architects began to rethink the construction of geometry and form as a process where "...designers first and foremost articulate an internal generative logic, which then produces, often in an automatic fashion, a range of possibilities..." as stated by Kolarevic [6]. In highly abstract terms parametric design may be described as a parameter set which contains the variables and a rule set of how to handle the parameters. The input parameters produces the design by the rule set. See further section 3.2.

Thus, parametric design entails a necessity by which the outcome follows from the input parameters and the rule set. The strength of parametric geometry lies in the potential of design exploration prior to final design. Herein lies also the weakness of the method, in a situation with an extreme range of possibilities, how one decides which is the best is no straightforward task.

Decisions must be guided by some form of evaluation, intuitive or rational based on numerical or practical values. Inherent in parametric design is a feedback between manipulation and outcome. The outcome is not limited to qualities which are easily quantified such as length or weight. Instead it reaches over areas such as indoor climate, sustainability and all the way to aesthetic expression. From this follows, that once the parameter and rule set is established, intuition as well as systematic variation is available to explore design alternatives. For such qualities which may be quantified accurately, computation provides a way to evaluate design alternatives. From this fact it is clear that parametric design, in many aspects, is closely related to the concept of computational design.

Another aspect of parametric design is the possibility to generate complex geometries. This raises new issues of the relation between digital model and physical representation. The geometric difficulties have given rise to a new field of study called architectural geometry.

2.2.1 Computational Design

Computational design is a design methodology where calculations are employed to evaluate design alternatives. Within the field of architectural engineering it can be thought of as a subcategory of parametric design where computational effort rules over intuition.

Computational design is a field with broad applications. One instance where it may be applied is within the field of structural engineering. Structural results such as material stress, displacements and force distributions may be used to guide design decisions. These results can be obtained by numerical evaluation of a design alternative.

2.2.2 Architectural Geometry

The emergence of powerful CAD software has rapidly rendered modelling of complex geometry possible. On the other hand, fabrication techniques have not developed as quickly and the conversion from digital to material surface is a difficult task which has attracted considerable attention recently. This field is called Architectural Geometry, of which Pottmann et al. [7] provide an extensive review of research directions. One important problem within architectural geometry is how to construct a free form surface from smaller parts. This problem can be treated by the discretization of free form surfaces into smaller segments, a process called panelization or by finding an approximation of the desired surface with more favourable geometric properties from a construction point of view.

2.3 Form Finding

The concept of form finding can be summarised by: form follows force. The structural form is determined by the forces it is subjected to such that loads are carried efficiently. When striving for minimum material consumption, optimization of single components is not enough. As described by Schober [8], form finding where the global structural behaviour defines the geometry is more efficient when aiming for structural optimization.

2.3.1 Shell Structures

An important class of structures where form finding may be applied is shell structures. A shell is a thin curved surface, continuous or faceted which, typically, carries load efficiently with little or no bending [8]. The geometric shape of a shell enables forces to be transmitted within the surface through compression and tension. The efficient load bearing yields thin, material efficient structures compared to structures in bending.

2.3.2 Grid Shell Structures

A subclass of shell structures are grid shell structures where a shape is composed of a discrete system of edges and vertices. The spatial relation of the vertices is such that imposed loads are carried by axial forces in the edge directions. The edges and vertices are represented physically as structural members and nodes in the grid.

Grid patterns commonly used in grid shells typically consists of quadrilaterals, right triangles or equilateral triangles. A combination of geometric shapes is also possible, discussed in e.g. [7]. But a grid shell may be composed of any grid pattern including ornamental patterns, which are used structurally.

However, not all patterns provide desired properties in for example how well they are able to approximate doubly curved surfaces. Honeycomb patterns has shown to be a good representation of arbitrary surfaces when aiming for repetition within the pattern. According to Pottman et al. an arbitrary, smooth surface can be approximated with a non-smooth grid pattern using the honeycomb figuration [7], see Figure 2.1.

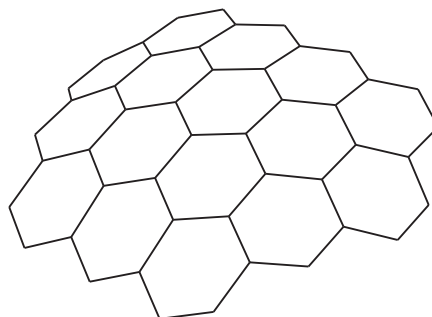


Figure 2.1: Honeycomb pattern approximating a curved surface

An extension of the honeycomb pattern was introduced by architect Alain Lobel, the so called Lobel frames to approximate a non smooth surface with repetition of equilateral triangles [7]. The Lobel frame grid pattern uses the shape of the honeycomb pattern to achieve triangular facets, see Figure 2.2. The triangular facets have two advantages over facets with more than three edges. Firstly, the triangular shape provides stability by itself and secondly, triangular facets ensure they remain planar.

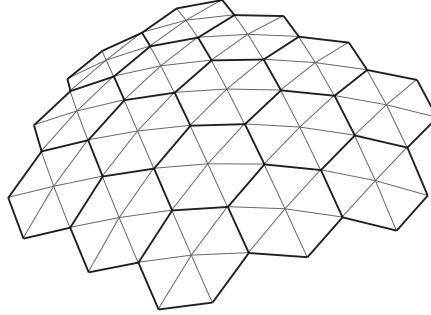


Figure 2.2: Lobel pattern of equilateral triangles in a honeycomb pattern

2.4 Early Work

To highlight how the conception of grid shells have evolved and put them in a historic context two early examples are presented.

2.4.1 Bundesgartenschau 1975 in Mannheim

The grid shell structure of the Herzogenriedpark building of the Mannheim Bundesgartenschau 1975 is an early example of a form found structure realised through a collaborative design process. Architect Carl Mutschler & Partners collaborated with architect and engineer Frei Otto to find structural efficiency through form [9]. The concept of a physical hanging chain model was adapted in order to find a hanging shape in pure tension when subjected to self weight. To achieve uniform tension in the hanging grid, springs were connected to the end of the chains and adjusted manually. The geometric expression was a result of the predefined boundary geometry and the hanging shape, see Figure 2.3.

According to Lidell [9] the finalised hanging chain model was measured and the nodal coordinates were adjusted through a newly developed numerical form finding method, the force density method. In order to find the shape in pure compression when subjected to self weight the hanging shape was inverted.

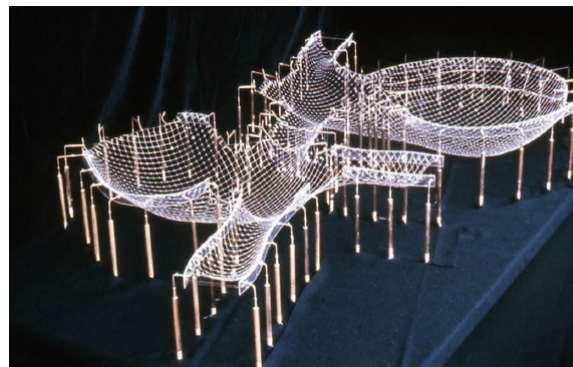


Figure 2.3: Hanging chain model of the Mannheim. Figure taken from [9]

2.4.2 Museum of Hamburg History

The L-shaped courtyard roof of the Museum of Hamburg History, in Figure 2.4, designed by architect Von Gerkan, Marg and Partners and structural engineer Jörg Schlaich is an early example of a steel-glass grid shell structure [10], completed in 1989. The L-shaped structure consist of two vault forms and a doubly curved dome form, see Figure 2.5, and was one of the first steel-glass grid shell structures built. The vault structure consist of a series of arches connected with longitudinal beams, creating a repeating quadratic grid pattern. To stiffen the arch form, additional cables were installed in a fan-like configuration. The technique of generating curved surfaces with planar panels was not developed at the time but was later solved based on the knowledge gained from the project.



Figure 2.4: Courtyard roof of Museum of Hamburg History. Figure taken from [11]

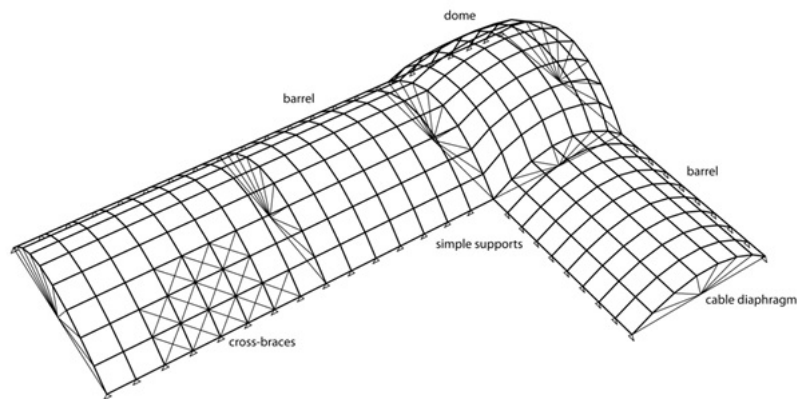


Figure 2.5: L-shaped roof structure of the Museum of Hamburg History. Figure taken from [10]

2.5 Recent Work

Two more recent examples show the further evolution of grid shell structures. The two structures use two different approaches to generate the form of the grid shell but share a common component, the use of dynamic relaxation.

2.5.1 British Museum

The courtyard roof of the British Museum, shown in Figure 2.6 was designed by Fosters & Partners in collaboration with structural engineering consultancy Buro Happold and completed in 2000. The grid shell structure is supported on an outer rectangular boundary of 72x96 m and an inner circular boundary with a diameter of 40 m [12]. The geometric shape of the shell was predefined by a mathematical surface expression on which the grid shell nodes lie as explained by Williams [13]. The method of dynamic relaxation was used to distribute a triangular grid pattern approximating the predefined surface geometry.



Figure 2.6: Interior view of the British Museum courtyard roof. Figure taken from [12]

2.5.2 Dutch Maritime Museum

The courtyard roof of the Dutch Maritime museum shown in Figure 2.7, designed by structural engineering consultancy Ney & Partners [14] is a recent example of a form found grid shell structure completed in 2011. The glass covered steel grid shell spans a distance of 34 m covering the courtyard of the historic Dutch Maritime Museum in Amsterdam, Netherlands. The historic environment introduced project restrictions in terms of structure height and boundary conditions, parameters affecting the form found shape.



Figure 2.7: View of the grid shell at the Dutch Maritime Museum. Figure taken from [14]

The form finding process started from a two dimensional grid structure subjected to self weight. A numerical representation of a hanging chain model was adapted to find a three dimensional shell form with only axially loaded members as presented by Adriaenssens et al. [5], see Figure 2.8. The numerical form finding was performed with the method of dynamic relaxation, see section 3.5.2, with an upward loading to find the structure in compression. The initial grid pattern was designed for aesthetic reasons which raised issues on glass panel planarity. The designers used a Maxwell Reciprocal Diagram, described in Adriaenssens et al. [5] to approximate a curved surface with flat panels.

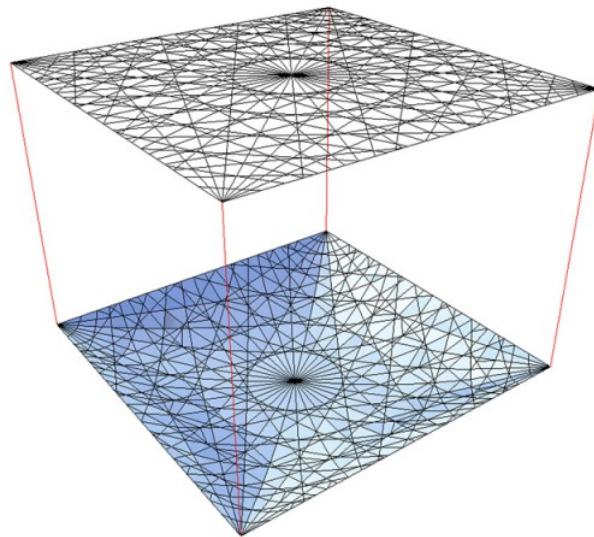


Figure 2.8: Planar grid and relaxed hanging form of Dutch Maritime Museum grid shell. Figure taken from [15]

The form finding process generated a shell shape with high architectural and sculptural qualities developed from an engineering approach on structural efficiency through form.

2.6 Definitions

Throughout the study this terminology is used in relation to shell structures and the methodology of the study.

2.6.1 Free Form Shells

A free form shell is defined as follows, "free-curved or sculptural shells are generated without taking into consideration structural performance." as defined by Adriaenssens et al. [15]. This type of shells are possible to generate in modelling software by manipulating a surface geometry without considering structural performance.

2.6.2 Form Found Shells

A form found shell is a shell for which the geometry has been determined such that it achieves structural efficiency through form. This means that the loads are carried by in-plane forces with little or no bending.

2.6.3 Form Finding

A method by which a structural geometry is determined by an imposed load such that the structure resists this load primarily by axial stress. The outcome of a form finding process is a form found structure which fulfils this criterion.

2.6.4 Structural Optimization

A procedure where a minimum is sought of an objective function. To find this minimum, a diverse range of methods are available. The objective function will be referred to as a cost function.

2.6.5 Parameter

A variable which is treated as an input by the parametric work flow. A parameter may be directly or indirectly controlled. Quantities which are potential parameters but given as a result from the algorithm is instead treated as costs in the optimization.

2.7 Reference Project: New Town Hall Uppsala

In 2016, Henning Larsen Architects and structural engineering consultancy Tyréns won the design competition [16] of the New Town Hall in Uppsala, Sweden, see Figure 2.9.



Figure 2.9: Section of the proposed design including the interior building. Figure taken from [16]

The design proposal comprises an extension of the existing L-shaped town hall with an additional L-shaped building. The two L-shaped building volumes enclose a glass roof covered courtyard, see Figure 2.10.

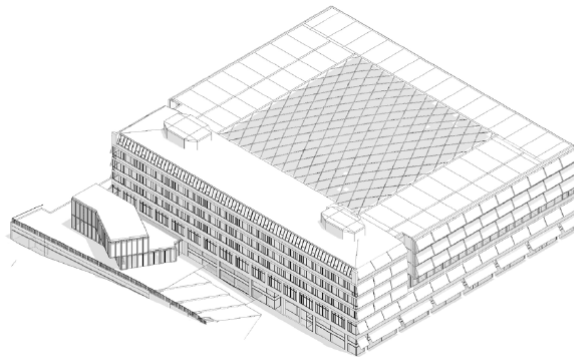


Figure 2.10: Exterior perspective view of building volumes and courtyard roof. Figure taken from [17]

The design for the courtyard roof proposes a structure with a quadrilateral grid pattern of steel members and glass facets with additional cables under the roof surface to carry tension loads in the structure [17], see Figure 2.11.

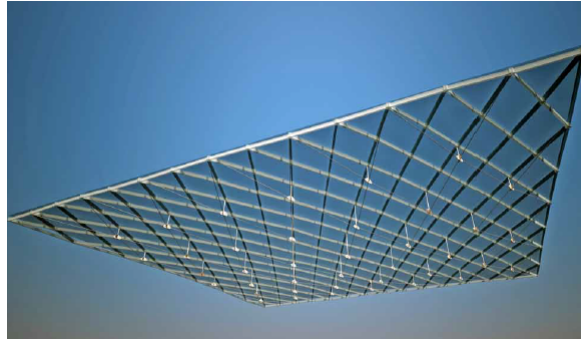


Figure 2.11: The proposed roof structure with glass panels supported on steel members with tension rods below. Figure taken from [17]

The project specific measurement of the Uppsala New Town Hall, see section 2.7, serves as a basis for the studied boundary geometries. The project specifications suggest a 40x40 m self supporting grid shell roof structure supported on two L-shaped buildings. The architectural competition proposal also includes an indoor building volume inside the courtyard which the authors suggest may serve as an additional boundary support. The precedence for the inner boundary design is the British Museum Roof, where a grid shell structure is supported on an outer rectangular boundary and inner circular structure, see section 2.5.1.

Chapter 3

Method

In this chapter the employed method is first summarised in an overview and then described in detail. Initial geometry of the grid shell, the form finding method along with optimization and verification is presented.

3.1 Coupled Form Finding Overview

The process used in this study to form find a grid shell structure is described schematically in Figure 3.1. The process consist of four major steps: Geometry, Form Finding, Optimization and Verification.

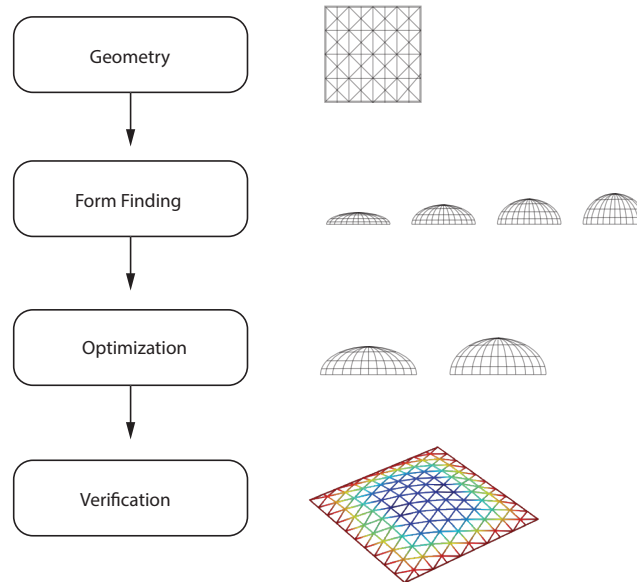


Figure 3.1: Overview of the four major steps in the procedure employed in this study

In the Geometry section a planar grid configuration is designed and the case geometries presented followed by a brief introduction to the parametric software used to generate geometry.

In the Form Finding section, geometric data from the planar grid configuration is used to generate grid shell structures by the method of dynamic relaxation, described in section 3.5.2. The form finding algorithm generates a series of grid shell structures which are evaluated in a cost function in the Optimization section. The structures with the lowest costs are then chosen to be verified and evaluated in an external FEM-software.

3.2 Parametric Approach

This study takes a parametric approach to the form finding of grid shells. The parametric approach consists of variables and of constraints. In the context of this study, such an approach does not mean that any form which stems from the parameters is admissible. This would render the parametric approach a CAD tool to draw shapes. Instead, the solution space is limited by a set of constraints on what forms are admissible. These constraints are formulated by structural mechanics. Any form found by the algorithm should satisfy static equilibrium, a high level of membrane action and member buckling as outlined in detail in section 3.5 and 3.8.

The relation of parametric and computational design proposed and adopted in this study is shown in Figure 3.2. In the parametric part, the parameter set, \mathbb{P} produces the design space, \mathbb{D} by the rules formulated in the parametric model. The design space, \mathbb{D} contains all the possible design outcomes from different parameter combinations. To arrive at a narrower design space, constraints are formulated according to structural mechanics and the design space \mathbb{D}^* is established. This subspace of \mathbb{D} is meant to contain all the possible designs which fulfils the constraints.

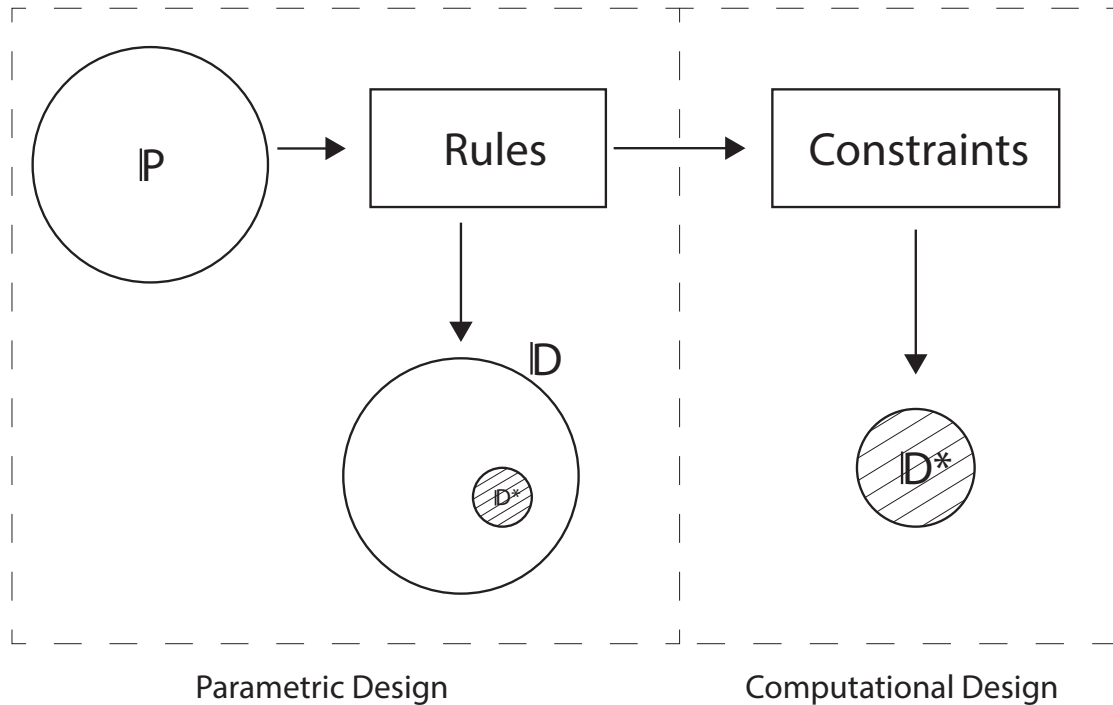


Figure 3.2: Parametric and computational design scheme

Table 3.1 presents the parameters which are available to manipulate in the form finding. The nodal coordinates are controlled by the parameter ϕ_E since they must be established by calculation as described in section 3.5.2. The term psuedo stems from the fact that ϕ_E lacks clear physical meaning. Other potential parameters are treated as characteristics of a certain form and considered as costs in the optimization outlined in section 3.8.2. The script is presented in section 3.5.1 and Grasshopper, which is a parametric modelling software, is presented in section 3.3.4.

Table 3.1: Parameters in study and how they are controlled

Parameter	Psuedo-parameter	Controlled in
Nodal coordinates	ϕ_E	Script
Grid pattern	-	Grasshopper
Grid density	-	Grasshopper
Grid boundary	-	Grasshopper
Load	-	Script

To reduce the need for excessive computations, choices are made of grid pattern in section 3.3.1 and two grid densities of 20 and 40 pattern divisions. The grid divisions refer to the number of elements along one side of the boundary geometry. Forms are first presented for a range of boundaries and then two grid boundaries are chosen for the case study according to section 2.7.

3.3 Geometry

This section describes the basic geometric configurations used to initiate the form finding of the grid shell structure. The choice of grid shell pattern and boundary geometry is described followed by a brief overview of how the grid shell geometry is represented to output data to the form finding algorithm, described in section 3.1 and 3.5.2

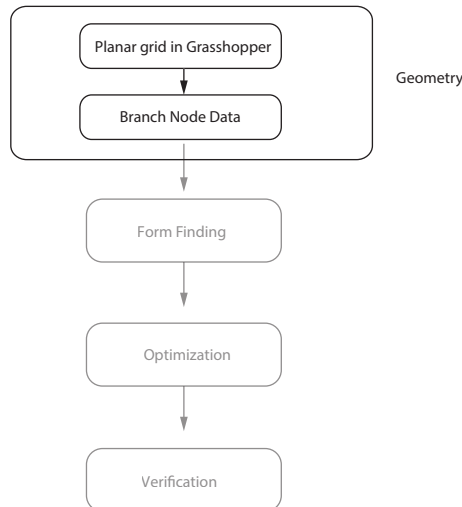


Figure 3.3: Geometry part of the study method

3.3.1 Grid Pattern

In this study the planar grid configuration is designed with a grid pattern approximating the Lobel frame pattern [7]. As describes in section 2.3.2 the Lobel frame pattern uses overlapping honeycomb geometries to achieve equilateral triangles. However, a pattern with equilateral triangles can not fill a square boundary without the introduction of asymmetries or triangles of different sizes. In order to achieve symmetry over a square boundary geometry, an approximation of equilateral triangles is used to generate the grid pattern in Figure 3.4 and throughout the study.

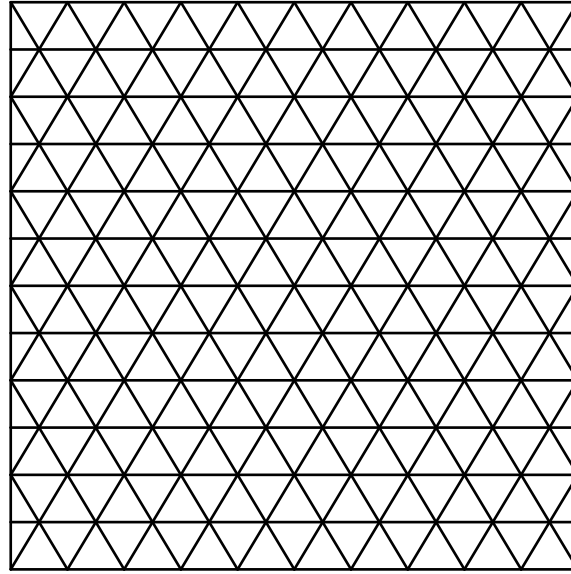


Figure 3.4: Studied grid pattern configuration designed with Lobel frames

3.3.2 Case Study 1

In Case study 1 the planar grid configuration presented in Figure 3.4 is used to generate form found grid shell structures. The square boundary geometry is 40x40 m and the case is studied for 20 and 40 pattern divisions.

3.3.3 Case Study 2

For Case study 2 the same grid pattern is used in the case of an interior boundary with the grid trimmed against the interior boundary as shown in Figure 3.5. The outer boundary geometry is 40x40 m and the inner circular boundary has a diameter of 10 m and is located with its center in (16,20,0), see Figure 3.5. To reach the inner building volume shown in Figure 2.9 the interior boundary is lowered 3 m relative the outer boundary. Case 2 is studied for 20 and 40 pattern division of the outer boundary.

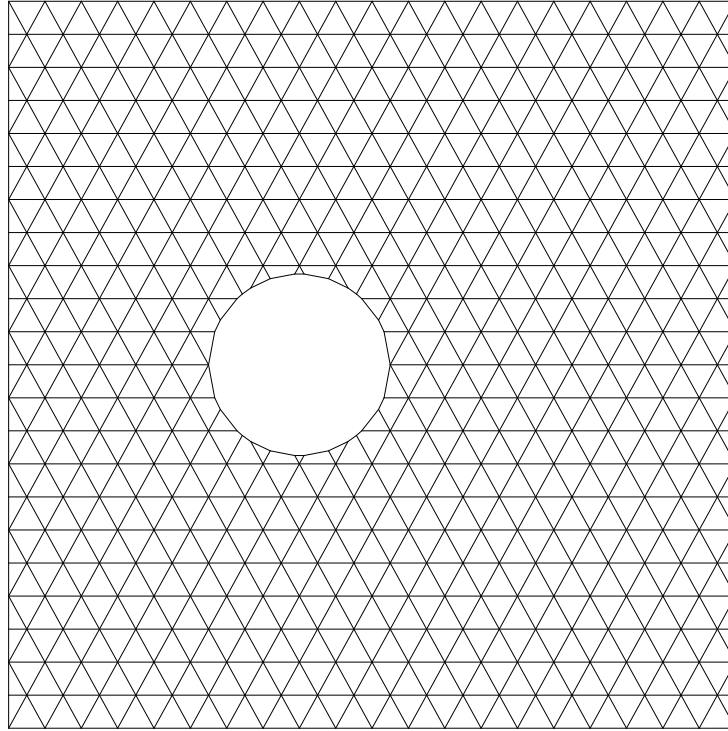


Figure 3.5: Grid pattern with interior boundary

3.3.4 Rhino and Grasshopper

To handle geometry, the 3D CAD software Rhinoceros 5 from McNeel and Associates is used along with the plug-in Grasshopper, version v.0.9.0076. Rhinoceros is hereafter referred to as Rhino. Grasshopper is an open source plug-in for algorithmic design. A typical Grasshopper work sheet is shown in Figure 3.6 where data flows from left to right in the wires connecting components which execute different operations on the data. The left to right flow of data renders looped processes difficult to handle with default components but Grasshopper can also be controlled by script for example in the Python language. This capacity is used in many of the operations in this study since it enables the user to tailor components to specific needs and bypasses the limitations on defining loops. An example of the script interface is shown in Figure 3.7. An advantage of scripting in the Grasshopper Python environment is that both the predefined components in Grasshopper and also native Rhinoceros commands are available to call as functions within the script.

In this study the main tasks performed in Grasshopper and Rhino is: draw the planar grid which will be the initial geometry in the form finding algorithm described in section 3.5. To convert the grid geometry to branch node data as explained in section 3.5.2. This data is also used as a basis for the FEM analysis described in section 3.7.1. Finally, Rhino is used to visualise the form found shape and export it in AutoCad .dwg format used in the FEM software.

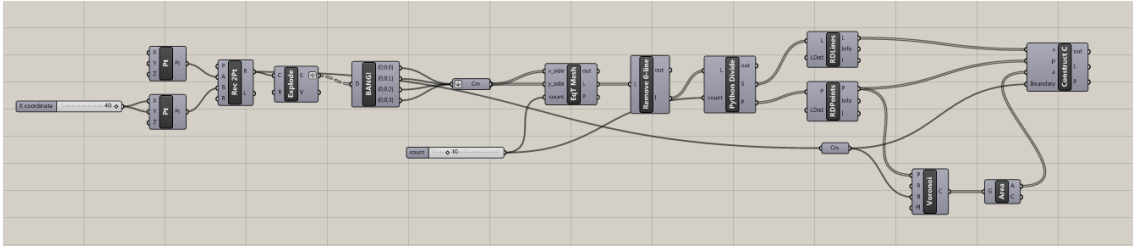


Figure 3.6: Screen shot of a typical Grasshopper workflow

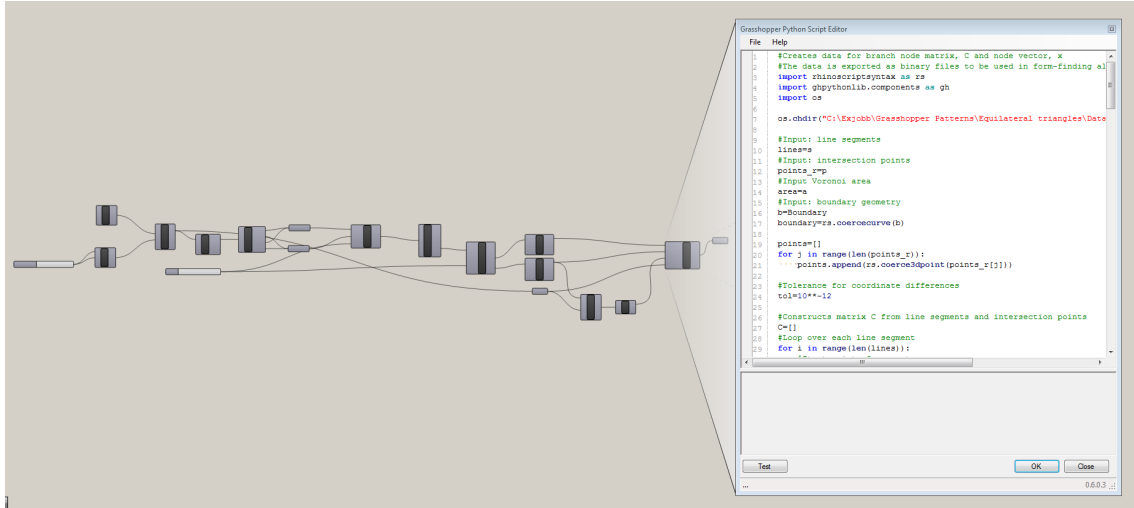


Figure 3.7: Screen shot from Grasshopper with a code window open where user defined components are scripted

3.4 Notations

Throughout the presentation of equations the following notations are used unless explicitly stated otherwise. Italics are used for scalar values, lower case bold letters represent vectors and uppercase bold letters represent matrices. An uppercase version of a letter representing a vector is a diagonal matrix where that vector is the diagonal.

3.5 Form Finding

The process of form finding, schematically presented in Figure 3.1, is presented in more detail in Figure 3.8. This section focuses on the form finding part where the method Dynamic relaxation is used to find a form in equilibrium with the imposed loads on the structure.

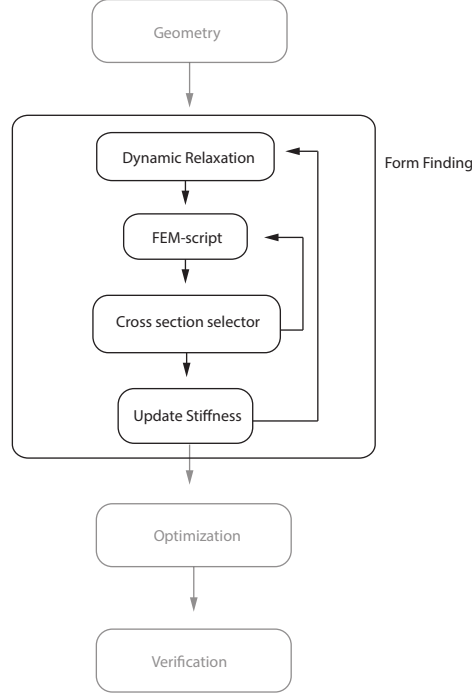


Figure 3.8: Form finding part of the study method

The main purpose of the form finding is to arrive at a geometry which by its shape carries a set of loads through axial forces. This objective is formulated as

$$\frac{W_{\text{axial}}}{W_{\text{total}}} \quad (3.1)$$

where W_{axial} is the axial strain energy of the structure and W_{total} is the total strain energy of the structure. Details on how the quantities are calculated are given in section 3.7.1. The ratio in equation 3.1 is used as a quantification of membrane action throughout the study and will approach 1 in a structure in pure membrane action.

3.5.1 Script

The algorithm, briefly presented in the form finding part of Figure 3.8, is implemented in the open source Python language with the numerical plug-in Numpy. The package used is Anaconda 4.3.0 available from [18]. The script is written in Python since it is one of the languages Grasshopper and Rhino works with. The scripted tasks include: data handling of export and import between Grasshopper and external Python script, form finding by dynamic relaxation, evaluation by linear FEM, cross section selection by member buckling and iteration of the above mentioned task to produce optimization data.

3.5.2 Dynamic Relaxation

The chosen form finding method in this study is dynamic relaxation [19]. In dynamic relaxation, DR, a structural configuration of static equilibrium is approximated from a dynamic formulation of a structure subjected to a set of loads. Dynamic relaxation was initially formulated as an analysis method of static equilibrium, used in e.g. [20] and [4]. Barnes [19] gives a brief history of the methods development. As a form finding tool it was first formulated for tensile structures by Barnes [21] but may be adopted for form finding of compressive structures by inverting the form found shape. An example of implementation is included in Appendix A.

Dynamic relaxation is chosen over other form finding methods for the following reasons:

- Dynamic relaxation is conceptually close to physical form finding methods such as hanging chain models.
- The method is a common tool for numerical form finding, used in for example [5], [19], [22] and [23].

The method is schematically presented in Figure 3.9 and described in detail below.

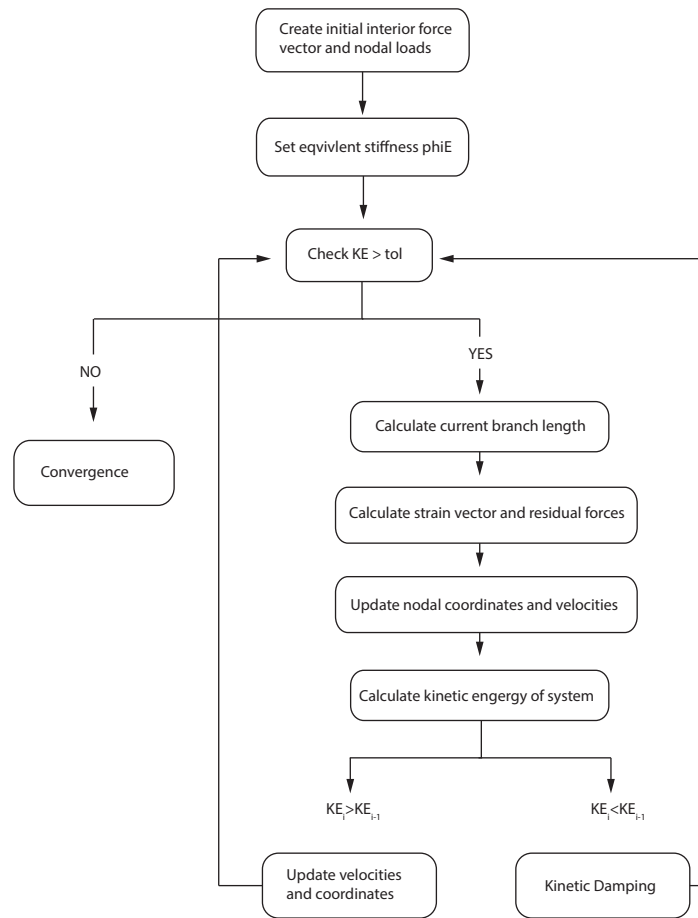


Figure 3.9: Flow chart of dynamic relaxation

Nodal loads and prescribed member forces are defined as described in section 3.5.5 and 3.5.7.

The mathematical presentation of Dynamic relaxation in equation 3.4 to 3.18 closely follows Block and Veenendaal [22]. In the paper by Block several form finding algorithms are presented with the same mathematical notation and data structure which renders this structure favourable from the point of flexibility in method choice.

However, this study, unlike Block and Veenendaal [22], focuses on structures in compression which renders the structural height important not only by the magnitude of member forces but also considering instability. For a grid shell structure the member forces will tend to decrease as the structural height increases. This is because the vertical component of member axial forces increases with height and forces necessary to counteract vertical loads are achieved at a lower axial load. Against this stands the risks of local member buckling and global instability which increase with height.

These conflicting effects call for a way to control structural height without changing physical properties such as member cross sections used elsewhere for example in the FEM evaluation and load definitions, see section 3.5.5. A way to control the structural height is proposed by Bagrianski and Halpern [23] where an elastic stiffness reduction is introduced according to

$$\phi_E = \frac{E_M}{E} \quad (3.2)$$

where E_M is the reduced elastic modulus used in form finding and E is the physical elastic modulus. The fictitious elastic modulus can be thought of as a reduction to the axial stiffness EA . This reduction is achieved without altering the cross section area, A which is used to calculate loads in equation 3.24 or the stiffness relation between elements. However, the method offers no rational way of anticipating structural height from ϕ_E which renders a trial and error approach inevitable.

The dynamic equilibrium is defined as Newton's second law of motion formulated as

$$\mathbf{M} \frac{\Delta \mathbf{v}}{\Delta t} = \mathbf{r} \quad (3.3)$$

where \mathbf{M} is a mass matrix for the structure, $\frac{\Delta \mathbf{v}}{\Delta t}$ is an expression for the acceleration and \mathbf{r} is the residual forces needed to maintain equilibrium. The sought equilibrium solution is the static solution where the residuals are zero implying that the velocities are constant. With stationary supports the velocities approach zero.

The structural grid is represented by a branch node matrix \mathbf{C} describing the topology of nodal coordinates \mathbf{x} . The branch node matrix \mathbf{C} is a $[3m \times 3n]$, where $\bar{\mathbf{C}}$, is a $[m \times n]$ matrix for a network with m branches and n nodes.

$$\mathbf{C} = \begin{bmatrix} \bar{\mathbf{C}} & & \\ & \bar{\mathbf{C}} & \\ & & \bar{\mathbf{C}} \end{bmatrix} \quad (3.4)$$

Each element C_{ij} in $\bar{\mathbf{C}}$ is

$$C_{ij} = \begin{cases} 1, & \text{if node } j \text{ is head of branch } i \\ -1, & \text{if node } j \text{ is tail of branch } i \\ 0, & \text{otherwise} \end{cases} \quad (3.5)$$

The $[3n \times 1]$ nodal coordinate vector is defined as

$$\mathbf{x} = \begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{y}} \\ \bar{\mathbf{z}} \end{bmatrix} = \begin{cases} \bar{\mathbf{x}} = (x_1, x_2, \dots, x_n)^T \\ \bar{\mathbf{y}} = (y_1, y_2, \dots, y_n)^T \\ \bar{\mathbf{z}} = (z_1, z_2, \dots, z_n)^T \end{cases} \quad (3.6)$$

To separate interior nodes with unconstrained translations from boundary nodes with prescribed positions the nodal coordinate vector is sorted according to

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_f \end{bmatrix} \quad (3.7)$$

where, \mathbf{x}_i is a $[3n_i \times 1]$ coordinate vector for the interior nodes and \mathbf{x}_f is a $[3n_f \times 1]$ vector for the fixed boundary nodes. The fixed boundary nodes are fixed in the sense that their positions are limited by some boundary conditions. Unless stated otherwise, all translations are restrained in fixed nodes.

Similarly, \mathbf{C} is separated as

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_i & \mathbf{C}_f \end{bmatrix} \quad (3.8)$$

where, \mathbf{C}_i is a $[3m \times 3n_i]$ branch-node matrix for the interior nodes and \mathbf{C}_f is a $[3m \times 3n_f]$ branch-node matrix for the fixed boundary nodes.

Coordinate differences between nodes of each branch can then be calculated as

$$\mathbf{u} = \begin{bmatrix} \bar{\mathbf{u}} \\ \bar{\mathbf{v}} \\ \bar{\mathbf{w}} \end{bmatrix} = \mathbf{C}\mathbf{x} \quad (3.9)$$

From the coordinate differences, branch lengths are calculated according to

$$\bar{\mathbf{L}} = (\bar{\mathbf{U}}^2 + \bar{\mathbf{V}}^2 + \bar{\mathbf{W}}^2)^{\frac{1}{2}} \quad (3.10)$$

where, $\bar{\mathbf{U}}$, $\bar{\mathbf{V}}$, $\bar{\mathbf{W}}$ are the coordinate differences in x, y and z-directions arranged in diagonal matrices for each Cartesian direction.

The branch lengths are then organised in the same way as in equation 3.4 into the $[3m \times 3m]$ branch length matrix, \mathbf{L} where $\bar{\mathbf{L}}$, is a $[m \times m]$ matrix.

$$\mathbf{L} = \begin{bmatrix} \bar{\mathbf{L}} & & \\ & \bar{\mathbf{L}} & \\ & & \bar{\mathbf{L}} \end{bmatrix} \quad (3.11)$$

Force densities in the branches are defined as

$$\mathbf{q} = \mathbf{L}^{-1}\mathbf{f} = \mathbf{L}^{-1}\mathbf{f}_0 + \mathbf{L}^{-1}\phi_E\mathbf{E}\mathbf{A}\boldsymbol{\varepsilon} \quad (3.12)$$

where, \mathbf{f}_0 is the prescribed forces, \mathbf{L} the current branch lengths, \mathbf{E} the Young's moduli, \mathbf{A} the cross section areas, ϕ_E as defined by equation 3.2 and $\boldsymbol{\varepsilon}$ strains. The strains are calculated member-wise with a small strain formulation as

$$\varepsilon = \frac{l_i - l_{r,i}}{l_{r,i}} \quad (3.13)$$

where l_i is the current branch length of member i and $l_{r,i}$ the relaxed length where the strain is zero. Here $l_{r,i} = l_{0,i}$. See further section 3.6 for possibilities to manipulate the solution by changing the relaxed lengths.

From the force densities, the $[3n_i \times 1]$ residual forces needed to maintain equilibrium in each node can be written as

$$\mathbf{r} = \mathbf{p} - \mathbf{C}_i^T \mathbf{U} \mathbf{q} = \mathbf{p} - \mathbf{C}_i^T \mathbf{L}^{-1} \mathbf{f}_0 + \mathbf{L}^{-1} \phi_E \mathbf{E} \mathbf{A} \boldsymbol{\varepsilon} \quad (3.14)$$

where \mathbf{p} is a vector of applied nodal loads.

Thus the reduction of elastic stiffness from equation 3.2 reduces the internal forces due to strain. This increases the strain necessary to reach equilibrium which in turn results in an increasingly deformed geometry.

For computational efficiency the lumped nodal masses can be written on vector form as,

$$\mathbf{m} = \frac{\Delta t^2}{2} |\mathbf{C}_i^T| (\mathbf{L}^{-1} \mathbf{F} + \mathbf{L}_0^{-1} \phi_E \mathbf{E} \mathbf{A}) \mathbf{1} \quad (3.15)$$

which is then rewritten as a diagonal mass matrix \mathbf{M} . The masses are fictitious and included strictly for computational needs. The lumped mass matrix formulation presented is based on stiffness in each node which according to Barnes [19] renders any time step Δt admissible. In this study $\Delta t = 1$ is used.

The nodal velocities and nodal coordinates are updated via

$$\mathbf{v}_{t+\Delta t/2} = \mathbf{v}_{t-\Delta t/2} + \Delta \mathbf{v}_t \quad (3.16)$$

which, with $\Delta \mathbf{v}_t$ replaced according to equation 3.3 is

$$\mathbf{v}_{t+\Delta t/2} = \mathbf{v}_{t-\Delta t/2} + \Delta t \mathbf{M}^{-1} \mathbf{r} \quad (3.17)$$

Finally, the nodal coordinates are updated according to

$$\mathbf{x}_{i,t+\Delta t} = \mathbf{x}_{i,t} + \Delta t \mathbf{v}_{t+\Delta t/2} \quad (3.18)$$

The convergence criterion used in this study is a limit on kinetic energy

$$E_k = \frac{1}{2} \mathbf{M} \mathbf{v}^2 < E_{k,tol} \quad (3.19)$$

where the solution is said to have converged to static equilibrium once the kinetic energy drops below the tolerance limit. The limit used in this study is determined through convergence analysis of the solution for successively lowered $E_{k,tol}$.

To increase convergence speed, kinetic damping is introduced to set velocities to zero when a peak in kinetic energy is detected, see further section 3.5.4.

3.5.3 Data Structure

The data structure described in equations 3.4-3.8 is valid in general. This data structure is used for both cases studied here. The case with a single boundary uses this structure precisely. The case with an interior boundary uses a coordinate vector of the boundary nodes according to

$$\mathbf{x}_f = \begin{bmatrix} \bar{\mathbf{x}}_{fo} \\ \bar{\mathbf{x}}_{fi} \\ \bar{\mathbf{y}}_{fo} \\ \bar{\mathbf{y}}_{fi} \\ \bar{\mathbf{z}}_{fo} \\ \bar{\mathbf{z}}_{fi} \end{bmatrix} \quad (3.20)$$

where subscript o refers to outer boundary and i refers to interior boundary. This gives the following structure of \mathbf{C}_f

$$\mathbf{C}_f = \begin{bmatrix} [\bar{\mathbf{C}}_{fo} \bar{\mathbf{C}}_{fi}] & & \\ & [\bar{\mathbf{C}}_{fo} \bar{\mathbf{C}}_{fi}] & \\ & & [\mathbf{C}_{fo} \bar{\mathbf{C}}_{fi}] \end{bmatrix} \quad (3.21)$$

The coordinate vector, \mathbf{x} is then defined as in equation 3.7 and \mathbf{C} as in 3.8. Other data structures are possible as long as the coordinate vector and branch node matrix are structured in the same way.

3.5.4 Kinetic Damping

In structural dynamics, a common form of damping is the Rayleigh damping, where a damping matrix is formulated as a "linear combination of the global mass and stiffness matrices" [24]. Since dynamic analysis is not the objective of using dynamic relaxation in this study, the simplified method of kinetic damping may be employed.

The principle of kinetic damping is illustrated in Figure 3.10. The kinetic energy is calculated at time steps A, B, and C and the energy levels are registered. When the kinetic energy at C, time step $t + \Delta t/2$, is less than previous at B, time step $t - \Delta t/2$, a kinetic energy peak has been passed.

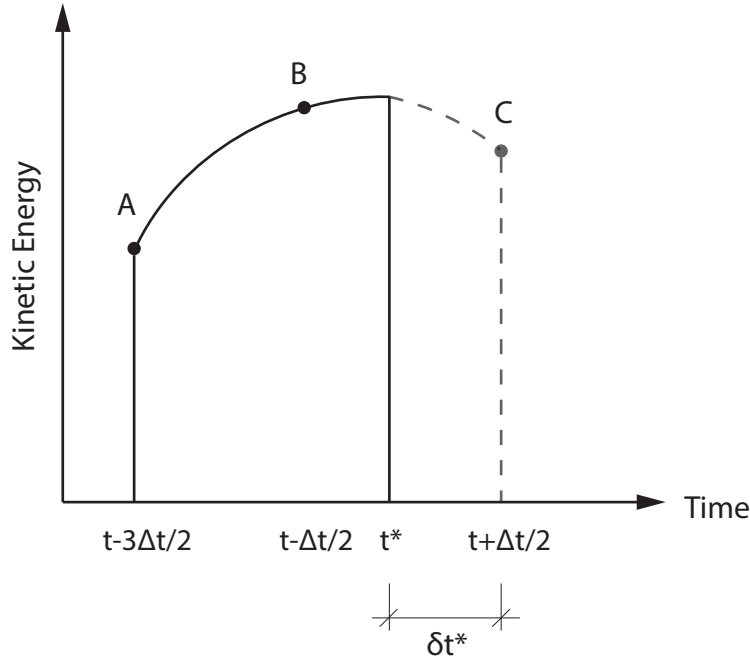


Figure 3.10: Principle of Kinetic Damping. Figure recreated from [19]

To find the real energy peak between B and C the time difference δt^* is calculated according to Barnes [19] as

$$\delta t^* = \Delta t \cdot \frac{E}{E - D} = \Delta t \cdot q \quad (3.22)$$

where, $E = B - C$ and $D = A - B$.

The time difference δt^* is used to calculate the nodal coordinated at the energy peak at time t^* as

$$\mathbf{x}_{i,t^*} = \mathbf{x}_{i,t+\Delta t} + \delta t^* \cdot \mathbf{v}_{t+\Delta t/2} \quad (3.23)$$

When the nodal coordinates are updated the residual forces and kinetic energy is set to zero before the process is restarted.

The process of kinetic damping yields an energy peak diagram which is shown in principal in Figure 3.11.

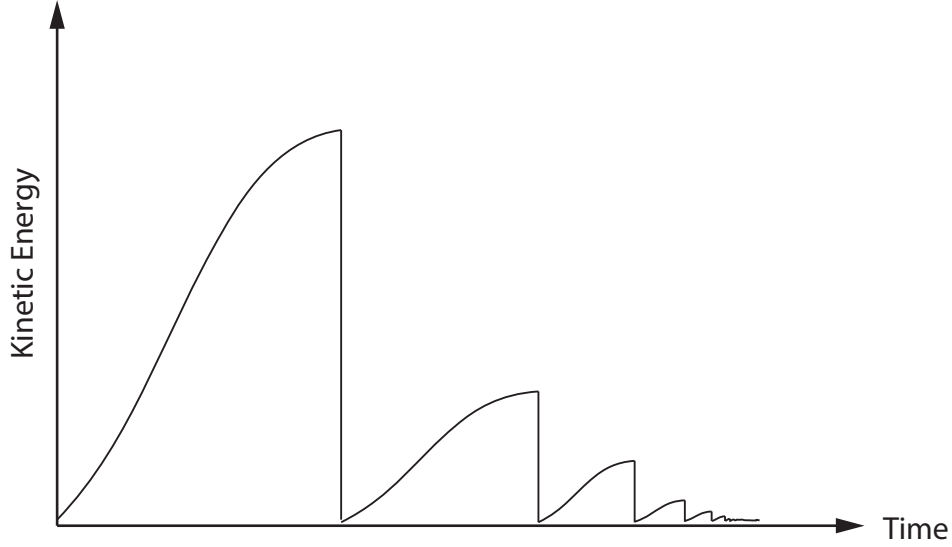


Figure 3.11: Kinetic Energy peaks. Figure recreated from [3]

As described by Barnes [3] the initial high energy peak represents high frequency modes which are damped out resulting in lower energy peaks corresponding to the overall structural shape. At the low energy peaks the structure approaches convergence and the low energy peaks are associated with low frequency modes. The dynamic relaxation meets convergence when the kinetic energy drops beneath a certain kinetic energy tolerance $E_{k,tol}$.

3.5.5 Loaded Network

The load considered in the form finding process is the sum of the uniformly distributed loads which are expected on the structure. In this case these loads are: self weight from the steel members, self weight of the glass panels and snow load. The loads are defined as follows.

The self weight of the steel is defined as

$$Q_{steel} = \sum_i A_{steel} L_i \rho_{steel} g \quad (3.24)$$

where A_{steel} is the cross section area of the steel members, L_i is the current length of member i . The density of steel, ρ_{steel} is here assumed to be 7800 kg/m^3 and g is the gravitational constant.

In a similar manner the weight of the glass panels are calculated as

$$Q_{\text{glass}} = A_{\text{glass}} \gamma_{\text{glass}} \quad (3.25)$$

where γ_{glass} is the weight of the glass panels in kN/m^2 . The glass load, γ_{glass} , used throughout this study is set to 0.6 kN/m^2 , slightly higher than the glass load used by Adriaenssens et al. [5] as a safe side approximation. The glass surface area of the current grid is approximated by

$$\begin{aligned} A_{\text{glass}} &= N_{\text{div}}^2 L_{\text{mean}}^2 \\ L_{\text{mean}} &= \text{mean}(L_r \geq 0.95 \frac{40}{N_{\text{div}}}) \end{aligned} \quad (3.26)$$

where $2N_{\text{div}}^2$ is the number of glass panels and $\frac{L_{\text{mean}}^2}{2}$ is the area of each panel under the assumption that height and base of each triangle in the grid is equal to L_{mean}

The snow load is defined by

$$Q_{\text{snow}} = A_{\text{planar}} s_k \quad (3.27)$$

where A_{planar} is the planar grid area and $s_k = 2.0 \text{ kN/m}$ is the characteristic snow load in Uppsala according to Appendix C of EN1991-1-3 [25].

The total load is then given by

$$Q = Q_{\text{steel}} + Q_{\text{glass}} + Q_{\text{snow}} \quad (3.28)$$

This load is then distributed to each interior node i according to

$$p_i = \frac{a_i}{A_{\text{planar}}} Q \quad (3.29)$$

where a_i is the Voronoi area of node i as described in section 3.5.6. The nodal loads will increase with the structure height as the length of the grid and the glass panel area increases and give a good approximation of loads to be expected on the realised structure.

3.5.6 Voronoi Distribution

The nodal loads on the grid shell are represented using a Voronoi diagram. Figure 3.12 shows the planar Voronoi distribution for a set of nodes in a grid shell. The horizontal area of the grid shell boundary is divided into partitions, Voronoi areas, enclosing the planar projection of the nodes in the grid. The area is divided so that any point on a Voronoi area is closer to the node projection than to any other node in the grid. Klein [26] gives a more rigorous definition. The resulting Voronoi area represents the tributary area of each node when the loads are distributed to the nodes. This tributary area is considered constant and equal to the horizontal Voronoi area throughout form finding.

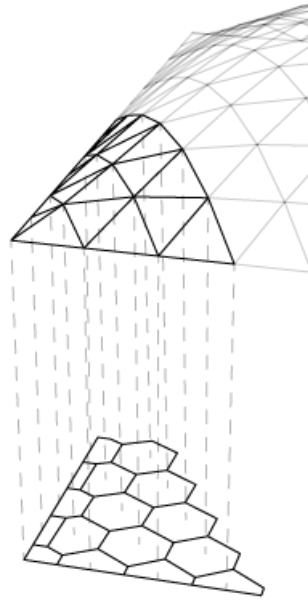


Figure 3.12: Part of a grid shell above the voronoi areas of the grid nodes below

3.5.7 Self-stressed Network

Dynamic relaxation can also be used to form find self stressed networks, that is networks where a prestress is applied in the members and an equilibrium configuration is form found under this condition. In this study a prestressed network is used as an initial configuration when the interior boundary does not lie on the same plane as the outer geometry. This is done by changing the nodal coordinates of the interior boundary and finding the self stressed equilibrium form. When running dynamic relaxation with prestress, the physical stiffness corresponding to $\phi_E = 1$ is used. The initial internal forces are set to a trivial value, $\mathbf{f}_0 = 1$ and the nodal loads are set to zero. Otherwise, the algorithm is applied according to section 3.5.2.

3.5.8 Algorithm Convergence

To judge when static equilibrium is achieved in Dynamic relaxation some tolerance of convergence must be introduced. Several criteria are suggested by Block and Veenendaal [22] and in this study the total kinetic energy is used as defined by equation 3.19. To

determine an acceptable level of kinetic energy in the algorithm, a convergence analysis is performed. The tolerance, $E_{k,\text{tol}}$ is successively lowered and the resulting structure height is noted along with maximum displacement as calculated by the FEM script. To further verify the convergence of $E_{k,\text{tol}}$ the membrane action, $W_{\text{axial}}/W_{\text{total}}$ is evaluated and verified in the commercial FEM software Comsol Multiphysics [27] by geometric non-linear analysis.

3.6 Prescriptive Dynamic Relaxation

The issue of unique members in grid shells produced by typical form finding techniques can be treated by a modification of relaxed member lengths prior to strain calculation. Such a process is Prescriptive dynamic relaxation first developed by Bagrianski and Halpern [23]. Typically in form finding, the form found shape is considered to be without strain from the form finding process unless prestress is needed to obtain the shape. This implies that it is possible to adjust the relaxed lengths of the members in the equilibrium solution and then re-run the algorithm. Adapted from [23] the process for bar elements is as follows.

At equilibrium, member strain is calculated according to equation 3.13, reprinted here for clarity

$$\varepsilon = \frac{l_{e,i} - l_{r,i}}{l_{r,i}}$$

where $l_{e,i}$ is the branch length at equilibrium of branch i and $l_{r,i}$ is the relaxed length, where the strain is zero, of the same member.

If a certain length, l_p is prescribed, the strain can also be formulated with regards to an updated relaxed length l_r^* as

$$\varepsilon = \frac{l_{p,i} - l_{r,i}^*}{l_{r,i}^*} \quad (3.30)$$

Setting the strains equal gives an expression of the updated relaxed lengths as

$$l_{r,i}^* = \frac{l_{p,i} l_{r,i}}{l_{e,i}} \quad (3.31)$$

This process gives an outer loop which finds equilibrium by dynamic relaxation and forces the members to approach the prescribed lengths.

Convergence of this outer loop is defined by

$$\|\max(\mathbf{l}_e - \mathbf{l}_p)\| < l_{\text{tol}} \quad (3.32)$$

The additions to the algorithm is minor but convergence is not a straight forward process since it is unclear how the prescribed lengths should be set in order to enable a structure made of prescribed lengths while maintaining structural performance. Bagrianski and Halpern [23] prescribes the lengths to be equal and in at least one case allows the supports to move in order to be able to meet this length requirement. However, this in addition requires the addition of applied forces at the supports to simulate support reactions and maintain equilibrium. This introduces several new parameters in how nodes are supported and how supports should be modelled.

In this study, three boundary conditions are tested while the lengths are set in the same way for each case. The lengths are set in different length groups according to their length in the planar grid which gives four different groups, see the grid in Figure 3.4. The prescribed lengths in these groups are set to the mean value of the lengths in the same group when equilibrium is achieved in a dynamic relaxation without prescriptive lengths. This will shorten the elements longer than the mean length and elongated the shorter ones. The boundary conditions tested are:

- All boundary nodes pinned
- Corner nodes pinned and remaining boundary nodes as rollers in the xy-plane
- Corner nodes pinned and all other nodes free

To simulate rollers along the boundary during dynamic relaxation the nodal velocities in z-direction of the boundary nodes are set to zero after calculation according to equation 3.17. In the case with only corner nodes pinned, the nodes on the edges of the boundary are treated as interior nodes but with nodal load zero.

In all cases only the grid shell supported on a single boundary is considered since the inner boundary gives rise to additional geometric complexity at the inner boundary.

3.7 Finite Element Method

The finite element method, FEM, is used to evaluate structural performance of the form found shapes. A linear solver for 3D beam element is scripted and coupled to the form finding algorithm. This scripted FEM solver enables quick evaluation of different options without the need to pre process geometry, loads etc. in an external FEM software. However, an external software is also deployed to study nonlinearities, different load cases and support conditions. The commercial software is also used to verify the script.

3.7.1 FEM Theory

The theory used to produce the FEM script is briefly presented below. The presentation assumes prior knowledge of the finite element method as described by eg. Cook et al. [24]. For further theory on FEM used in the nonlinear simulation refer to Cook et al. [24] and the Comsol user manual [27].

In structural mechanics the basic equation of a linear FEM formulation is

$$\mathbf{K}\mathbf{d} = \mathbf{f} \quad (3.33)$$

where \mathbf{K} is the global stiffness matrix, \mathbf{d} is the vector of nodal displacements and \mathbf{f} is the load vector acting on the structure.

The stiffness matrix \mathbf{K} is assembled from element stiffness matrices such that stiffness are added where more than one element share a common node. The local element stiffness matrix of a 3D Euler Bernoulli two node beam is taken from Cook et al. [24].

$$\mathbf{k} = \begin{bmatrix} X & 0 & 0 & 0 & 0 & 0 & -X & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_1 & 0 & 0 & 0 & Y_2 & 0 & -Y_1 & 0 & 0 & 0 & Y_2 \\ 0 & 0 & Z_1 & 0 & -Z_2 & 0 & 0 & 0 & -Z_1 & 0 & -Z_2 & 0 \\ 0 & 0 & 0 & S & 0 & 0 & 0 & 0 & 0 & -S & 0 & 0 \\ 0 & 0 & -Z_2 & 0 & Z_3 & 0 & 0 & 0 & Z_2 & 0 & Z_4 & 0 \\ 0 & Y_2 & 0 & 0 & 0 & Y_3 & 0 & -Y_2 & 0 & 0 & 0 & Y_4 \\ -X & 0 & 0 & 0 & 0 & 0 & X & 0 & 0 & 0 & 0 & 0 \\ 0 & -Y_1 & 0 & 0 & 0 & -Y_2 & 0 & Y_1 & 0 & 0 & 0 & -Y_2 \\ 0 & 0 & -Z_1 & 0 & Z_2 & 0 & 0 & 0 & Z_1 & 0 & Z_2 & 0 \\ 0 & 0 & 0 & -S & 0 & 0 & 0 & 0 & 0 & S & 0 & 0 \\ 0 & 0 & -Z_2 & 0 & Z_4 & 0 & 0 & 0 & Z_2 & 0 & Z_3 & 0 \\ 0 & Y_2 & 0 & 0 & 0 & Y_4 & 0 & -Y_2 & 0 & 0 & 0 & Y_3 \end{bmatrix} \quad (3.34)$$

with

$$\begin{aligned} X &= \frac{AE}{L} & Y_1 &= \frac{12EI_z}{L^3} & Y_2 &= \frac{6EI_z}{L^2} \\ Y_3 &= \frac{4EI_z}{L} & Y_4 &= \frac{2EI_z}{L} & S &= \frac{GK}{L} \end{aligned} \quad (3.35)$$

All terms Z_i are defined as Y_i by an exchange of subscript z for y.

The element stiffness matrix in equation 3.34 is formulated with regards to a local coordinate system. This element is derived from a 3rd order interpolation of transverse deformations and 1st order interpolation of axial and torsional deformations. The element only handles Saint Venant torsion why only closed sections with negligible warping are used in analysis.

Prior to assembly of the global stiffness matrix, each stiffness must be transformed from local to global coordinates according to

$$\mathbf{k}_{\text{global}} = \mathbf{T}^T \mathbf{k} \mathbf{T} \quad (3.36)$$

where \mathbf{T} is a [12x12] matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{t} & & & \\ & \mathbf{t} & & \\ & & \mathbf{t} & \\ & & & \mathbf{t} \end{bmatrix} \quad (3.37)$$

with submatrix \mathbf{t} taken from Ghali et al. [28]

$$\mathbf{t} = \begin{bmatrix} \lambda_{x*x} & \lambda_{x*y} & \lambda_{x*z} \\ \lambda_{y*x} & \lambda_{y*y} & \lambda_{y*z} \\ \lambda_{z*x} & \lambda_{z*y} & \lambda_{z*z} \end{bmatrix} \quad (3.38)$$

Each element in \mathbf{t} refers to a direction cosine where, λ_{a*b} is the cosine of the angle between local axis a and global axis b , see section 3.7.2 for details on computing local axis orientations.

The system stiffness matrix \mathbf{K} is then assembled by placing submatrices of $\mathbf{k}_{\text{global}}$ in the degrees of freedom of each element. Element numbering is ordered by \mathbf{C} . The branch node matrix also defines element start points as $C_{ij} = -1$ and element end points as $C_{ij} = 1$ in accordance with equation 3.5 which is necessary to correctly couple elements in common nodes.

Nodal displacements are then solved from equation 3.33 and the total work done by strain in the elements is given by

$$W_{\text{total}} = \frac{1}{2} \mathbf{d}^T \mathbf{K} \mathbf{d} \quad (3.39)$$

Axial work by one element is given by

$$w_{\text{axial},i} = \int_0^L \frac{1}{2} E \varepsilon_x^2 A dx \quad (3.40)$$

where ε_x is defined by transforming global translations to local x direction and then calculating strain as length change divided by initial length according to

$$\varepsilon_x = \frac{(u_2 - u_1) * \lambda_{x*x} + (v_2 - v_1) * \lambda_{x*y} + (w_2 - w_1) * \lambda_{x*z}}{L} \quad (3.41)$$

With the axial strain known, the axial forces are given by

$$N = EA\varepsilon_x \quad (3.42)$$

Total axial work is then the sum of axial work over all elements, i

$$W_{\text{axial}} = \sum_i w_{\text{axial},i} \quad (3.43)$$

3.7.2 Beam Orientation

Prior to assembly of the global stiffness matrix \mathbf{K} , element stiffness must be transformed to global stiffness according to equation 3.36. To establish the local beam system, the following procedure is applied. The coordinates of the vector defining the local x axis is given by the coordinate difference according to equation 3.9. The vector to define local y is then calculated according to

$$\mathbf{y}_{\text{local}} = \begin{cases} \begin{bmatrix} x_{\text{local},x} \\ x_{\text{local},y} \\ x_{\text{local},z} \end{bmatrix} \otimes \begin{bmatrix} x_{\text{local},x} \\ x_{\text{local},y} \\ 0 \end{bmatrix}, & \text{if } x_{\text{local},z} \neq 0 \\ \begin{bmatrix} -x_{\text{local},y} \\ x_{\text{local},x} \\ 0 \end{bmatrix}, & \text{otherwise} \end{cases} \quad (3.44)$$

where the cross product is used to obtain a vector in the global xy -plane. An exception occurs if the local x already lies in the global xy -plane in which case the local x axis is rotated 90 degrees to obtain local y . To handle numerical errors the condition in equation 3.44 is replaced by

$$\|x_{\text{local},z}\| < 1 \cdot 10^{-6} \quad (3.45)$$

The local z -axis is then calculated by the cross product of local x and y axes.

The principle of equation 3.44 is illustrated in Figure 3.13

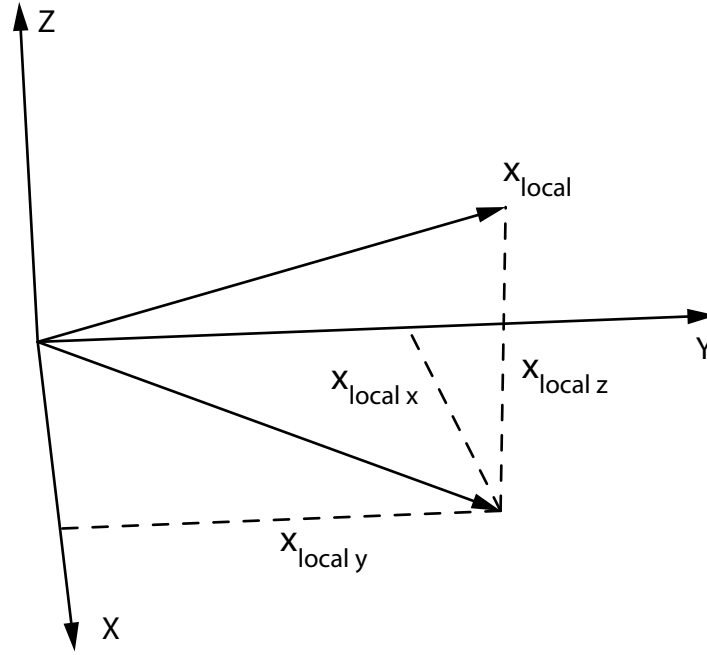


Figure 3.13: Local x -axis, x_{local} and components in global coordinate directions

3.7.3 Commercial Software

The commercial FEM software used is Comsol Multiphysics v.5.2 and v.5.3 from [27]. The analysis is performed in the structural mechanics module using Euler-Bernoulli beam elements. An exception to the use of Euler-Bernoulli beams is the analysis where imperfections are introduced as outlined in section 3.9.3. To introduce the imperfections, a Timoshenko beam formulation is used to achieve a state of zero stress with applied imperfections as suggested by Lejon at the Comsol support [29].

3.8 Optimization

The parametric study is set up to achieve a number of possible solutions within the studied parameter ranges. To determine which of these possible solutions is the best requires some form of evaluation of the alternatives. A particular project will often be constrained in some ways, for example by legal regulations on structure height or by structural demands such as inability to carry certain types of loads. However, it is unlikely that the solution is constrained in such a way that only a single solution is possible. Then, some form of optimization can be employed to single out the best solution according to the preferences set by the project.

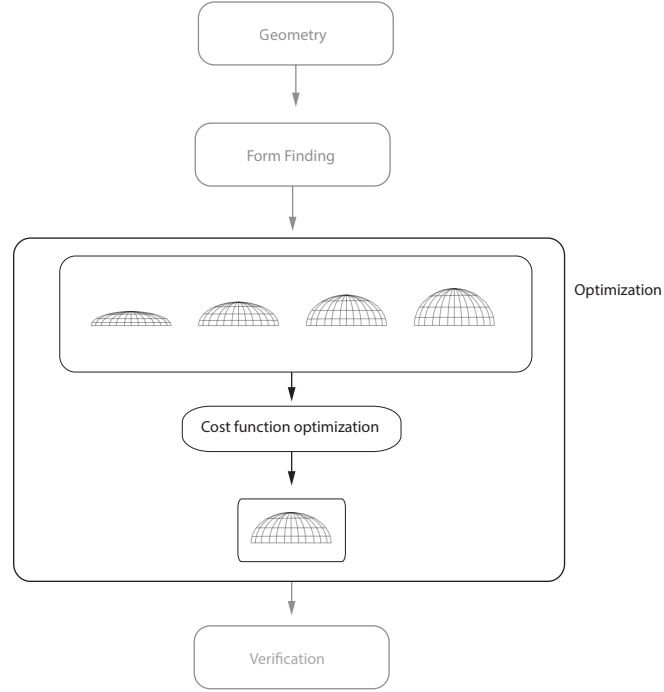


Figure 3.14: Optimization part of study method

In general terms the optimization employed here can be expressed as

$$Optimum \begin{cases} minimize(C) \\ \text{subject to} \begin{cases} \text{Equilibrium} \\ \text{Member buckling} \end{cases} \end{cases} \quad (3.46)$$

where Equilibrium is defined as in equation 3.19 by

$$E_k = \frac{1}{2} \mathbf{M} \mathbf{v}^2 < E_{k,tol}$$

and the Member buckling constraint is formulated according to 6.3.1 in EN1993-1-1 [30] as

$$\eta_i = \frac{N_i}{N_{b,Rd,i}} \leq 1 \quad (3.47)$$

for all members i where

$$N_{b,Rd} = \chi A f_y \quad (3.48)$$

with χ being the reduction factor with regards to column buckling. The buckling constraint is formulated as column buckling of individual members rather than global buckling. The member buckling check is easy to implement while a linear global buckling solver lies beyond the scope of this study. The global buckling is briefly studied as described in section 3.9.3.

In this report the optimization will be performed manually. The scope of the process is not to test the result of an optimization algorithm but to show the possibility to evaluate different form found alternatives by a set of criteria. Further, the results are intended to show that different solutions will be "optimal" depending on how the criteria are weighed against each other.

The process is explained in greater detail below.

3.8.1 Optimization Procedure

The optimization is divided such that some solutions may be excluded based on simple analysis before more detailed studies are carried out. The first step is to run the form finding algorithm and extract optimization data for multiple solutions by a successive reduction of ϕ_E . This data is then evaluated in a cost function described in section 3.8.2.

The solutions with the lowest costs in each case will then be studied in a geometrically nonlinear FEM simulation with different boundary conditions presented in section 3.9.2. Based on these results a second order cost will be established, defined as

$$C^{II} = \frac{d_{\max}^{II}}{d_{\max}^I} \quad (3.49)$$

The cost evaluated targets displacements since displacements will capture second order effects in the nonlinear solution. These effects are critical in form found structure. This second step is performed outside the form finding loop since a nonlinear FEM solver is computationally expensive in an iterative form finding algorithm. Further, nonlinear analysis of solutions which are not promising to start with are unlikely to give reason to consider them anyway.

3.8.2 Cost Function

To evaluate possible form found structures according to given criteria a cost function of the studied variables is formulated. A formulation similar to eq. (1.2) in Christensen and Klarbring [31] is applied as

$$C = a_1 * C_{\text{steel}}^2 + a_2 * C_h^2 + a_3 * C_D^2 + a_4 * C_{R_h}^2 \quad (3.50)$$

with

$$\begin{aligned} \sum_i a_i &= 1 \\ C_i &= \frac{1}{\text{mean}(A)} A \end{aligned} \quad (3.51)$$

where a_i is the weight of cost i and C_i is cost i which is the value of any parameter A normalised by the mean of the variable to ensure that the weights of the individual parameters determines the relative importance of each cost. Each cost in equation 3.50 is squared to accentuate differences between different solutions.

To further penalise costs in the optimization, the relation between costs can be modified. One way is to change relative weights according to 3.51. However, this will increase the cost equally across a parameter range which may not be what is intended. For instance, even if an increased height of the structure makes it sensitive to buckling, the cost should not increase before a thresh hold is reached after which the buckling sensitivity rapidly increases. This is exemplified by introducing a weight for the cost of structural height which is a function of the height itself as

$$a_2 = \begin{cases} a_{2,0}, & h \leq h_{\text{tol}} \\ a_{2,0} + kh - kh_{\text{tol}}, & h > h_{\text{tol}} \end{cases} \quad (3.52)$$

where h_{tol} is a tolerance of the structural height after which the cost should increase more rapidly with height. The weight according to equation 3.52 increases linearly beyond h_{tol} as shown in Figure 3.15 with $h_{\text{tol}} = 4.5$ and $k = 0.05$.

When the height is penalised according to equation 3.52 the sum of the weights are no longer equal to 1 as in equation 3.51. The values of the tolerance height, h_{tol} and the slope, k are in this study set arbitrarily since there is insufficient information to rationally determine their values.

Two different cases of weights are used in this study. Equal weights of all costs without penalty for structural height and equal weights with penalty for structural height.

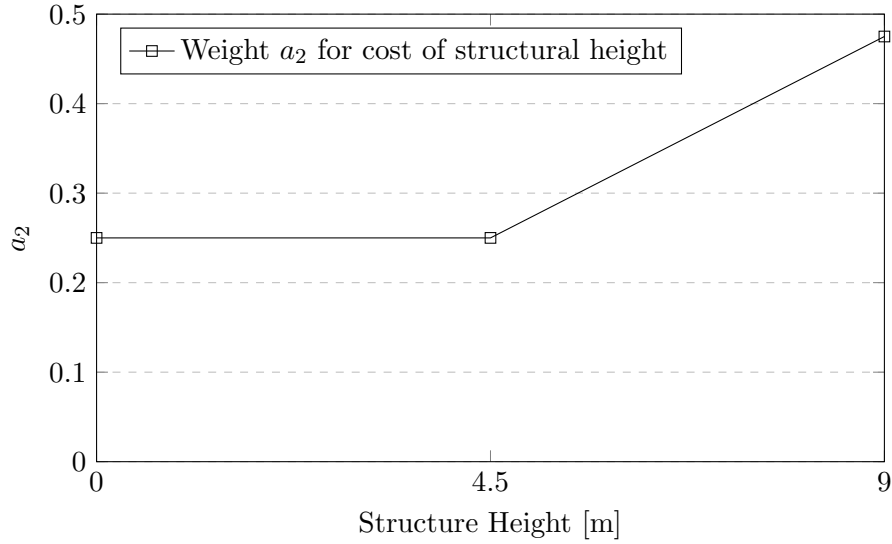


Figure 3.15: Weight of cost of structural height as a function of height

The costs considered by equation 3.50 are presented in Table 3.2. The steel volume is included since the amount of steel should be reduced. The structural height is considered a cost for two reasons: firstly, there may be legal or aesthetic restrictions on maximum structure height. Secondly, the risk for global instability is likely to increase with height, even if this is not strictly considered here. Maximum vertical displacement is a measure on how susceptible the structure will be to second order effects due to deformations. Finally, the sum of horizontal reaction forces is included since increased horizontal reaction forces will increase the forces on the supports and or ring beam of the shell.

Table 3.2: Legend of cost variables

Variable	Description
C_{steel}	Volume steel
C_h	Height of structure
C_D	Maximum vertical displacement
C_{R_h}	Sum of horizontal reaction forces

3.9 Verification and Structural Stability

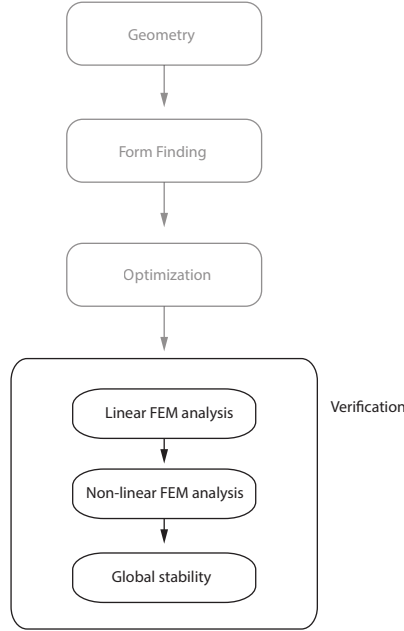


Figure 3.16: Verification scheme of form found shapes

The form found shapes chosen in the optimization according to section 3.8 are verified according to the scheme in Figure 3.16. The verification is performed in the commercial FEM software Comsol Multiphysics. Two different load cases and six different boundary conditions are tested in linear elastic analysis and in a geometrically nonlinear analysis with 200 load increments. The number of load increments is kept constant at 200 throughout the study. In addition global stability is studied through linear buckling and introduction of imperfections to the geometry.

Throughout verification the following material properties are used, a Young's modulus of 210 GPa, a density of 7800 kg/m³ and a yield stress of 355 MPa. The abbreviation VKR is a Swedish standard used for rectangular hollow sections of steel members, usually followed by numbers for height, width and thickness.

3.9.1 Load Cases

Two different load cases are studied for the form found geometries. The first includes, uniform snow load according to EN1991-1-3 [25] along with self weight of steel and glass. Secondly, a case of uneven snow load is applied. For the case with a single boundary this uneven load is snow load applied on half the structure as shown in Figure 3.18. For the inner boundary the uneven snow load is obtained by considering the interior boundary as a snow pocket as shown in Figure 3.20 and calculating an exceptional snow load according to EN1991-1-3. The loads are shown in Figures 3.17 - 3.20 and the loads are summarised in Table 3.3. The quantity, s is the snow load according to 5.3.5 and 5.3.4 of EN1991-1-3 [25] for single and inner boundary respectively. The quantity, s_e is the exceptional snow load in case of snow pockets according to 5.3.4 of [25] for the grid shell with inner boundary.

Table 3.3: Load summary used in verification

Structure	Load type	Value [$\frac{\text{kN}}{\text{m}^2}$]
Single boundary	s	1.28
Inner boundary	s	0.577
	s_e	2.56
Both	w_{glass}	0.6

In the FEM analysis the self weight of steel is applied as a gravity load. The load from snow and glass is applied as point loads at the nodes and distributed according to the Voronoi diagram as explained in section 3.5.6.

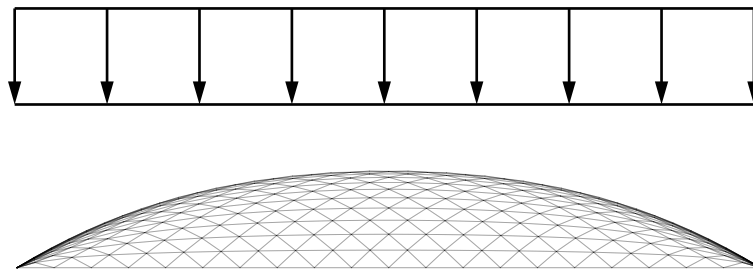


Figure 3.17: Load case 1 for grid shell with single boundary

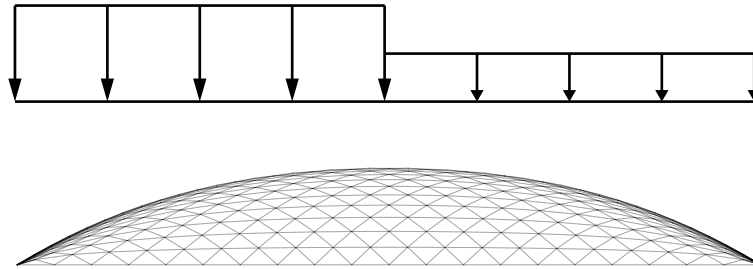


Figure 3.18: Load case 2 for grid shell with single boundary

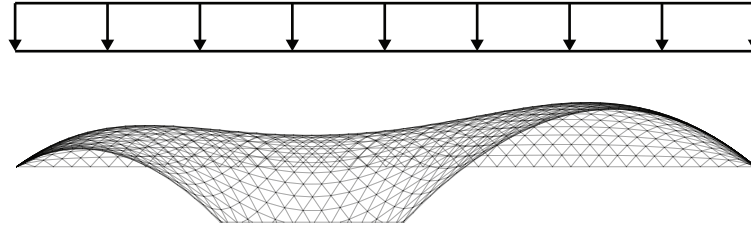


Figure 3.19: Load case 1 for grid shell with interior boundary

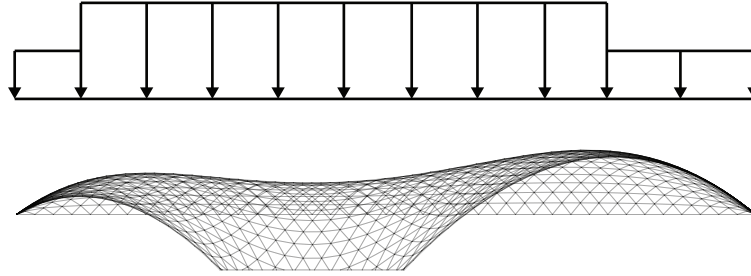


Figure 3.20: Load case 2 for grid shell with interior boundary

3.9.2 Boundary Conditions

In the form finding and optimization, all boundary nodes are pinned. These boundary conditions are schematically shown in Figure 3.21. In further analysis and verification, five other boundary conditions according to Figure 3.22 are treated. The case with interior boundary uses the same five cases as in Figure 3.22 with the addition of supports according to Figure 3.23 along the inner boundary.

To handle boundary conditions with rollers on some nodes, a stiff ring beam needs to be included to hold together the grid shell. In this study the choice of ring beam is arbitrary since the objective is not to study the optimum ring beam of a single grid shell. A ring beam with section VKR300x300x16 is used, that is a square section with 300 mm side length and 16 mm thickness. In a real application, the design of the ring beam will require considerable attention.

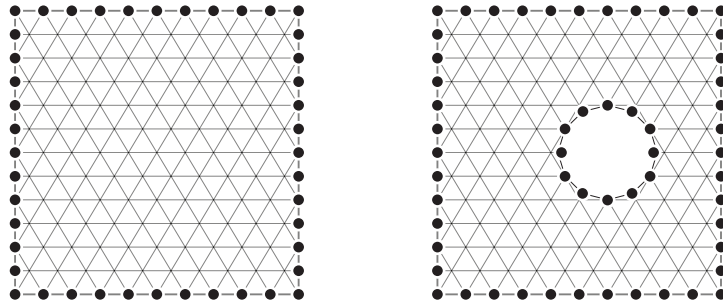


Figure 3.21: Boundary conditions used in form finding and in FEM script

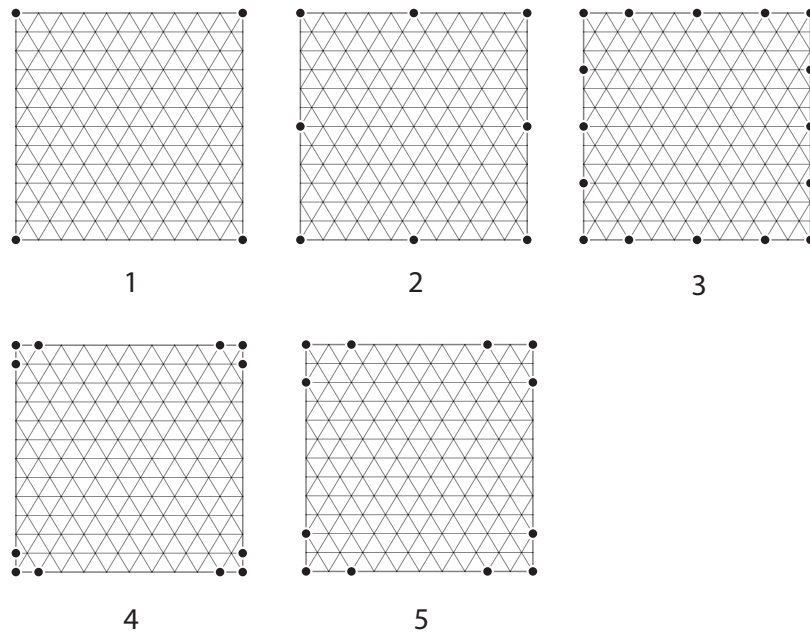


Figure 3.22: Schematic representation of studied BC in verification with numbers to identify them, a dot symbolises a pin support and unmarked nodes around the boundary are rollers

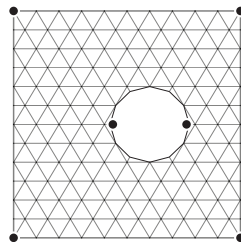


Figure 3.23: The support conditions of the inner boundary are the same for all cases with two nodes pinned and the others as rollers

3.9.3 Global Stability

Analysis of global stability is of major importance since the objective of form finding is to establish a shape which carries load primarily by compressive membrane action. In this study global stability is only studied by buckling of the structure with pinned boundary nodes and the uniform load used in form finding. The objective is to evaluate the proposed solution of the script and therefore the form finding load is used to enable comparison with the solution from the script.

At the time of this study Comsol v.5.3 was unable to properly assign cross section orientation to rectangular cross sections. A linear buckling solver for beam is not included in Comsol v.5.2. Therefore, the buckling analysis are performed for circular cross sections where the orientation can be arbitrary without affecting the results obtained. For these analyses the form finding script is run with circular pipe sections.

The stability is analysed in two steps. First, the linear buckling load of the structure is computed and secondly, the buckling mode is introduced as an imperfection to the structure prior to analysis. The linear buckling load is calculated according to the Comsol user manual [32] by solving the eigenvalue problem

$$(\mathbf{K} + \lambda \mathbf{K}_{\text{NL}}(\mathbf{u}_0))\mathbf{u} = \mathbf{0} \quad (3.53)$$

where \mathbf{K} is the stiffness matrix of the structure, \mathbf{K}_{NL} is the non linear stiffness matrix calculated for some displacement vector \mathbf{u}_0 resulting from a chosen load level and λ is the critical load level multiplier. With the design load applied to give \mathbf{u}_0 , the eigenvalue λ gives the safety factor against global buckling in a linear context.

Secondly, the buckling mode is introduced as a geometric imperfection to the structure. The mode is scaled to a maximum amplitude of 10 mm and applied prior to a geometrically non linear analysis. The results will give an indication of the sensitivity to imperfections.

No post buckling study, where structural behaviour is traced after initial buckling and softening, is performed. This type of study is hardly a part of preliminary design and further, the objective here is to avoid structural collapse.

3.9.4 Dynamic Analysis

The dynamic analysis of this study is restricted to computing the first natural frequency of the grid shells. To compute the natural frequencies of the shell a modified density of the steel members are calculated to include the self-weight of the glass as well as the steel itself. Equation 3.24 gives the self weight of the steel as

$$Q_{\text{steel}} = \sum_i A_{\text{steel}} L_i \rho_{\text{steel}} g$$

An expression for the load from steel and glass is

$$Q_{\text{steel}} + Q_{\text{glass}} = \sum_i A_{\text{steel}} L_i \rho_{\text{mod}} g \quad (3.54)$$

where ρ_{mod} is the sought after modified density. Replacing $\sum A_{\text{steel}} L_i g$ from the expression for the steel weight gives the modified density as

$$\rho_{\text{mod}} = \frac{Q_{\text{steel}} + Q_{\text{glass}}}{Q_{\text{steel}}} \rho_{\text{steel}} \quad (3.55)$$

The steel members are given this modified density and the natural frequencies are solved by an eigenfrequency solver in Comsol. The dynamic load to be expected on a grid shell roof is dynamic wind load. Only the first natural frequency is computed because of the low frequency of a typical wind load along with the limited scope of dynamic analysis in this study

3.10 Method Summary

The employed method is summarised in Figure 3.24 with its four components: Geometry, Form Finding, Optimization and Verification.

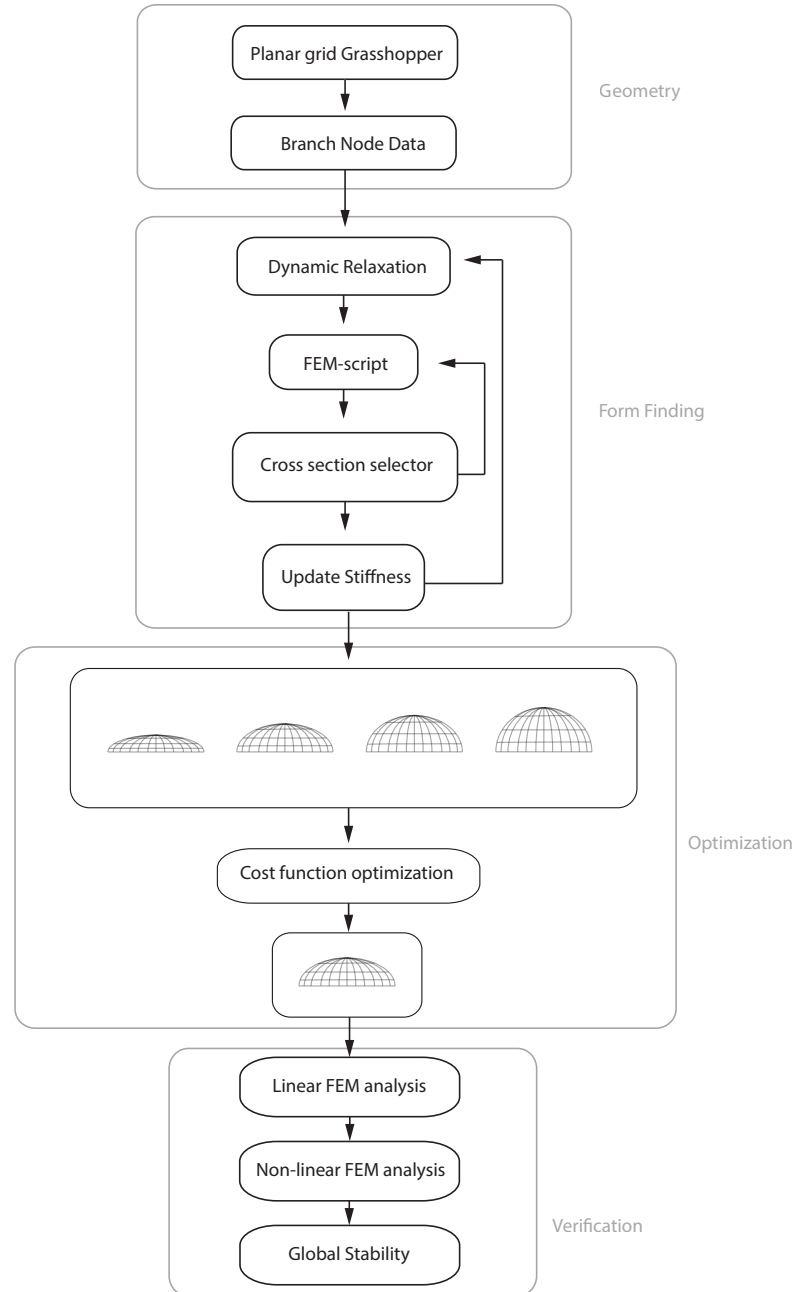


Figure 3.24: Flow chart summary of employed method

Chapter 4

Results

This chapter presents the results from the study in the following order. First, general form finding results are presented which include geometric capacity of the deployed method and initial analysis results. Then the FEM script is verified and the convergence of dynamic relaxation studied. This section is followed by the case studies where forms are chosen based on the optimization and verified by analysis. The issue of global stability is then briefly studied and finally, results of prescriptive dynamic relaxation are presented.

4.1 General Form Finding Results

This section presents general form finding results. A series of geometries generated with the form finding algorithm are presented followed by initial analysis.

4.1.1 Form

Form found shapes with varying boundaries are presented in Figures 4.2 and 4.4. The boundaries with the initial planar grids presented in Figure 4.1 and 4.3 show a successive deviation from the square boundary at top left in each Figure. The resulting forms illustrate the variety of shapes possible with the developed form finding algorithm.

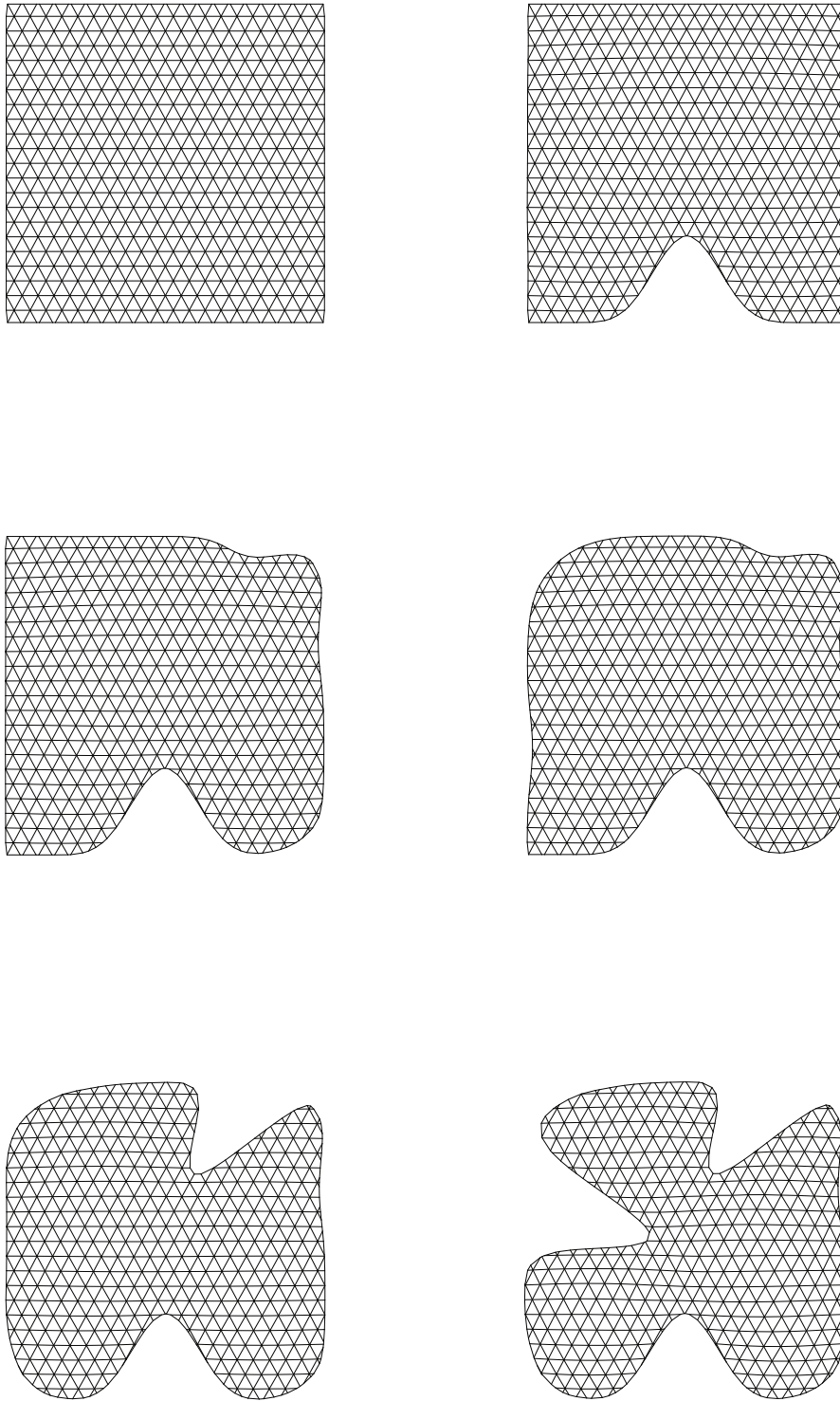


Figure 4.1: Boundary geometry and grid pattern for single boundary grid shells

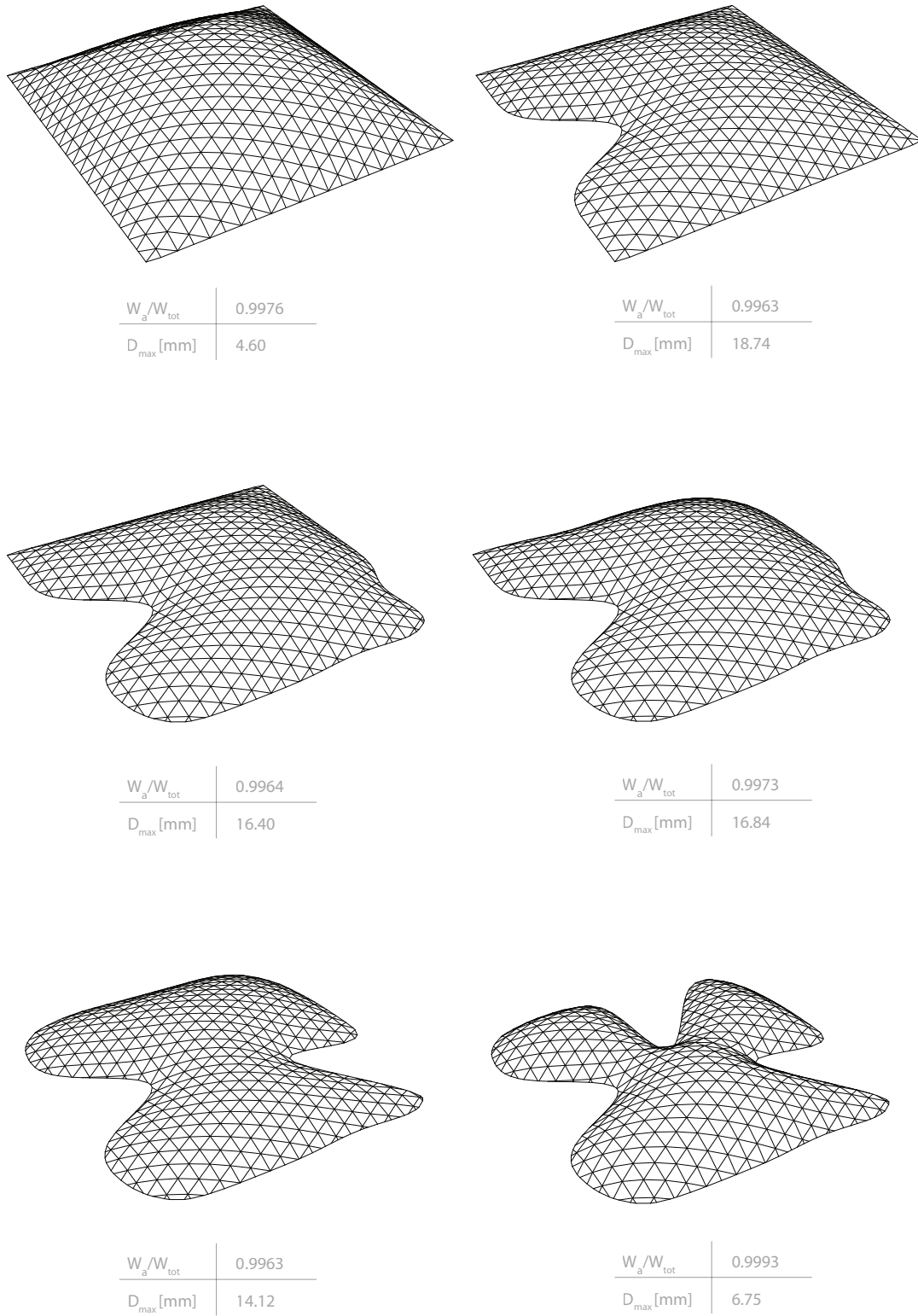


Figure 4.2: Form found single boundary grid shells with data on membrane action and displacement

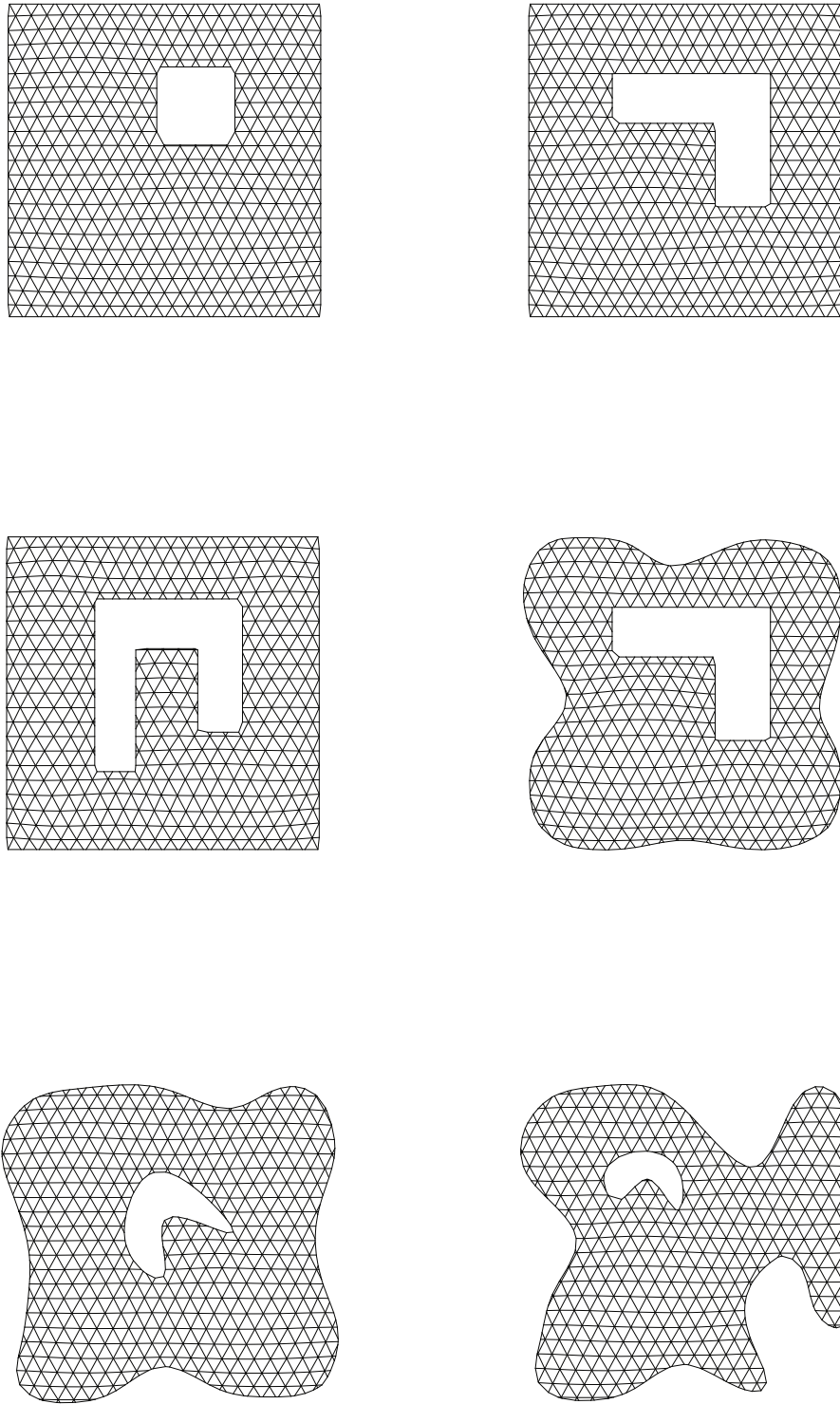


Figure 4.3: Boundary geometry and grid pattern for grid shells with interior boundary

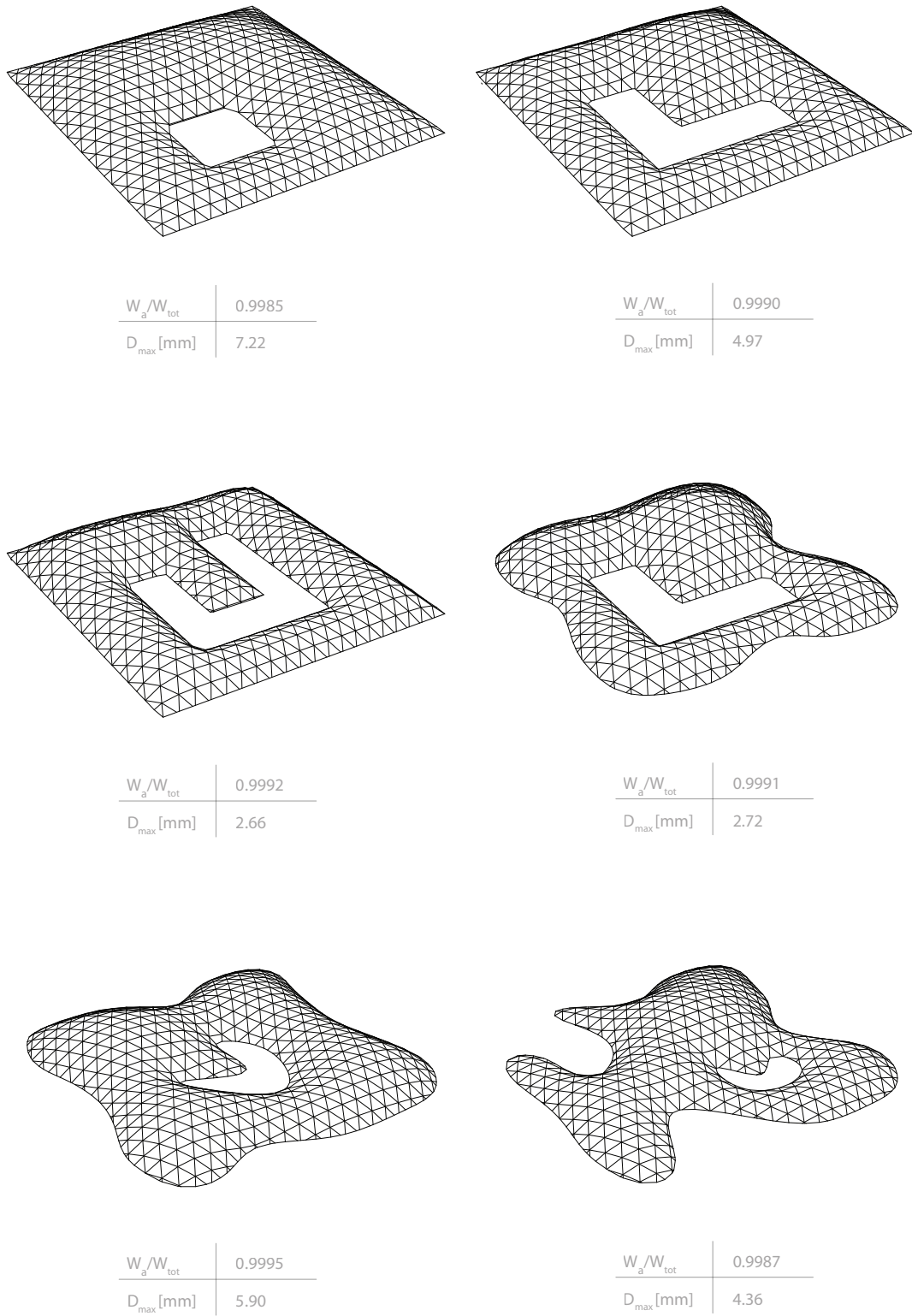


Figure 4.4: Form found shells with interior boundary along with data on membrane action and displacement

4.1.2 Case Study Form

In the case study, two forms will be treated, loosely based on the geometry of the proposed new town hall of Uppsala described in section 3.3.2 and 3.3.3. The form in the case of a single boundary is determined as shown in Figure 4.5 by form finding the shape from the planar grid.

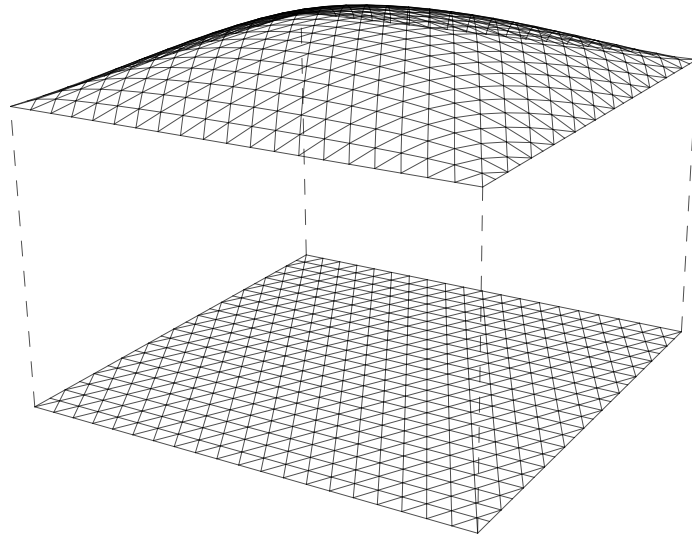


Figure 4.5: Process of dynamic relaxation for a single boundary grid shell

To form find the shell shape in the case of an interior boundary, a two step approach is deployed as shown in Figure 4.6. First the inner boundary is lowered to the level of the inner building and the self stressed form is found according to section 3.5.7 for this configuration. This intermediate geometry is then used as the initial geometry in form finding with vertical load applied.

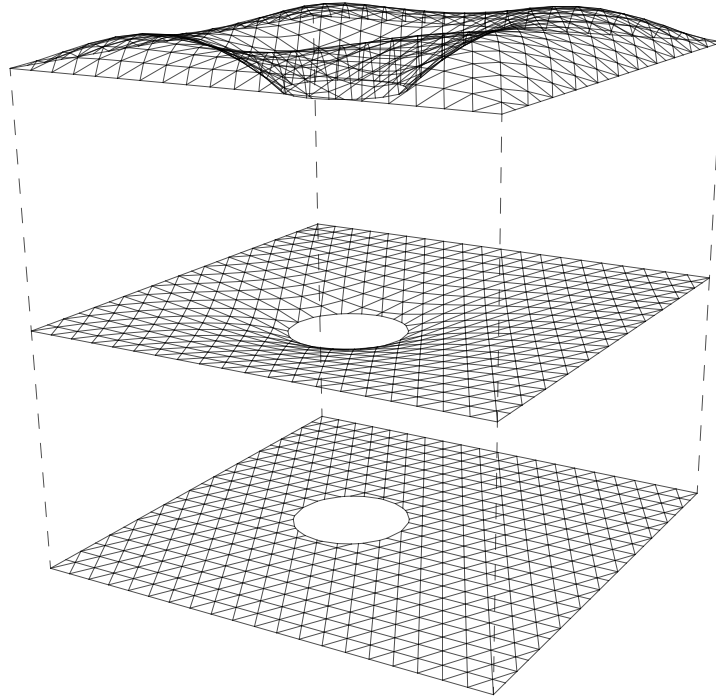


Figure 4.6: Process of dynamic relaxation for a grid shell with lowered interior boundary

The intermediate, self stressed state, is shown in an elevation in Figure 4.7. The form found shape with the interior boundary lowered resembles a cone with the top cut off. As seen in the Figure, the grid distortions are small from the operation, making this a good initial configuration when load is applied in the consecutive form finding step.

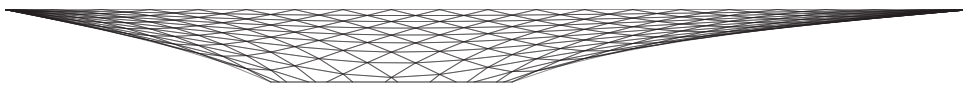


Figure 4.7: Elevation of self stressed intermediate configuration

4.1.3 Preliminary Studies

The distribution of vertical displacements and stresses of the two case geometries is presented to visualise how the grid shells behave when subjected to the form finding load. The vertical displacement field is presented in Figure 4.8 and 4.10 for case 1 and case 2 respectively. The stress distribution is shown in Figure 4.9 for a single boundary grid shell and in Figure 4.11 and for a grid shell with interior boundary.

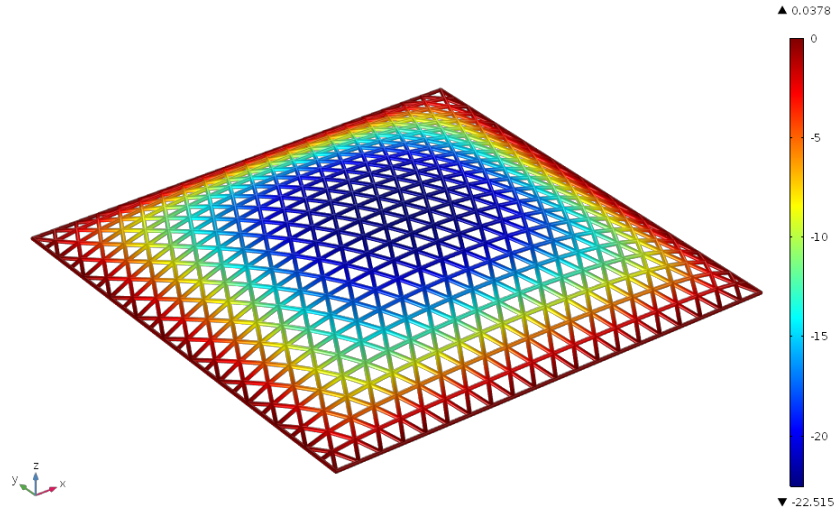


Figure 4.8: Comsol plot of vertical displacements of single boundary grid shell with 20 divisions. The legend unit is mm

The linear analysis yields an evenly distributed displacement field as expected for a structure with high degree of membrane action, see Figure 4.8. The stress plot in Figure 4.9 show that the highest stress levels appear around the sides of the structure close to the boundary. Worth noting is that the minimum stress levels appear at the boundary corners.

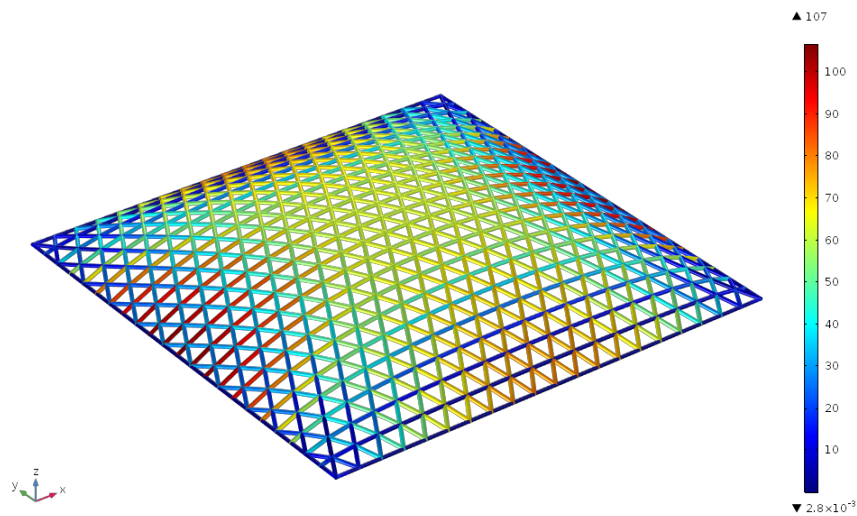


Figure 4.9: Von Mises stress in MPa of single boundary grid shell with 20 divisions

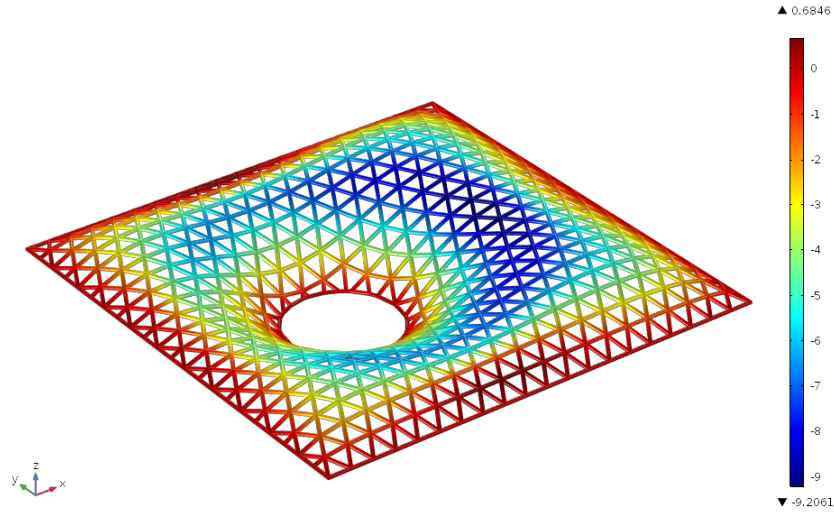


Figure 4.10: Vertical displacements in mm of grid shell with interior boundary and 20 divisions

For the structure with a circular inner boundary, the highest displacement appear at the symmetry line and not in the maximum span between the inner boundary and the corner of the outer boundary. The vertical displacements are presented in Figure 4.10. Figure 4.11 shows that the highest stress levels appear close to the inner boundary.

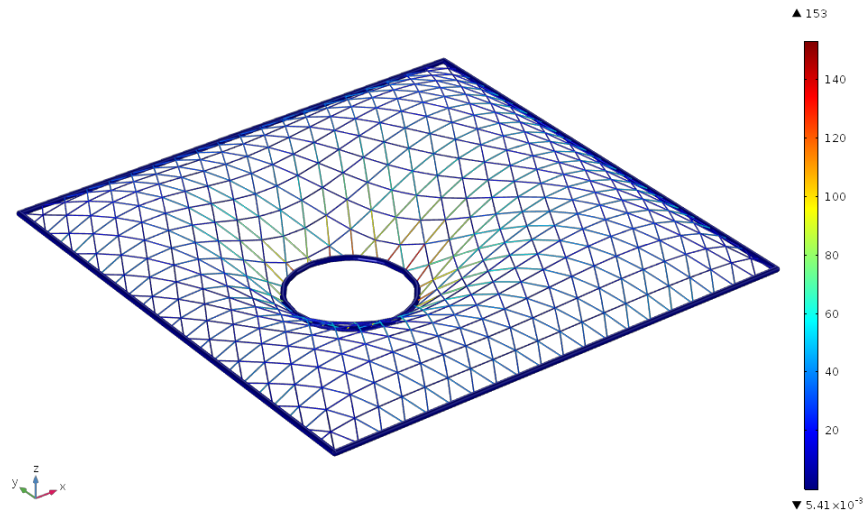


Figure 4.11: Von Mises stress in MPa of grid shell with interior boundary and 20 divisions

4.1.4 Boundary Conditions

Boundary conditions applied node-wise along the ring beam will produce highly different results in stress at the support nodes. Figure 4.12 shows the von Mises stress variation around the support at one corner. The support node is the node connecting to the short beam marked in red in the figure, in this case the red colour corresponds to a stress of 500 MPa. Here the nodal support is located where one member of the grid intersects the ring beam. This can be compared to Figure 4.13 where the support node is moved one node closer to the corner where three grid members intersect the ring beam. This support gives a maximum von Mises stress of 291 MPa.

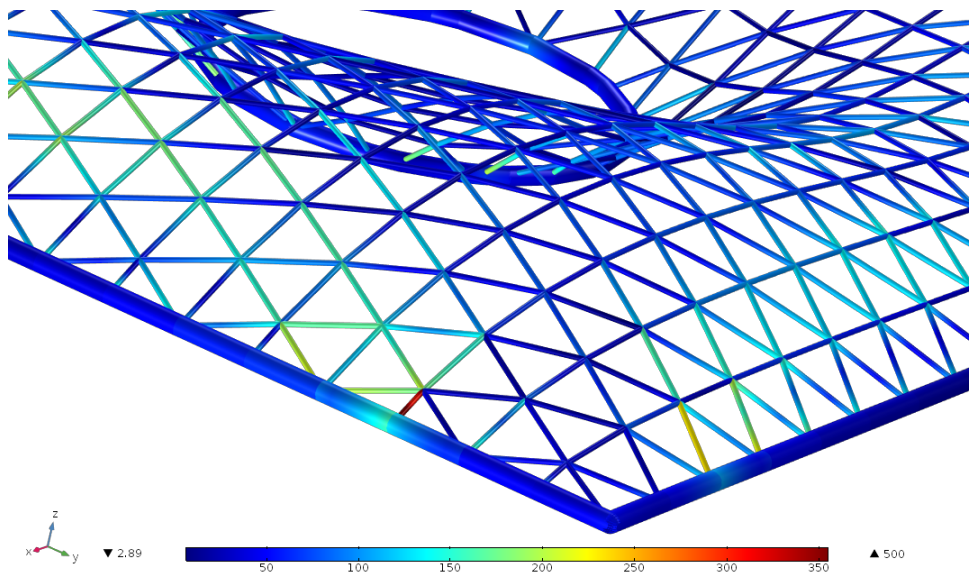


Figure 4.12: High stress at element at boundary

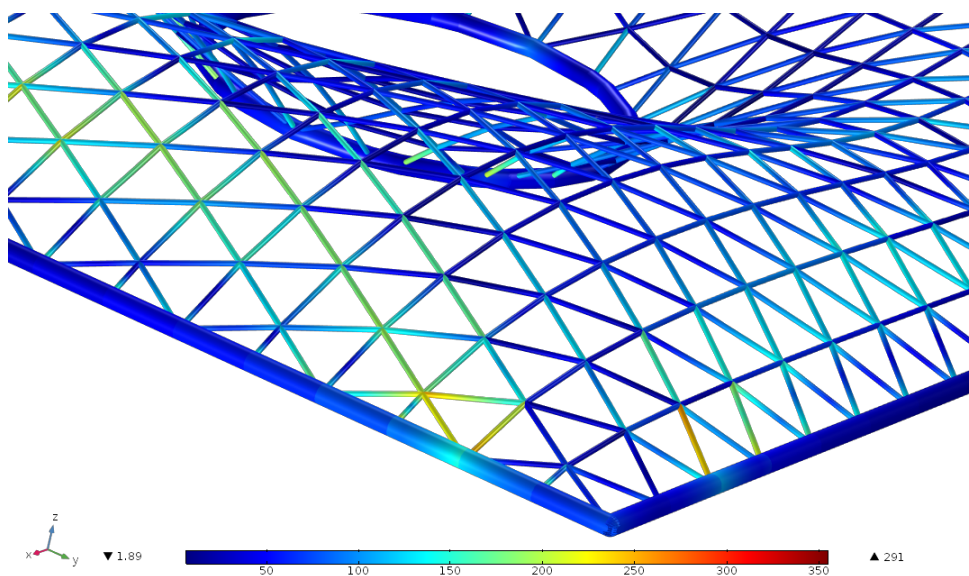


Figure 4.13: Low stress at element at boundary

4.2 FEM script Verification

The FEM script is verified by analysis of two grid shells with single boundary, with 20 and 40 pattern divisions, 2964 and 10704 DOF respectively. The number of degrees of freedom refers to a mesh of 1 element per beam, results for 2 elements per beam are included as a mesh convergence study. The results are presented in Table 4.1.

Table 4.1: FEM verification results

DOF	Result	FEM script (1 element/beam)	Comsol (1 element/beam)	Comsol (2 element/beam)
2964	d_{\max} [mm]	22.46	22.52	22.52
	Sum P [kN]	4179.5	4179.5	4179.5
	Sum R_v [kN]	4179.5	-	-
	N_{\max} [kN]	133.0	132.9	132.9
	$R_{h,\max}$ [kN]	152.0	151.9	151.9
10704	d_{\max} [mm]	35.75	35.77	35.77
	Sum P [kN]	4305.7	4305.7	4305.7
	Sum R_v [kN]	4305.7	-	-
	N_{\max} [kN]	73.74	73.74	73.74
	$R_{h,\max}$ [kN]	86.66	86.66	86.66

The greatest difference in results are in displacements. To compare this difference, the ratio of maximum displacement in script and maximum displacement in Comsol is presented in Table 4.2. Compared with Table 4.1 an additional entry is made for 894 DOF which corresponds to 10 pattern divisions.

Table 4.2: Difference between displacements in script and Comsol presented in percent

DOF	$d_{\text{script}}/d_{\text{Comsol}}$
894	100.0002
2964	99.73
10704	99.94

4.3 Convergence Analysis of Kinetic Energy Tolerance

The principle appearance of how the kinetic energy decreases towards static equilibrium in dynamic relaxation is shown in Figure 4.14.

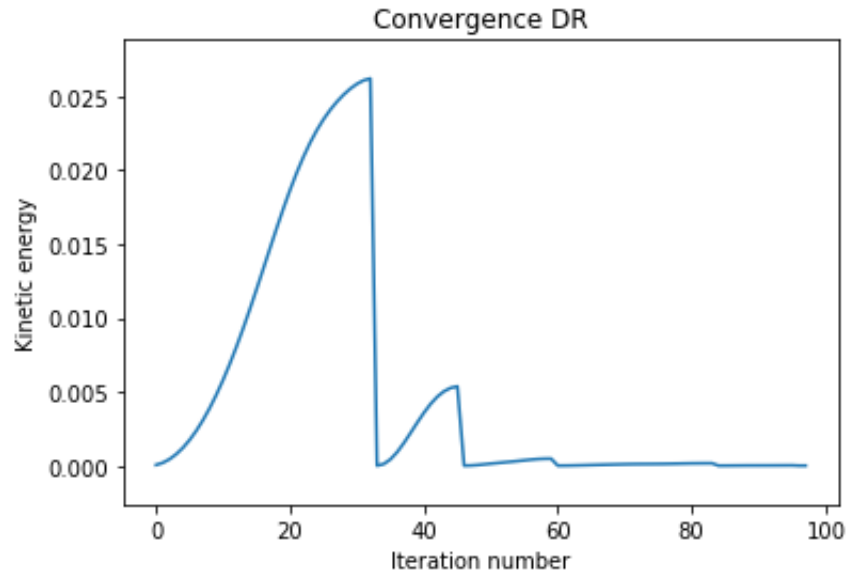


Figure 4.14: Convergence plot for dynamic relaxation

The results in Figures 4.15 and 4.16 show convergence data for the kinetic energy tolerance for a structure with 624 DOF. The kinetic energy tolerance is decreased from 10^{-4} to 10^{-9} and plotted against structure height and membrane action. Figure 4.15 reveals great differences in structure height when the tolerance is high but also a rapid convergence. On the other hand, the level of membrane action in Figure 4.16 show little difference depending on tolerance but a slow convergence. The similarities in appearance are due to the scale of the vertical axis which has been chosen separately for each plot.

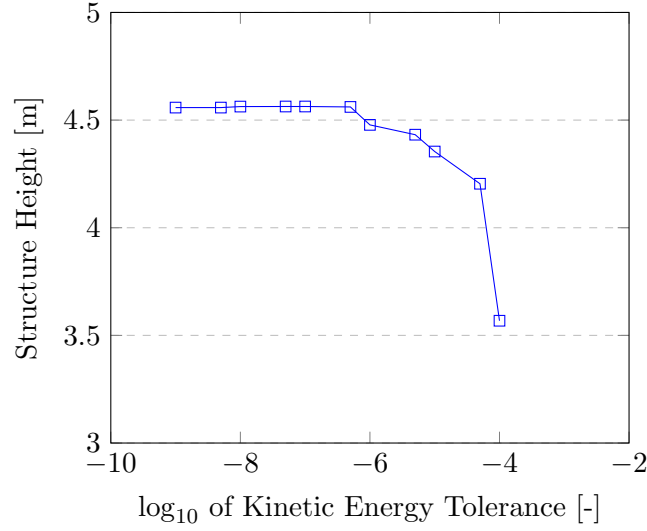


Figure 4.15: Convergence analysis of Kinetic Energy tolerance for 624 DOF model

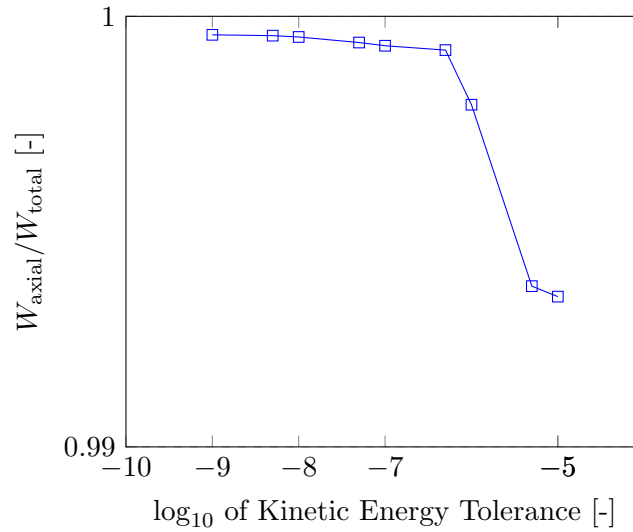


Figure 4.16: Convergence analysis of Kinetic Energy tolerance for 624 DOF model

Figures 4.17 and 4.18 show convergence data for kinetic energy tolerance for a structure with 1212 DOF. The tolerance is plotted against structure height and membrane action. As for the 624 DOF model, the structure height show great differences depending on tolerance and a rapid convergence. However, the level of membrane action for the 1212 DOF model in Figure 4.18 show a greater variation with tolerance compared with the 624 DOF model of Figure 4.16. Again, the axes are adjusted to highlight data in each individual plot.

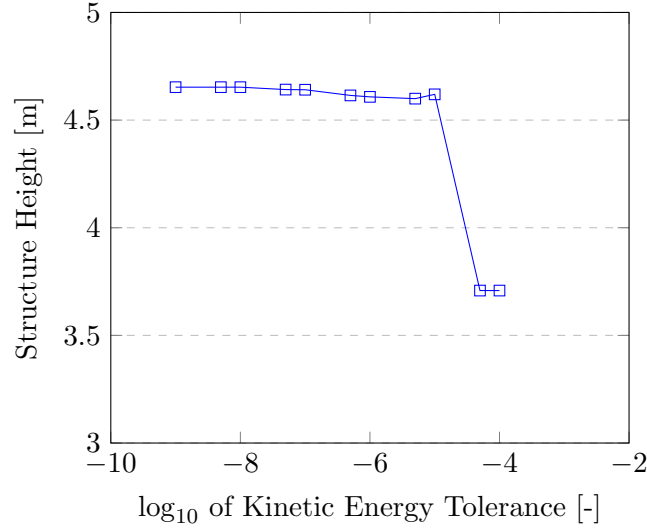


Figure 4.17: Convergence analysis of Kinetic Energy tolerance for 1212 DOF model

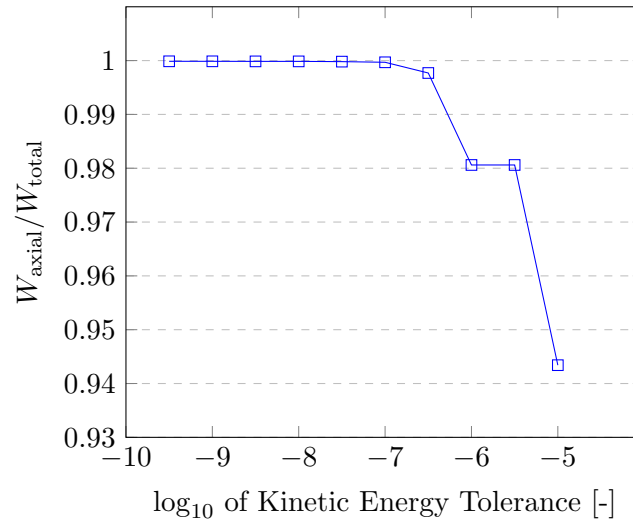


Figure 4.18: Convergence analysis of Kinetic Energy tolerance for 1212 DOF model

The results in Table 4.3 are for a structure with 1212 DOF, with pinned supports and evenly distributed nodal loads. The results show a decrease in membrane action for structures with higher, i.e non converged tolerance, both linearly and non-linearly. The non-linear effect is evident for structures with high kinetic tolerance, whereas the the non-linear behaviour is limited for structures with a low kinetic energy tolerance. This stems from the second order effects, that the axial force in a beam will increase the bending moment. In a converged shape, the bending moment is low as shown by the high level of membrane action. Thus, the second order effect on bending moment will be small and the results similar in a linear and geometrically nonlinear analysis.

Table 4.3: Kinetic energy tolerance effect on shell membrane action

Tolerance	$W_{\text{axial}}/W_{\text{total}}$ from script	$W_{\text{axial}}/W_{\text{total}}$ linear analysis	$W_{\text{axial}}/W_{\text{total}}$ non-linear analysis
10^{-7}	0.9995	0.9995	0.9992
10^{-6}	0.9986	0.9986	0.9976
10^{-5}	0.9399	0.9413	0.8873

For further studies on structures with a higher number of DOF, the level of kinetic energy tolerance is based on the results presented in this section. Ideally, a separate convergence analysis should be performed for the correct DOF. Due to the time consuming process the tolerance 10^{-7} is used for all structures presented in the report.

4.4 Convergence Analysis of Geometric Exportation Tolerance

The result in Table 4.4 show the convergence analysis for the geometric tolerance in Rhino. The absolute tolerance in Rhinoceros 5 is decreased from 10^{-3} to 10^{-5} and evaluated against maximum vertical displacement in Comsol Multiphysics. The results show no difference in vertical displacement for decreasing export tolerance in Rhino.

Table 4.4: Effect of geometric tolerance on the analysis of the form found grid shell

Absolute tolerance	Relative tolerance [%]	Angle tolerance [°]	Vertical displacement [mm]
10^{-3}	1	1	123.47
10^{-4}	0.1	0.1	123.47
10^{-5}	0.01	0.01	123.47

4.5 Grid Shell with Single Boundary

Results for the grid shell with single boundary are presented in this section. First, the optimization data is presented followed by analysis results of the structure with the lowest cost for 20 and 40 pattern divisions.

4.5.1 Optimization Data

Optimization data for 20 and 40 pattern divisions are presented in Figures 4.19 and 4.20. The minimum of the cost function with height penalty yield structures with a height of 5.2 m and 4.86 m, respectively, for further analysis.

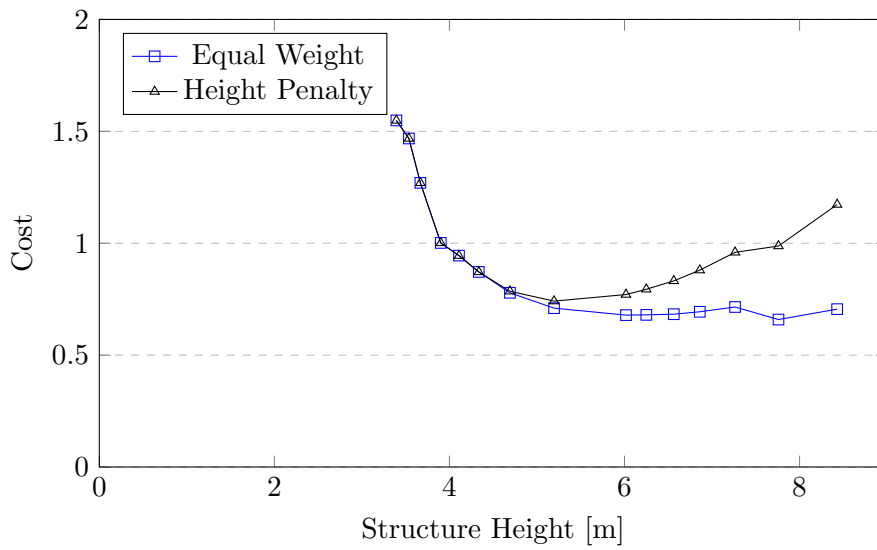


Figure 4.19: Optimization data for 20 pattern divisions and single boundary

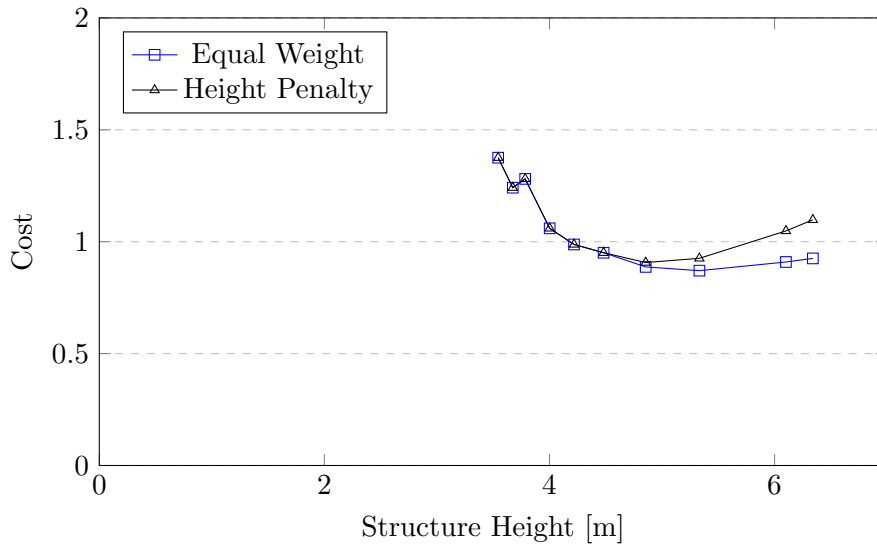


Figure 4.20: Optimization data for 40 pattern divisions and single boundary

4.5.2 Verification with Pattern Division 20

The verification results for all boundary nodes pinned are presented in Table 4.5. Results are given for load case 1: evenly distributed snow load and self weight and load case 2: evenly distributed snow load and self weight with additional snow drift load. In addition, the eigenfrequency, EF for the first mode is presented.

Table 4.5: Results for 20 pattern divisions with all boundary nodes pinned

Load Case	Analysis	d_{\max} [mm]	$\sigma_{\max, \text{vM}}$ [MPa]	$\frac{W_a}{W_{\text{tot}}}$ [-]	$\frac{d^{II}}{d^I}$ [-]	f_1 [Hz]
1	L	16.8	80.8	0.999		
	NL	16.9	80.3	0.999	1.006	
2	L	21.8	90.7	0.992		
	NL	22.6	91.5	0.990	1.037	
EF	L	-	-	-	-	2.80

The first mode shape of the grid shell is presented in Figure 4.21.

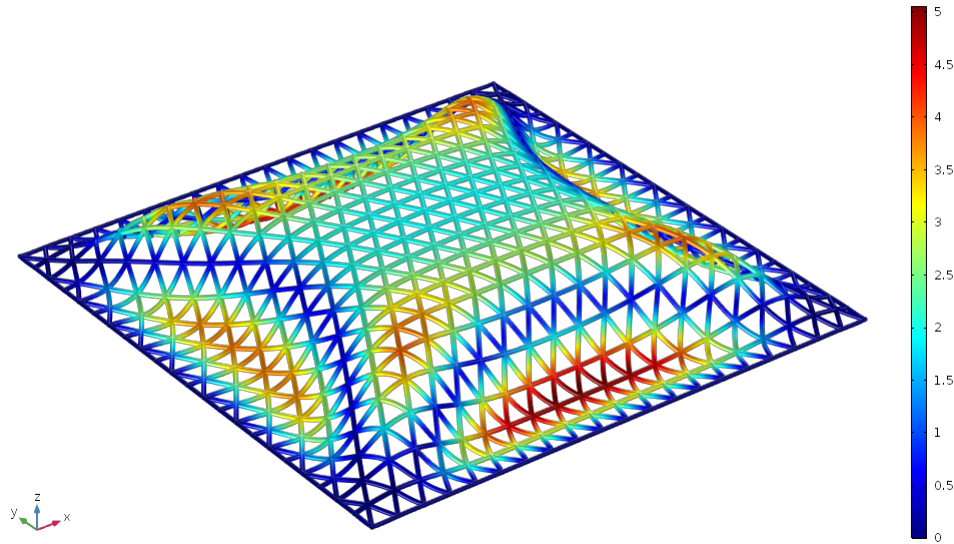


Figure 4.21: First mode of vibration at 2.80 Hz for grid shell with 20 pattern divisions and single boundary

The most adverse load case, Load Case 2, is further analysed with different boundary condition configurations, see Figure 3.22 in section 3.9.2. Tables 4.5 and 4.6 present results for cross section VKR 90x50x5 derived from the member buckling FEM-script in the form finding algorithm.

Table 4.6: Results for 20 pattern divisions with different boundary conditions

BC	Analysis	d_{\max} [mm]	$\sigma_{\max, \text{vM}}$ [MPa]	$\frac{W_a}{W_{\text{tot}}}$ [-]	$\frac{d^{II}}{d^I}$ [-]	f_1 [Hz]	comment
1	L	222	2280	0.744		1.53	
	NL	327	2710	0.678	1.473		Collapse 0.99875P
2	L	54.5	752	0.853		2.2	
	NL	64.8	826	0.823	1.189		
3	L	25.8	264	0.957		2.45	
	NL	26.5	274	0.952	1.027		
4	L	192.5	758	0.862		1.77	
	NL	269.6	823	0.822	1.401		
5	L	161.9	509	0.855		1.78	
	NL	242.8	527	0.776	1.500		

4.5.3 Verification with Pattern Division 40

The verification of the grid shell with 40 pattern divisions and single boundary is only performed for the two load cases and all boundary nodes pinned. In the nonlinear analysis the grid is unable to carry the full design load. The results are summarised in Table 4.7 with a comment on at which load level the structure collapse.

Table 4.7: Results for 40 pattern divisions with all boundary nodes pinned

Load Case	Analysis	d_{\max} [mm]	$\sigma_{\max, \text{vM}}$ [MPa]	$\frac{W_a}{W_{\text{tot}}}$ [-]	$\frac{d^{II}}{d^I}$ [-]	f_1 [Hz]	Comment
1	L	26.7	142	0.997			
	NL	41.7	113	0.915	1.562		Collapse 0.48P
2	L	27.8	112	0.980			
	NL	58.7	130	0.897	2.112		Collapse 0.74P
EF	L	-	-	-	-	1.81	

The shell buckles at a load significantly lower than the form finding load for which the cross section is chosen. Figure 4.22 shows the mode of collapse for asymmetric load with regions of high relative displacements.

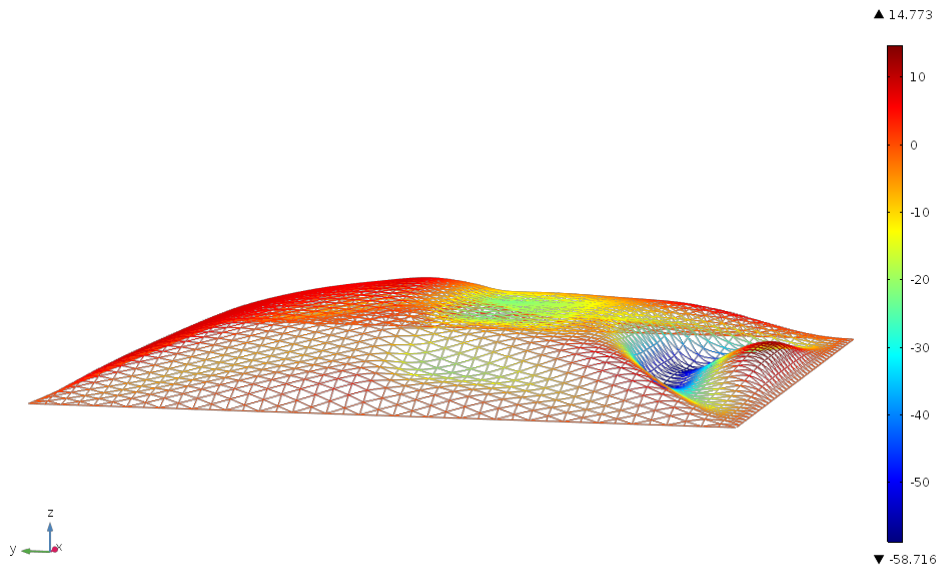


Figure 4.22: Mode of collapse for grid shell under asymmetric load with 40 pattern divisions and single boundary, the deformation is scaled by a factor 45

4.6 Grid Shell with Interior Boundary

Results for the grid shell with interior boundary are presented in this section. First, the optimization data is presented followed by analysis results of the structure with the lowest cost for 20 and 40 pattern divisions.

4.6.1 Optimization Data

Optimization data for 20 and 40 pattern divisions is presented in Figures 4.23 and 4.24. The minimum of the cost function with height penalty yield structures with a height of 3.16 m and 2.9 m, respectively, for further analysis.

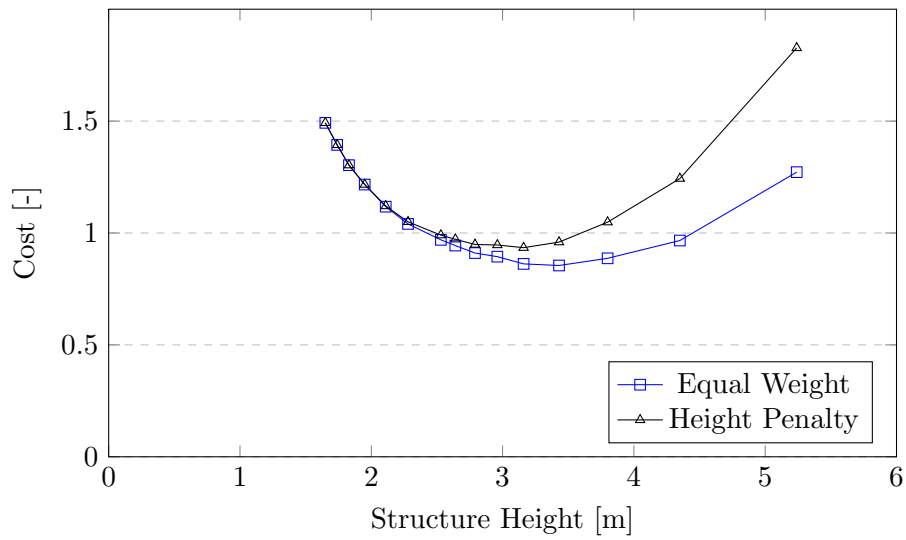


Figure 4.23: Optimization data for 20 pattern divisions with interior boundary

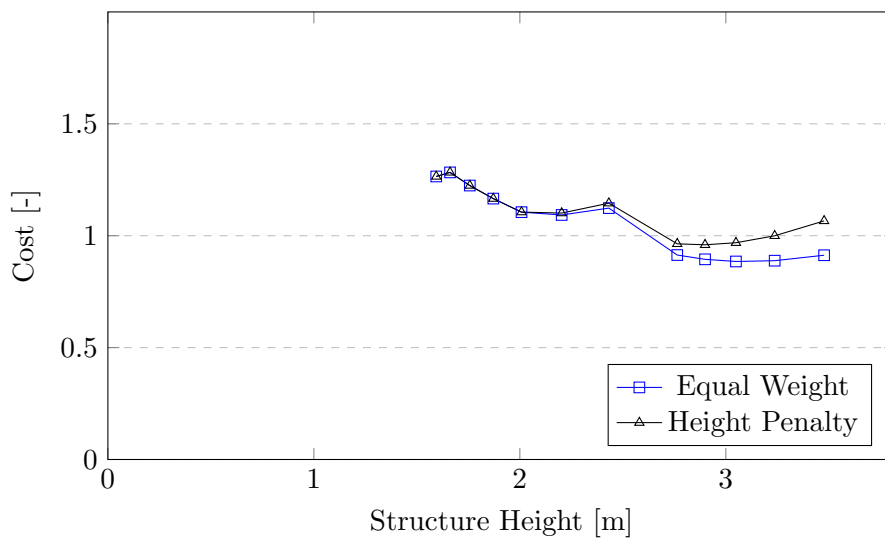


Figure 4.24: Optimization data for 40 pattern divisions with interior boundary

4.6.2 Verification with Pattern Division 20

For the shell with all boundary nodes pinned, Table 4.8 presents results for load case 1: evenly distributed snow load and self weight and load case 2: evenly distributed snow load and self weight with additional snow drift load. Further, the eigenfrequency, EF for the first mode is presented.

Table 4.8: Results for 20 pattern divisions with all boundary nodes pinned

Load Case	Analysis	d_{\max}	$\sigma_{\max, \text{vM}}$	$\frac{W_a}{W_{\text{tot}}}$	$\frac{d^{II}}{dI}$	f_1
		[mm]	[MPa]	[-]	[-]	[Hz]
1	L	4.59	72.5	0.9976		
	NL	4.64	72.6	0.9975	1.010	
2	L	32.80	201	0.9154		
	NL	35.60	211	0.9022	1.086	
EF	L	-	-	-	-	2.75

As for the grid shell with single boundary, the most adverse load case is Load Case 2, which is further analysed for different boundary condition configurations, see Figure 3.22 in section 3.9.2. The results presented in Table 4.9 are for cross section VKR 60x60x5 derived from the member buckling FEM-script in the form finding algorithm.

The first mode shape of the grid shell is presented in Figure 4.25.

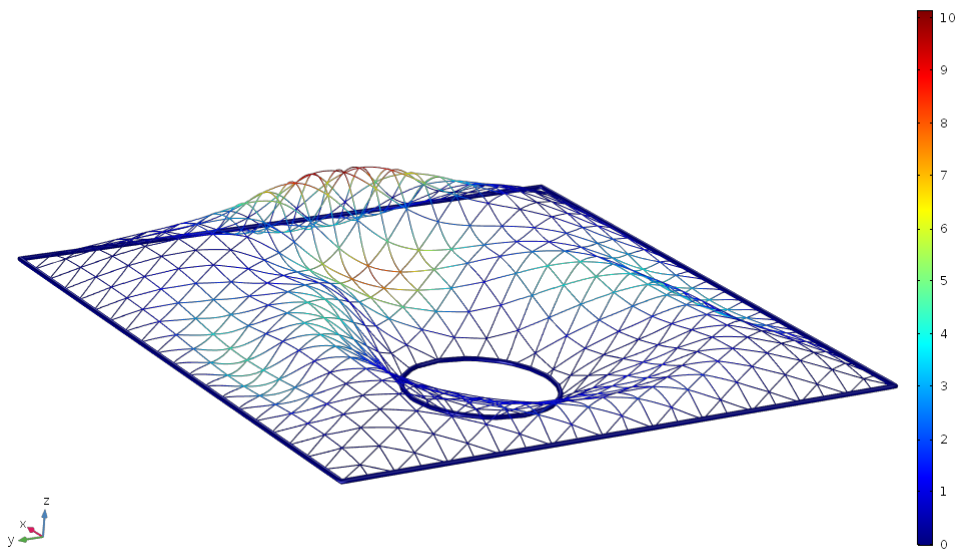


Figure 4.25: First mode of vibration at 2.75 Hz for grid shell with 20 pattern divisions and interior boundary

Table 4.9: Results for 20 pattern divisions with different boundary conditions

BC	Analysis	d_{\max} [mm]	$\sigma_{\max, \text{vM}}$ [MPa]	$\frac{W_a}{W_{\text{tot}}}$ [-]	$\frac{d^{II}}{dI}$ [-]	f_1 [Hz]
1	L	98.9	915	0.852		1.86
	NL	119.7	958	0.833	1.211	-
2	L	37.86	513	0.894		2.43
	NL	43.59	522	0.874	1.151	-
3	L	35.30	240	0.897		2.45
	NL	40.77	247	0.882	1.155	-
4	L	77.80	286	0.910		1.94
	NL	92.43	291	0.893	1.188	-
5	L	61.23	250	0.923		2.02
	NL	71.30	275	0.907	1.164	-

4.6.3 Verification with Pattern Division 40

For 40 pattern divisions and all boundary nodes pinned the results are presented in Table 4.10. As for the previous cases, results are presented for load case 1: evenly distributed snow load and self weight, load case 2: evenly distributed snow load and self weight with additional snow drift load and eigenfrequency, EF of the first mode.

Table 4.10: Results for 40 pattern divisions with all boundary nodes pinned

Load Case	Analysis	d_{\max} [mm]	$\sigma_{\max, \text{vM}}$ [MPa]	$\frac{W_a}{W_{\text{tot}}}$ [-]	$\frac{d^{II}}{dI}$ [-]	f_1 [Hz]	comment
1	L	7.08	123	0.996		-	-
	NL	7.29	121	0.997	1.0297	-	-
2	L	53.8	299	0.935		-	
	NL	130	426	0.739	2.694	-	Collapse at 0.90P
EF	L	-	-	-	-	2.75	-

Load Case 2 is further analysed with the different boundary condition configurations according to Figure 3.22 in section 3.9.2. The results in Table 4.11 are for cross section VKR 40x40x3 derived from the member buckling FEM-script in the form finding algorithm. The empty rows of non linear analysis stems from the fact that the structure collapses for the two boundary conditions which exhibit the lowest stress in the linear analysis. Since it is unlikely that the remaining cases will carry the load without collapse, the results from nonlinear analysis of the remaining cases are omitted.

Table 4.11: Results for 40 pattern divisions with different boundary conditions

BC	Analysis	d_{\max} [mm]	$\sigma_{\max, \text{vM}}$ [MPa]	$\frac{W_a}{W_{\text{tot}}}$ [-]	$\frac{d^{II}}{dI}$ [-]	f_1 [Hz]	comment
1	L	173	1073	0.807		1.37	-
	NL	-	-	-	-	-	-
2	L	66.4	626	0.875		1.80	-
	NL	-	-	-	-	-	-
3	L	61.7	388	0.921		1.81	-
	NL	143	469	0.754	2.760	-	Collapse at 0.84P
4	L	156	587	0.831		1.40	-
	NL	-	-	-	-	-	-
5	L	140	445	0.842		1.43	-
	NL	304	727	0.664	2.716	-	Collapse at 0.80P

Boundary condition indices 3 and 5 in Table 4.12 exhibit the lowest von Mises stress levels and are reanalysed for cross section VKR 50x50x5. This cross section is not a result from the script but chosen manually to study the difference obtained with a slightly larger cross section. The results are presented in Table 4.12.

Table 4.12: Results for 40 pattern divisions with boundary conditions index 3 and 5 for cross section VKR50x50x5

BC	Analysis	d_{\max} [mm]	$\sigma_{\max, \text{vM}}$ [MPa]	$\frac{W_a}{W_{\text{tot}}}$ [-]	$\frac{d^{II}}{dI}$ [-]	f_1 [Hz]
3	L	29.6	233	0.914		2.41
	NL	36.1	242	0.888	1.039	
5	L	73.7	263	0.859		2.01
	NL	90.9	261	0.831	1.234	

4.7 Global Stability and Imperfections

The stability of single boundary results from the form finding script are studied by linear buckling analysis and by introduction of the first buckling mode as an imperfection. The first buckling mode for 20 pattern division is shown in Figure 4.26, the first mode for the other pattern divisions are nearly identical and therefore left out of the presentation. Figure 4.27 shows how the load multiplication factor, λ varies with the number of pattern divisions. The load applied is the uniform form finding load for which the cross section is chosen. A critical load factor greater than 1 thus means that the critical load for global buckling is greater than the applied load and conversely if it is lower than 1, the critical load is lower than the applied load.

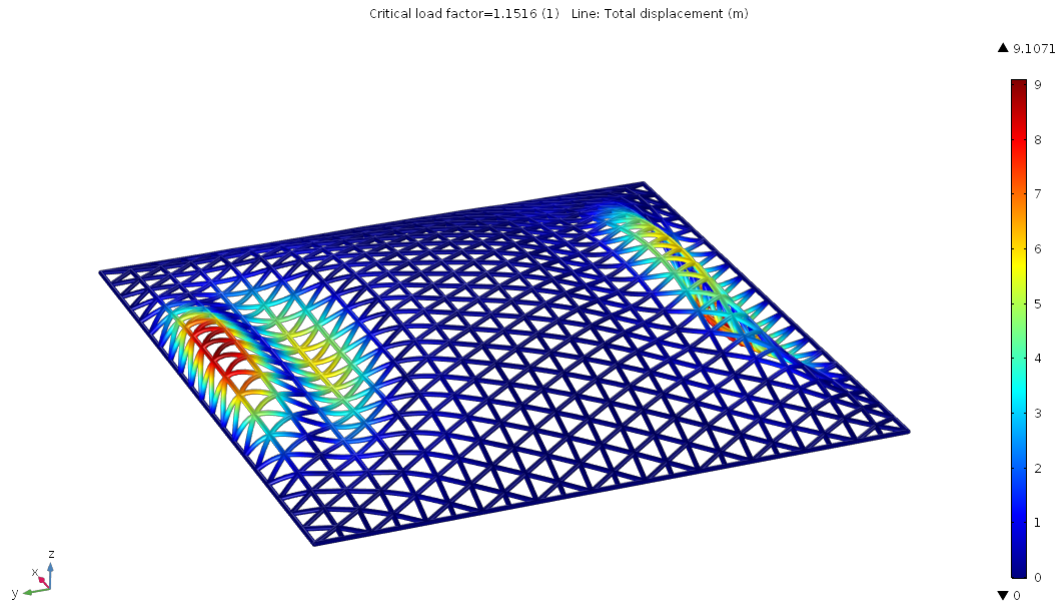


Figure 4.26: First buckling mode for 20 pattern divisions

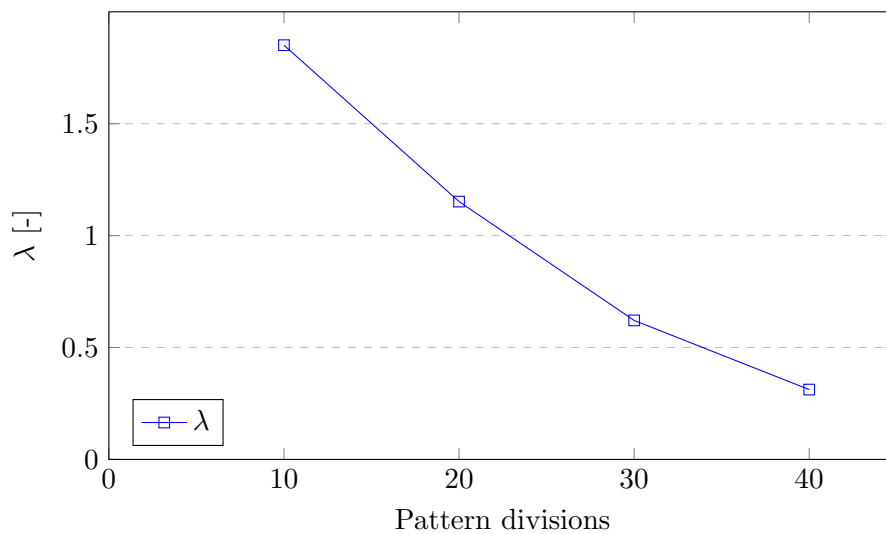


Figure 4.27: Critical load factor against number of pattern divisions

These results are presented again in Table 4.13 along with a comparison between analysis results for perfect geometry, labelled P, and for geometry with imperfections with a maximum amplitude of 10 mm, labelled 10mm IP. The section column refers to the cross section of the members, here circular pipes where the first number is the outer diameter and the second number is the thickness, both in mm. Where the critical load factor, λ is lower than 1 the grid shell collapses before the applied load is reached. The comment gives the load level where the solution crashes indicating collapse of the structure.

The difference in how displacements vary across the shell between perfect and imperfect geometry is shown in Figures 4.28 and 4.29. Figure 4.28 show an evenly distributed displacement field with a maximum at midspan for the perfect geometry. The displacements in Figure 4.29 for the geometry with imperfections show significant similarities to the buckling mode in Figure 4.26. The maximum displacement is off center at the largest downwards displacement of the buckling mode and upwards deflection of the same order of magnitude occurs at the leftmost boundary in Figure 4.29.

Table 4.13: Data of linear buckling and nonlinear analysis with and without imperfections

Divisions	h [m]	Section	$\frac{N_{Ed}}{N_{b,Rd}}$	λ	Geometry	d_{max} [mm]	σ_{max} [MPa]	Comment
10	4.97	139.7x4	0.730	1.851	P 10mm IP	33.4 33.5	150 150	
20	5	88.9x4	0.487	1.152	P 10mm IP	28.4 39.8	130 166	
30	5.13	60.3x4	0.525	0.621	P 10mm IP	45.3 54.6	147 160	Collapse 0.74P Collapse 0.48P
40	4.9	42.4x4	0.650	0.312	P 10mm IP	41 39.1	102 93.2	Collapse 0.34P Collapse 0.22P
40	4.9	88.9x4	0.208	2.07	P 10mm IP			Section chosen manually

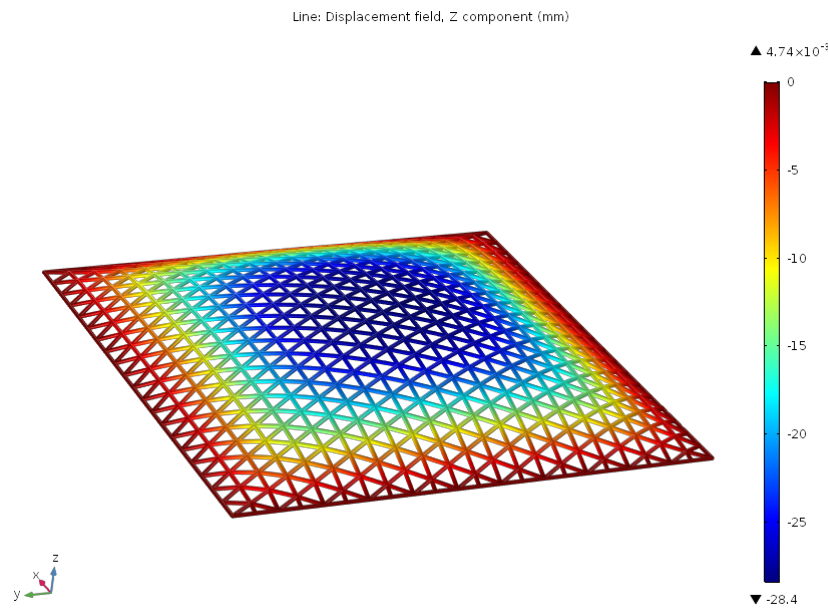


Figure 4.28: Vertical displacements in mm for a perfect geometry with 20 pattern divisions

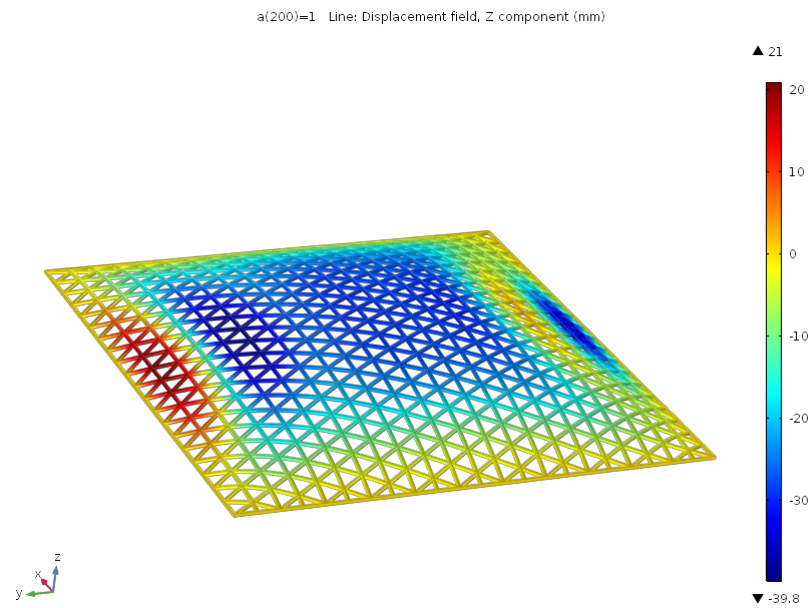


Figure 4.29: Vertical displacements in mm from a nonlinear analysis for an imperfect geometry with 20 pattern divisions

4.8 Prescriptive Dynamic Relaxation

This section presents some results by prescriptive DR. First a 2D arch is form found for uniformly distributed load with and without prescriptive lengths. Then results for different boundary conditions are presented for a grid shell supported on a single boundary.

4.8.1 Continuous Arch

The form found arch is a continuous arch with pin supports at both ends with a span of 40 m. The arch is presented in Figure 4.30 with numbering of the elements.

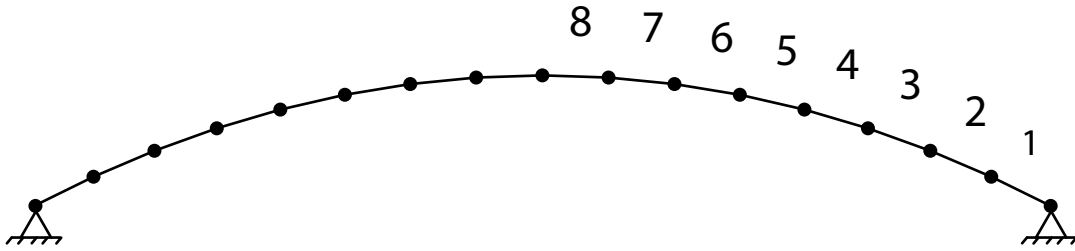


Figure 4.30: Arch form found by prescriptive dynamic relaxation

The results for the arch using DR and PDR is presented in Table 4.14. The prescribed lengths are set to the mean of the relaxed lengths in the results from DR as outlined in section 3.6. The length tolerance for convergence is set to 1 mm. Notable is the higher degree of membrane action achieved by PDR.

Table 4.14: Comparison between DR and PDR

Method	h [m]	$h \frac{W_a}{W_{tot}}$ [-]	L_{max} [m]	L_{min} [m]	L_p
DR	5.1381	0.9104	2.6130	2.6012	-
PDR	5.1361	0.9984	2.6069	2.6056	2.606

The lengths of each arch segment is presented in Table 4.15, the arch is symmetric about the midpoint and hence only data for one side is presented.

Table 4.15: Lengths of element in arch for DR and PDR

Element	Length in DR [m]	Length in PDR [m]
1	2.6130	2.6061
2	2.6107	2.6057
3	2.6073	2.6066
4	2.6074	2.6069
5	2.6022	2.6068
6	2.6047	2.6065
7	2.6012	2.6056
8	2.6016	2.6060

4.8.2 Grid Shell on Single Boundary

Prescriptive lengths are applied on a grid shell with single boundary and 20 pattern divisions. Three different boundary conditions are tested: all boundary nodes pinned, corners pinned and other rollers, corner pinned and others free. The results for these cases are presented in Table 4.16 and compared with the results without prescribed lengths. In the count of unique lengths, members within a difference of 1 mm are treated as equal. The tolerance in length difference is set with trial and error for each case since a lower tolerance could not be met by the algorithm.

The shape without prescriptive lengths is presented in Figure 4.31. The shapes obtained by prescriptive lengths are presented in Figures 4.32 - 4.34. No analysis results are presented since the shapes clearly deviate from the smooth surface expected from an effective membrane active structure.

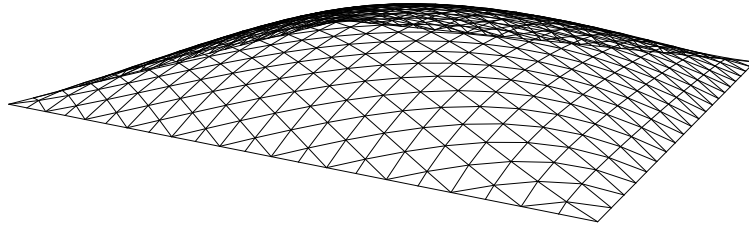


Figure 4.31: Grid shell without prescriptive lengths

With all boundary nodes pinned, the shape in Figure 4.32 is obtained. The algorithm could converge with a tolerance of 55 mm, far from the desired tolerance of 1 mm. Still, the shape shows considerable folding and creasing along the boundary.

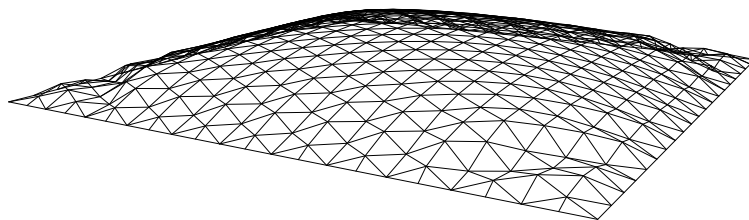


Figure 4.32: Grid shell with prescribed lengths and pinned boundary nodes

The shape of Figure 4.33 is obtained by applying rollers to the boundary nodes along the sides. The sides with members parallel to the boundary fold to a vertical position while the two other sides show creasing but maintain a cupola like shape globally.

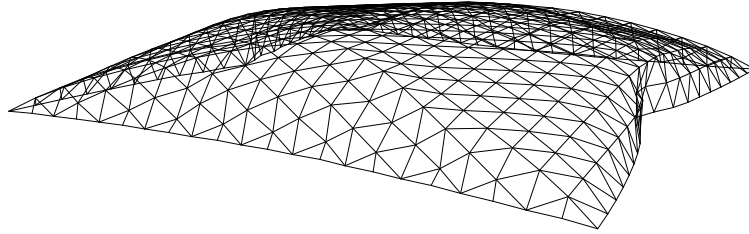


Figure 4.33: Grid shell with prescribed lengths and roller along sides of boundary

Finally, the boundary nodes along the sides are treated as free producing the shape in Figure 4.34. The shape exhibits less creasing than the other two boundary conditions tested but significant folds run from the corners towards the inner parts of the grid shell.

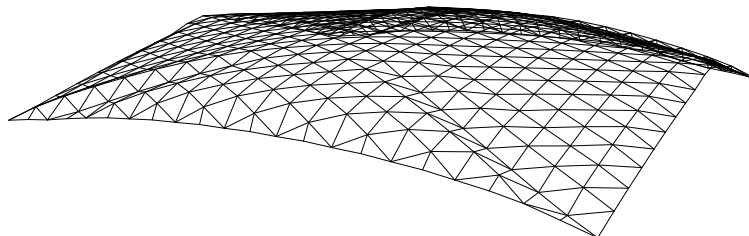


Figure 4.34: Grid shell with prescribed lengths and free nodes along sides of boundary

Table 4.16: Comparison of boundary conditions in PDR

Boundary conditions	l_{tol} [mm]	Unique lengths
Pinned	55	104
Roller	30	98
Free	16.5	42
Without prescription	-	186

Chapter 5

Discussion

As described in section 3.1, a form found grid shell is generated and analysed in four major steps: geometry, form finding, optimization and verification. The capabilities of geometric operations in Grasshopper were used to generate the initial geometry in a parametric model. In the form finding, the limited possibilities of performing mathematical operations in Grasshopper forced the authors to build the form finding algorithm in an external Python script which uses the mathematical plug-in Numpy. The process of transferring data from Grasshopper to the external script and back to Grasshopper is a somewhat tedious process and requires a profound insight in the scripted algorithm. A mathematical operating programming language in Grasshopper would make the form finding process more user friendly and accessible for external users. However, a scripted form finding algorithm forces the designer to have full control over parameters and allowed optimization data to be extracted from the script. This data laid the foundation of which structures where further analysed.

5.1 Form and Geometry

Geometrically, the form finding algorithm has shown to be able to generate a broad variety of geometric shapes. Linearly, an arbitrary boundary geometry results in a shell structure with a high degree of membrane action. The shapes presented in section 4.1.1 shows the strength of the algorithm. The power of an FEM solver inside the form finding algorithm gives the designer the possibility to evaluate the generated forms based on structural parameters. This enables rational design decisions at an early stage in the design process and more specifically in this study, the generation of optimization data used to find an optimal shell structure. Since the equilateral triangle pattern is initially generated for square boundary geometries, any deviation from a square boundary geometry requires the pattern to be trimmed. This enables the designer to adjust the boundary geometry without generating a new grid pattern. However, the trim operation give rise to geometric distortion in the grid pattern along the boundary line. The geometric distortions are not handled in this report since the global geometry is likely unaffected to a high extent. In addition, a uniform grid is difficult or even impossible to generate for any boundary geometry. However, in later stages of design, the authors suggest three possible solutions to handle the problem of geometric distortion.

Firstly, a rearrangement of nodes is suggested. Adjacent nodes which lie within a defined distance from each other could be merged into a single node. Considerations are needed to ensure all parts of the grid are supported along the boundaries where this process might result in nodes unconnected to the boundary.

Secondly, a reformulation of the grid generation algorithm is suggested. The studied grid formulation is based on a division of a square boundary geometry, giving an approximation of an equilateral triangle pattern. To overcome geometric distortions the authors suggest that the grid formulation could be tailored to fit the geometric shape of the boundary geometry. This method requires case by case adaption since an algorithm to generate a well suited pattern for any arbitrary boundary geometry is very difficult to produce, if it is at all possible.

Finally, a pattern approximation of a form found surface is suggested. As described in section 2.5.1, the designers of the British Museum Roof used the method of dynamic relaxation to approximate a predefined surface geometry with a grid pattern. The authors suggest an extension of this method by using dynamic relaxation to find the surface geometry. When the surface geometry is found, the same method can be used to find the best fitted grid pattern configuration.

The power of the scripted form founding algorithm lies in the capacity to form find grid shells with a great variety of boundary geometries.

5.2 Convergence Analysis

Dynamic relaxation works by a dynamic equilibrium formulation which gradually approaches static equilibrium as the nodal coordinates are updated. This gradual approach to static equilibrium makes a judgement of convergence necessary. As shown in Figures 4.15 - Figure 4.18 both structural height and the degree of membrane action is highly dependant on the convergence of the method. On the other hand, the number of iterations needed to reach a certain tolerance of kinetic energy becomes an important factor in cases of many degrees of freedom since the time for each iteration rapidly increases. Based on this consideration, convergence plots and on the results in Table 4.3, a kinetic energy tolerance of 10^{-7} is deemed sufficient for convergence. The convergence study performed here treats models with 624 and 1212 DOFs with similar results and the same tolerance on kinetic energy is used throughout the study.

The verification results gives a strong indication that the convergence criterion used is sufficient also for structures with a higher number of degrees of freedom. The structures with 20 pattern divisions presented in Tables 4.5 and 4.8 show a high degree of membrane action and low influence of non-linearity for the uniformly distributed load. In a similar manner, the structures with 40 pattern divisions display a high degree of membrane action linearly. The drop in membrane action of the shell with 40 divisions and single boundary in the nonlinear analysis is due to the shell being under-dimensioned with regards to global buckling. Based on these results, the convergence study is deemed valid also for the structures in the case studies.

It is difficult to predict what tolerance is necessary to reach a converged solution. Further, an insufficient convergence does not immediately manifest itself as an obviously erroneous structure by inspection. The need for convergence analysis on a case by case basis is

therefore evident to achieve a structure with a high level of membrane action.

In contrast to the tolerance of convergence in DR, the tolerance of geometric operations in Rhino and on the export to AutoCad drawing does not affect the solution found by the algorithm. This result is shown in Table 4.4 and indicates that the geometric handling of the used method is robust.

However, the FEM script verification presented in Table 4.1 yields a minor difference between the scripted FEM-solver and the commercial software Comsol. This difference may be explained by numerical rounding of the loads used in Comsol. The geometric export tolerance convergence presented in Table 4.4 shows that the absolute tolerance in Rhino does not affect the result in Comsol. However, data transfer from the FEM-script to Comsol via Rhino and AutoCad may result in numerical rounding and geometric differences. Worth noting is that the shape function used by Comsol is ambiguous. Even with Euler-Bernoulli beam formulation chosen, the software displays that a Timoshenko shape function is used, which may have some effects on the results.

5.3 Case Studies

All structures in the case studies are determined by choosing the structure with the lowest cost according to the height penalty curve. This curve generates more distinct minima compared to the data with equal weights. The optimization data presented in sections 4.5.1 and 4.6.1 show that all data exhibit parabolic shapes globally. Most cases also exhibit local minima in addition to the global which would render the use of an optimization algorithm more difficult. One should also keep in mind that it is possible to steer the optimization by the choice of weights and by adding penalties to one or more costs.

In common for all cases, they produce the most advantageous results for the form finding case of all boundary nodes pinned and uniformly distributed load. That their effectiveness drops for any deviation from this situation is not surprising since they are optimized for these conditions.

For both cases, 20 pattern divisions are more successful. The maximum displacement and maximum stress are lower for the more sparse grid. However, these results give no indication of whether 20 or 40 pattern divisions is to be preferred. The difference stems from the inability to treat global buckling adequately for 40 divisions and no conclusions can be drawn since the structures are not comparable. Further, the results of Table 4.12 where the grid shell with interior boundary and 40 divisions is analysed with a larger cross section. This analysis yields results with maximum stress well below the yield stress. The second order cost drops from about 2.7 with the smaller section to between 1 and 1.2 suggesting that the structure with 40 divisions would work well given the right cross section. Also worth noting is the results of Table 4.7 where the grid shell with single boundary and 40 divisions is analysed with all boundary nodes pinned. Linearly the results look promising while the structure collapses in the non linear analysis. This shows the importance of non linear analysis of form found structures.

When different boundary conditions are tested, it is clear that high local stresses occur for many of the boundary condition cases. Especially, the case with only corner points pinned give rise to stresses far above the yield stress of 355 MPa. The peak stresses generally decrease as more than the four corner nodes are supported. The grid shell with interior

boundary is less prone to produce high stress concentrations when compared to the grid shell on single boundary. The differences between the two case geometries are: a shorter span in the case of an interior boundary and that the shell with interior boundary connects to the ring beam at a steeper angle. A solution to the issue of stress concentrations in elements at support nodes could be to reinforce the corners of the shell with external trusses. This would produce a similar effect of BC 5 and provide greater stiffness at the corners without prescribing infinite stiffness at some nodes. Another possibility is to experiment with a decreased stiffness of some or all elements around the boundary of the shell in the form finding to generate a steeper angle to the ring beam.

In summary, boundary conditions have a major influence on the structural behaviour of the shells. In later design stages the ring beam and modelling of boundary conditions needs to be thoroughly studied.

5.4 Global Stability

The studies of global stability reveal an important shortcoming of the applied form finding script. The critical load factor presented in Table 4.13 is low for all pattern divisions. For 30 and 40 divisions it is lower than 1 and the structure buckles globally before the form finding load is reached. That no structure reaches yield stress in the analysis shows that the structural behaviour is governed by instability. For 10 divisions the critical load factor is roughly 2 and the results of Table 4.13 barely shows any influence of the imperfection on the non linear analysis.

The results in Figure 4.27 show how the critical load factor decreases with an increase in pattern divisions. As the number of divisions increases, the elements become shorter and the effect of column buckling no longer governs the design with respect to stability. This issue could be overcome by a scripted linear buckling solver in the form finding algorithm. This would introduce another constraint in the optimization of equation 3.46 and implicitly on the solution space of the parametric model as outlined in section 3.2. Of course this requires additional script development but, based on the presented results, this development is necessary if the aim is to produce only structurally sound solutions.

By comparing the geometrically nonlinear results in Table 4.13 it is evident that the nonlinear effect increases as the critical load factor goes down. This is in accordance with common knowledge of instability phenomena, that the nonlinearity increases as the load increases towards the critical load. The structures can, with initially perfect geometry, sustain somewhat higher loads than the critical, for example 74 percent compared to a λ of 0.621, corresponding to 62 percent, for 30 divisions. This means that some post-buckling resistance is available in the idealised case. However, with the introduction of imperfections, the structure collapses before the critical load is reached. Again, this corresponds well with fundamental knowledge of structural stability.

Further, the imperfection introduced in this study is only an example. In an optimized structure working in compression, analysis of stability is crucial. In a design situation, additional imperfections needs consideration. Examples are: imperfections of higher buckling modes and possibly combinations of different modes, buckling under asymmetric loading and the effects of support displacements. These effects could be treated by the same method as applied here.

5.5 Prescriptive Dynamic Relaxation

The idea of minimising the number of unique members in a grid shell is tempting and would provide a tool to enable cheaper construction of complex geometries. The approach tested in this study to achieve this is prescriptive dynamic relaxation, a method based purely on mechanics.

An arch of span 40 meters is form found with prescriptive lengths with a tolerance of 1 mm. Even though this results may look promising, the algorithm was just able to converge at this tolerance limit. But it did converge which prompted testing on a grid shell in section 4.8.2.

The grid shell was a greater challenge due to increased geometric complexity. The three boundary conditions tested all produced results which deviated clearly from the form found shape without prescriptive lengths. A comparison between this geometry in Figure 4.31 and the three cases in Figures 4.32 - 4.34 clearly shows how the prescriptive lengths caused wrinkles and folds in the surface of the shell. In all cases the wrinkles appear along the boundary of the grid shell.

The comparison between the results presented in Table 4.16 clearly shows how both the difference in length to the prescribed length and the number of unique lengths decreases when the freedom of movement increases along the boundary. This is in line with the results presented by Bagrianski and Halpern [23] where boundary nodes are treated as rollers and support nodes are modelled as rollers with prescribed forces to simulate reaction forces. This allows the boundary nodes to move but introduces several issues. Firstly, the need to allow support movements in order to reach convergence may lead to a form finding process which finds a shape for a boundary geometry which is too dissimilar from the specification of a certain project. Secondly, a new range of parameter combinations are available if the boundary conditions as well as reactions become variables to adjust. Finally, the issue of how to set the prescriptive lengths of a complicated geometry remains unresolved. A similar approach to the one taken here would be to group the element according to their strained lengths instead of their planar lengths. These issues taken together, makes prescriptive dynamic relaxation of grid shells a difficult task.

An alternate approach to prescriptive lengths is treating the problem from the point of architectural geometry. This lies beyond the scope of this study and would require great control of the geometry as well as rigorous understanding of the structural mechanics in order to adjust geometry without jeopardising structural performance.

5.6 Parametric Approach

A parametric approach vastly increases flexibility in early design phases since alternatives and changes may be visualised and tested at the rate of computational convergence. Generally, a parametric model where the designer can control a group of parameters simultaneously produces a high number of combinations. To reduce the number of combinations to a series of viable options, the parameters may be coupled to an evaluation of one or more criteria. This is especially true for form found structures, where even the aesthetic expression is dependent on structural mechanics.

Thus, with the objective to find structurally efficient forms, the authors suggest a parametric model with structural constraints. The purpose, in this study, is to generate a series of alternative structures which all fulfil the predefined constraints. The series of structures with different combinations of output gives the designer the possibility to rule out ill-suited design proposals and to find the best suited structure for the context studied. A simple, manual cost function optimization is employed. Of course more sophisticated optimization algorithms may be employed.

Given the rather demanding calculations needed to perform dynamic relaxation a more sparse grid pattern, perhaps in combination with a higher convergence tolerance, can be used to visualise alternatives. Meanwhile, from the structural point of view these changes do not effectively represent the structural behaviour. The tolerance for convergence in dynamic relaxation must be kept strict to ensure a form utilising membrane action is found. Further, a more sparse grid pattern to reduce the number of degrees of freedom alters the principal behaviour of the grid shell, as discussed earlier.

To explore design alternatives a parametric geometry is highly useful. However, as the geometrical handling grows more general so does the need to handle exceptions from the rules by which the geometry is generated. As long as the planar grid can be unambiguously represented the method proposed here will provide a global geometry which is structurally correct in the sense that it will carry the design load by membrane action. But an exception occurs as the grid grows denser, the method does not handle global instability in a satisfactory way. The rather large parametric variation changes the phenomena which govern the structural design. In this case the governing structural response changes qualitatively from member buckling to global instability. Further, the solutions become increasingly approximate as the boundary geometries deviate from the square for which the initial grid is drawn.

The limitations of performing mathematical operations in Grasshopper limited the possibilities of developing a closed loop process. In such a loop, the grid typology and global geometry could be reformulated through feedback from structural evaluation. If and when this becomes available there will be great possibilities to update geometry and grid pattern based on FEM results.

In summary, a parametric approach to form find grid shells which are structurally sound is within reach. A wide variety of boundary geometries can be handled. The form finding algorithm can be coupled to an FEM evaluation of the form found geometry. To mend the issue of inappropriate cross section selection of the members in some grids, a global buckling analysis should be included in the cross section selection. Finally, to work with a platform used by architects lays the foundation of successful collaboration between architects and structural engineers.

Chapter 6

Conclusions

The results presented show that a structurally meaningful geometry can be generated by a parametric modelling approach where constraints limit the solution space, even if multiple parameters are included.

Structural stability must be addressed with care and precision when dealing with optimized structures which primarily work by compression. Specifically to the method employed here, the solution space of the model must be limited by a linear global buckling solver to ensure that only structurally sound solutions are produced.

One advantage of the proposed method which lies beyond structural mechanics is the fact that the method utilises Rhino and Grasshopper which are common tools within architecture. A direct link between these interfaces and computational capacity enables the structural engineer to contribute to the design from early stages until final design.

Structural engineering is especially crucial in design of form found structures. These are structures where the shape and the performance go hand in hand and, as shown by the precedents, engineering pushes the boundaries of form and expression.

6.1 Further Studies

In relation to grid shells there are many interesting prospects for future research. A few will be highlighted here. The parametric approach will be touched upon with grid shells in mind. The possibilities of parametric modelling in general is vast and will likely attract considerable attention but are deliberately left out of this section.

The analysis results reveal stress distributions where, in many cases, a large part of the grid is under low stress. This shows the possibility of grid optimization based on structural evaluation. The grid could be optimized by manipulating the topology of the grid, however this may produce highly irregular facets and an impractical solution. One possibility to mitigate this issue would be to manipulate the density of the grid rather than the topology. Self similar fractal patterns could be employed to vary the density of the pattern in an optimal way. Using fractal patterns, the different facets would only be scaled versions of a set of facets.

Related to the issue of grid pattern optimization is the relation between grid pattern and the form finding process. Strictly, a change in pattern requires the form finding to be carried out again in order to form find the global shape for this pattern. A procedure of grid pattern optimization then becomes an iteration dependent on the heavy computational demands of a form finding procedure. A question is then, if, and under what circumstances, an optimization of the grid may be undertaken without form finding the shape in each iteration.

An inverted form of these problems is also interesting, to study what pattern is optimal given that the shape is determined. This is for example the case if the task is to produce a structurally efficient approximation of a free form shell.

Another interesting topic is related to practical concerns such as fabrication and construction of a grid shell. The capacity to model and digitally represent complex geometries have never been greater. However, to turn the digital representation to a physical object can cause difficult problems. The parametric solution space could be limited not only by structural mechanics but also by practical concerns. For instance, facet planarity, number of unique elements and maximum fabrication tolerances could be included as constraints or as parameters in the model.

Studies of the dynamic behaviour of grid shells are scarce and this topic deserves further attention. Specific to grid shells are questions of how to estimate the physical damping values with regards to the many connections between elements and different materials. With a broader outlook, the dynamics of wind in interaction with structural movements is a difficult problem which is further complicated by advanced geometry.

Bibliography

- [1] Sture Samuelsson. *Ingenjörrens Konst-Material och Konstruktion ger form*. Stockholm, Sweden: Balkong Förlag, 2015.
- [2] Rivka Oxman and Robert Oxman. “THE NEW STRUCTURALISM DESIGN, ENGINEERING AND ARCHITECTURAL TECHNOLOGIES”. In: *Architectural Design* 80.4 (2010), pp. 15–23. DOI: [10.1002/ad.1101](https://doi.org/10.1002/ad.1101).
- [3] Michael R. Barnes. “Form-Finding and Analysis of Prestressed Nets and Membranes”. In: *Computers & Structures* 30.3 (1988), pp. 685–695. DOI: [https://doi.org/10.1016/0045-7949\(88\)90304-5](https://doi.org/10.1016/0045-7949(88)90304-5).
- [4] J. S. Brew and D. M. Brotton. “Non-linear structural analysis by dynamic relaxation”. In: *International Journal for Numerical Methods in Engineering* 3.4 (1971), pp. 463–483. DOI: [10.1002/nme.1620030403](https://doi.org/10.1002/nme.1620030403).
- [5] Sigrid Adriaenssens et al. “Finding the Form of an Irregular Meshed Steel and Glass Shell Based on Construction Constraints”. In: *JOURNAL OF ARCHITECTURAL ENGINEERING* 18.3 (2012), pp. 206–213. DOI: [10.1061/\(ASCE\)AE.1943-5568.0000074](https://doi.org/10.1061/(ASCE)AE.1943-5568.0000074).
- [6] Branko Kolarevic. “PARAMETRIC EVOLUTION”. In: *Inside Smartgeometry: Expanding the Architectural Possibilities of Computational Design* 1 (2013), pp. 50–59. DOI: ISBN:978-1-118-52247-9.
- [7] Helmut Pottman et al. “Architectural geometry”. In: *Computers & Graphics* 47 (2015), pp. 145–164. DOI: [http://dx.doi.org/10.1016/j.cag.2014.11.002](https://dx.doi.org/10.1016/j.cag.2014.11.002).
- [8] Hans Schober. *Transparent Shells Form Topology Structure*. Berlin, Germany: Wilhelm Ernst & Sohn, 2015.
- [9] Ian Liddell. “Frei Otto and the development of gridshells”. In: *Case Studies in Structural Engineering* 4 (2015), pp. 39–49. DOI: [http://dx.doi.org/10.1016/j.csse.2015.08.001](https://dx.doi.org/10.1016/j.csse.2015.08.001).
- [10] Department of Civil and Environmental Engineering Princeton University. *The Courtyard Roof of the Museum of Hamburg History*. 2013. URL: <http://shells.princeton.edu/Ham.html>.
- [11] Marg gmp Architects von Gerkan and Partners. *gmp Architects von Gerkan, Marg and Partners*. 2017. URL: <http://www.gmp-architekten.com/projects/museum-of-hamburg-history-glass-roof.html>.
- [12] Fosters and Partners. *Fosters and Partners*. 2017. URL: <http://www.fosterandpartners.com/projects/great-court-at-the-british-museum/>.
- [13] Chris J K Williams. “The Analytical and Numerical Definition of the Geometry of the British Museum Great Court Roof”. In: *Mathematics & design* (2001), pp. 434–440. DOI: ISBN0-7300-2526-8.

- [14] NEY & Partners. *Glass roof Dutch Maritime Museum*. 2017. URL: <http://www.ney.be/project/glass-roof-dutch-maritime-museum.html>.
- [15] Sigrid Adriaenssens et al. *Shell Structures for Architecture*. Abingdin, Oxon: Routledge, 2014.
- [16] Henning Larsen Architects. *Henning Larsen Architects*. 2016. URL: <http://www.henninglarsen.com/news/archive/2016/09/new-town-hall-in- uppsala.aspx>.
- [17] Henning Larsen Architects and Uppsala Kommun. *Uppsala Nya Stadshus Uppdaterat Tävlingsförslag*. Uppsala, Sweden, 2016.
- [18] Inc Continuum Analytics. *Anaconda Powered by Continuum Analytics*. 2017. URL: <https://www.continuum.io/>.
- [19] Michael R. Barnes. “Form Finding and Analysis of Tension Structures by Dynamic Relaxation”. In: *International Journal of Space Structures* 14.2 (1999), pp. 89–104. DOI: <https://doi-org.focus.lib.kth.se/10.1260/0266351991494722>.
- [20] W. J. Lewis. “The efficiency of numerical methods for the analysis of prestressed nets and pin-jointed frame structures”. In: *Computers and Structures* 33.3 (1989), pp. 791–800. DOI: [https://doi.org/10.1016/0045-7949\(89\)90254-X](https://doi.org/10.1016/0045-7949(89)90254-X).
- [21] Michael R. Barnes. *Form finding and analysis of tension space structures by dynamic relaxation*. London, Great Britain: Unpublished Doctoral thesis, City University London, 1977.
- [22] D.Veenendaal and P.Block. “An overview and comparison of structural form finding methods for general networks”. In: *International Journal of Solids and Structures* 49 (2012), pp. 3741–3753. DOI: <http://dx.doi.org/10.1016/j.ijsolstr.2012.08.008>.
- [23] Allison B. Halpern Serguei Bagrianski. “Form-finding of compressive structures using Prescriptive Dynamic Relaxation”. In: *Computers and Structures* 132 (2013), pp. 65–74. DOI: <http://dx.doi.org/10.1016/j.compstruc.2013.10.018>.
- [24] Robert D. Cook et al. *Concepts and Applications of Finite Element Analysis*. Hoboken, NJ: John Wiley & sons, 2002.
- [25] CEN. *Eurocode 1 - Actions on structures - Part 1-3: General actions - Snow loads*. Brussels, Belgium: European Committee for Standardization, CEN, 2003.
- [26] Rolf Klein. “Abstract voronoi diagrams and their applications”. In: *Noltemeier H. (eds) Computational Geometry and its Applications* 333 (1988). DOI: 10.1007/3-540-50335-8_31.
- [27] COMSOL Inc. *Comsol*. 2017. URL: <https://www.comsol.se/>.
- [28] Amin Ghali, Adam Neville, and Tom G. Brown. *Structural Analysis A Unified Classical and Matrix Approach*. Abingdin, Oxon: CRC Press, 2009.
- [29] Jonas Lejon. Personal communication. Comsol support. May 2017.
- [30] CEN. *Eurocode 3 - Design of steel structures - Part 1-1: General rules and rules for buildings*. Brussels, Belgium: European Committee for Standardization, CEN, 2005.
- [31] Peter W. Christensen and Anders Klarbring. *An Introduction to Structural Optimization*. Netherlands: Springer Netherlands, 2009.
- [32] Comsol. *Structural Mechanics Module User’s Guide*. Stockholm, Sweden: Comsol, 2016.

Appendix A

Dynamic Relaxation Script

```

#Dynamic Relaxation script V2.1
#Henrik Green & Daniel Lauri
#Updated: 2017-04-05

def Dynamic_Relaxation(x,xi,xf,C,Ci,Cf,ai,ci,cf,rho,E,A_steel,A_grid,
                      gamma_glass,sk,tol,count,phiE,directory,
                      f=0,z_i=0,n_f_i=0):
    import numpy as np

    import os
    os.chdir(directory)

    ###-----Dynamic relaxation-----

    from matplotlib import pyplot as plt

    #m number of branches
    m=(len(C[:,0]))//3
    #ni number of interior nodes
    ni=len(xi)//3
    #nf number of boundary nodes
    nf=len(xf)//3
    #Creates initial interior force vector, f0
    f0x=np.ones((m,1))*f
    f0y=np.ones((m,1))*f
    f0z=np.ones((m,1))*f

    f0=np.concatenate((f0x,f0y,f0z))

    x=np.concatenate((xi,xf))

    #Creates a node difference vector
    u=np.dot(C,x)

    #Reshapes u into 3xm, one row for each coordinate component x,y,z
    u=u.reshape(3,m)

    #Node differences size [mxm]
    U=np.diag(u[0,:])
    V=np.diag(u[1,:])
    W=np.diag(u[2,:])

    #Branch Lengths size [mxm]
    L_0=np.sqrt(U**2+V**2+W**2)

    #Total Length of members in grid
    L_=L_0

    L_diag=np.diag(L_)

    L_ind=[]
    #Extracts indices of elements longer than 40/count
    for i in range(m):
        if L_diag[i]>0.9*40/count:
            L_ind.append(int(i))
        else:
            pass

    #Index List converted to array
    L_index=np.array((L_ind))

```

```

#Mean Length of elements to calculate glass area
L_mean=np.mean(L_diag[L_index])

#Length of grid members
L_grid=np.sum(L_)

#Roof height
h=np.max(xi[int((2/3)*np.shape(xi)[0]):np.shape(xi)[0]])
#Area of glass panels to approximated by equilateral triangles
#with side=mean(element length)
A_glass=(count**2)*(L_mean**2)

#Weight glass [N]
W_glass=A_glass*gamma_glass
#Weight snow [N]
W_snow=A_grid*sk

#Calculates the total on the planar grid [N]
q=(L_grid*A_steel*rho*9.81)+W_glass+W_snow

CiT=np.transpose(Ci)

#Creates L0 array of initial branch lengths [3mx3m]
L0=np.zeros((3*m,3*m))

for i in range(m):
    for j in range(m):

        L0[i][j]=L_0[i][j]
        L0[m+i][m+j]=L_0[i][j]
        L0[2*m+i][2*m+j]=L_0[i][j]

### ----- Change coordinates on interior boundary -----
#If there are interior boundary nodes
if n_f_i!=0:
    #Loop to change their z-coordinate to z_i
    for i in range(n_f_i):
        xf[-n_f_i+i,0]=z_i
else:
    pass

#Time step
dt=1

#Kinetic energy KE to initiate loop
KE_1=0
KE0=0
KEc=2*tol

v0=0
xi0=xi

#-----Loop-----
it=0
KE=[]
#Equivalent stiffness
EA_eqv=np.ones((3*m,1))*phiE

```

```

#Relaxed Length for which the branch strain is zero
Lr=np.diag(L0)*1.0
Lr=Lr.reshape((3*m,1))

while KEc>tol:
    #Roof height
    h=np.max(xi0[int((2/3)*np.shape(xi0)[0]):np.shape(xi0)[0]])
    #Total Length of members in grid is updated
    L_grid=np.sum(L_)

    #Vecor with element Lengths
    L_diag=np.diag(L_)

    #Mean length of elements to calculate glass area
    L_mean=np.mean(L_diag[L_index])

    #Area of glass panels to approximated by equilateral triangles
    #with side=mean(element Length)
    A_glass=(count**2)*(L_mean**2)

    #Weight glass [N]
    W_glass=A_glass*gamma_glass

    #Calculates the total load on the planar grid [N]
    q=(L_grid*A_steel*rho*9.81)+W_glass+W_snow

    #Loop to distribute the total Load according
    #to the voronoi area of each node
    pz=np.zeros((ni,1))

    #If self stress - do not change Load from zero
    if f!=0:
        pass
    #Else distribute total updated Load according to voronoi area
    else:
        for i in range(len(ai)):
            pz[i,0]=((ai[i]/A_grid*q)*(1/(E*A_steel)))

    px=np.zeros((ni,1))
    py=np.zeros((ni,1))

    #Vector of external Loads at each node
    p=np.concatenate((px,py,pz))

    #Concatenate nodal coordinate vecor
    x0=np.concatenate((xi0,xf))

    #Creates a node difference vector
    u=np.dot(C,x0)

    #Reshapes u into 3xm, one row for each coordinate component x,y,z
    u=u.reshape(3,m)

    #Node differences, size [mxm]
    U=np.diag(u[0,:])
    V=np.diag(u[1,:])
    W=np.diag(u[2,:])

```



```

#Branch Lengths, size [mxm]
L_=np.sqrt(U_**2+V_**2+W_**2)

#Creates U array [3mx3m]
U=np.zeros((3*m,3*m))

U[0:m,0:m]=U_
U[m:2*m,m:2*m]=V_
U[2*m:3*m,2*m:3*m]=W_

#Creates L array [3mx3m]
L=np.zeros((3*m,3*m))

for i in range(3):
    L[i*m:i*m+m,i*m:i*m+m]=L_

#Creates mass vector, mv
mv=(0.5*(dt**2))*(np.abs(CiT)).dot(np.linalg.solve(L,f0)+
    np.linalg.solve(L0,EA_eqv))

#Creates Lumped mass matrix M
M=np.diag(mv[:,0])

#Strain vector
Lt=np.diag(L)
Lt=Lt.reshape((3*m,1))
dl=Lt-Lr
eps=dl/Lr

#Residual forces
r=p-(CiT.dot(U)).dot(np.linalg.solve(L,f0)+
    np.linalg.solve(L,EA_eqv*eps))

#Velocity at t+dt/2
v1=v0+dt*(np.linalg.solve(M,r))

#Coordinates at t+dt
xi1=xi0+dt*v1

#Calculate Kinetic energy of system
KE1=np.sum(0.5*M.dot(v1**2))
KEc=KE1

D0=KE_1-KE0

D1=KE0-KE1

if KE1>KE0:
    #Updates Kinetic energies, velocities and coordinates

    xi0=xi1
    v0=v1

#Apply kinetic damping
elif KE1<=KE0:

    #Updates Kinetic energies, velocities and coordinates
    dts=dt*(D1/(D1-D0))

```

```

        xi0=xi1-dt*v1-dts*v0
        v0=0
        KE0=0
        KE1=0

        #Updates kinetic energies for next step
        KE_1=KE0
        KE0=KE1
        KE.append(KE0)
        it=it+1

    if it==10000:
        break

#Plots convergence
    axes=plt.gca()
    axes.set_ylim([-0.1*max(KE),1.1*max(KE)])
    plt.xlabel('Iteration number')
    plt.ylabel('Kinetic energy')
    plt.title('Convergence DR')
    plt.plot(np.arange(it),KE)

#Physical nodal Loads in [N] for use in FEM
    Pz=pz*E*A_steel

    xi=xi0
    #Returns coordinates of nodes, xi and xf,
    #vector of vertical Loads, Pz and area glass
    return [xi,xf,Pz,q,A_glass]

```