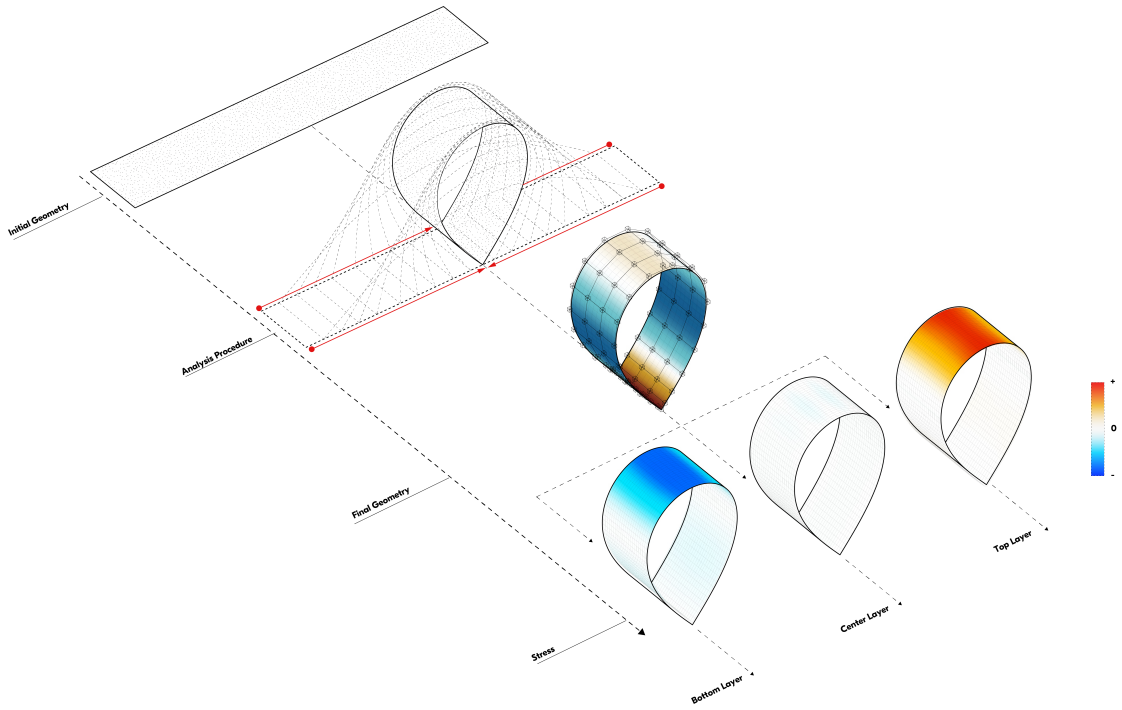
CHALMERS
UNIVERSITY OF TECHNOLOGY



Isogeometric Analysis and Form Finding

An integrated approach to the modeling of thin, elastically bent shells in a design context

Master's thesis in Structural Engineering and Building Technology

Joel Hilmersson

Department of Industrial and Materials Science, Dept of Computational Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019:NN

Isogeometric Analysis and Form Finding

An integrated approach to the modeling of thin, elastically bent shells in a design context

JOEL HILMERSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Industrial and Materials Science
Division of Computational Mechanics
Chalmers University of Technology
Gothenburg, Sweden 2019

Isogeometric Analysis and Form Finding
An integrated approach to the modeling of thin, elastically bent shells in a design context
JOEL HILMERSSON

Supervisor: Mats Ander, Jens Olsson
Examiner: Fredrik Larsson, Department of Industrial and Materials Science

Master's Thesis 2019:nn
Department of Industrial and Materials Science
Division Computational Mechanics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Forward Prediction

Typeset in LaTeX
Gothenburg, Sweden 2019

Isogeometric analysis and Form Finding
An integrated approach to the modeling of thin, elastically bent shells in a design context
JOEL HILMERSSON
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

Recent developments within the design of shells have seen an increased interest in utilizing active bending as form giving procedure [1]. This enables complex structures to be built from simple off-the-shelf materials. However, forming bending-active structures is highly dependent on the material properties, which makes the design process reliant on either physical testing or digital simulations. An associated problem with the simulation of this behavior is the lack of integration between modeling and analysis in conventional simulation techniques, a crucial concern since the final design is always an equilibrium shape with requirements on both structural and spatial integrity. IsoGeometric Analysis (IGA) is a method that aims to bridge precisely that gap between analysis and design, making it a suitable method for bending active structural design.
This thesis explores an approach to the modeling and digital design of actively bent shells using the implementation of nonlinear IGA. Further on, two different ways of controlling the geometry, either by following the process *forward* and predicting the outcome or tracing it *backwards* during the construction procedure are proposed. Following the process forwards returns an implicitly controlled shell geometry through stepwise displacement of the boundaries of a flat sheet. However, as a design approach, one appealing strategy is to explicitly control the final geometry by a *backwards* tracing. This allows the designer to start from a desired outcome and instead tailor the stiffness to approximate this desired form. The procedure is tested in a case study where a combination of both *forward* prediction and *backwards* tracing is included. Both processes apply the Kirchhoff-Love shell theory [27] and uses the total Lagrangian formulation for the nonlinear computations.

Keywords: Isogeometric Analysis, Shell Design, Bending Active, Nonlinear, Form Finding.

Acknowledgements

First and foremost I would like to thank my examiner Fredrik Larsson for all the knowledge you provided me with during the course of this work, without which it would not have been possible. Secondly I would like to thank Mats Ander and Jens Olsson for their interest and insights, and for often providing interesting reflection on my work in a bigger perspective.

I am also very grateful for Sara Almstedt and Puria Safari for providing me with their implementation which helped me much with my own implementation and made this project a whole lot easier.

Finally, thank fellow thesis students, opponents and good friends Malin Borgny and Linda Wallander for their comments and general support along the way.

Joel Hilmersson, Gothenburg, June 2019

Contents

List of Figures

List of Tables

1
Introduction

This chapter presents the initial aims and motivations for carrying out the work that follows in the thesis. It will start with the initial setting and proceed to clarify why and in what way the following study has been done. Finally it will also list the limitations that had to be applied given the selected focus and priorities.

1.1 Context

The background of this thesis proposal lies in two separate, yet highly entwined, topics that are currently gathering much interest within the field of structural design and architectural geometry [3]. Those are the topics of integrated design and analysis by bridging the geometric gap and the application of elastic deformation as a form giving procedure. Below follows a brief of them, along with their relation to the thesis work.

1.1.1 Integrating Design and Analysis

The design process of today is to a great extent dependent on digital representations of geometry, which is the main tool to describe the form of potential objects. During the recent years, Non-Uniform Rational B-Splines (NURBS) has emerged as a highly popular method for designing and controlling curved geometries. However, this is a different form of representation compared to the one used in CAE processes, where analysis mainly operates on polygonal meshes. With the design of curved surfaces there is often no difference between the form of the object and the design of the structures, as the structure is the surface. Although there is no difference between the geometries, the two different processes (Analysis and Design) are using two different models, with two different geometric descriptions. The main reason for this is the fact that the modern CAD technologies were developed years before the equations and procedures of mesh based finite element methods [6]. Thus there exists one model for geometry and another one for analysis, and engineers spend time converting the design geometries into analysis suitable geometries by discretizing them. These meshing processes can often take time and provide no additional benefits apart from suitability with current analysis procedures. Therefore, integrating a common geometry basis would bridge a gap between these two aspects of design, and hopefully bring them closer to each other.

A new method was proposed by Hughes et al. [1], called IsoGeometric Analysis, or IGA for short, which aims to close this geometric gap and use the same underlying geometric principle. The method aims to integrate a common geometry basis for both design and analysis by adapting the mathematical description of NURBS instead of the regular meshes and basis functions used in regular finite element modeling. The method has shown great potential and seems to provide more accurate analysis than conventional FEM [6] for the same number of dofs. Previously students Puria Safari and Sara Almqvist have studied the implementation of, among other things, thin shells using IGA. As geometry modeling proceeds analysis, the logic would be to impose the analysis constraints and information directly on the design geometry, thus making analysis a step closer to design, bridging the potential for interdisciplinary design processes. The topic of IGA has very recently gathered interest within the AEC (Architecture, Engineering, Construction) industry, with a workshop on the topic being held at both the IASS Symposium and the Advances in Architectural Geometry Conference in 2018) [3].

1.1.2 Bending Active Structures

Recently, an interest in the use of bending active structures have emerged. The term "Bending active" was coined by Julian Lienhardt in his Ph.D dissertation at the university of Stuttgart [2] and refers to freeform structures which get their shape from the deformation of initially planar elements. By constructing curved geometries this way one can create complex structures with rather simple methods and materials.

However, the process of constructing these structures can make them hard to design. Here, the design is the result of large deformations, which makes the the shape directly related to physical behavior. The final shape needs to be found as an equilibrium shape, based on the initially planar member [2]. The topic has been subject to much research recently by, among others [5] [17] [4]. It was also the topic of two workshops at the Advances in Architectural Geometry conference hosted here at Chalmers University at the time the research of this thesis was done (*Sept 2018*) [3].

1.1.3 Physical behaviour and Design

A perception of the potential of IGA is that it is in the conceptual and design oriented phases of a project were the common geometry framework is the most useful [6]. Lately, a trend in design processes has grown increasingly more interested in blending physical and digital processes [12]. One interesting potential of IGA therefore lies in dealing with the surfaces with regards to their physical properties. Within the field of designing shells, a commonly used term is that of form finding. Form finding is commonly known as the process of finding the form of a structure using some underlying logic or demands. In the context of this thesis it will refer to the process of finding the form of a shell segment as the result of a deformation process.

With the current developments in computation, the digital and physical world can now be interconnected, giving the material aspects a greater role in the design of the form of an object or building. Isogeometric analysis can then form a natural framework for implementing actual physical behaviour as a natural part of the design process. In actively bent structures the final surface is not necessarily modelled but derived from the construction process. These new methods can provide an essential link to predict and use physical behaviour as a driver for form.

As mentioned, with bending active it means that the geometry is the result of a deformation process. The role of the deformed geometry is then different than the traditional. Deformation is today only interesting as a means of understanding the behaviour and not as an actual form. This is not the case with bending active structures. Isogoemetric analysis can provied a common framework for the simulation and the most common form of geometric representation. Furtermore detailing etc might need to be done on the deformed geometry. Here a NURBS based description would potentially be more favourable to work with as a design base.

1.2 Purpose

The essence of the thesis would be to investigate how the previously mentioned shells can be designed and analyzed in a suitable way by integrating the structural behaviour through the application of large displacement IGA. The main focus is to study the application of a novel numerical method as a design tool. Further, this study will focus on how the stiffness distribution through the thickness can be used to manipulate the final form.

However, developing a tool for a specific process should also be concerned with the practical aspects of the problem. The topic of shells and their geometric description will therefore be touched upon in order to understand what one could expect from a form finding tool. In part an approach and understanding of how to put bent pieces together in a logical way that makes for a good shell design is also needed, to steer a suitable modeling approach in a useful direction.

Based on this the computational mechanics aspects should be derived and implemented in a tool that will enable a seamless profile. While designing the above seen smooth surfaces a NURBS based design

software would be ideal to get a good shape control over the smooth surfaces.

The aim is therefore to implement an IGA based simulation into a NURBS based modeling software to bridge the gap between analysis and design and to use the actual deformation behaviour as a means of design.

The following questions should therefore be the foundation for the work.

- **How can we integrate the numerical simulations of the physical behaviour into a design process in a seamless way, for the design of actively bent forms?**
    1. How does one deal with the digital design of assemblies of bent sheets as a shell structure?
    2. What is needed to achieve a robust design tool that uses IGA?

- **How can manipulation of the stiffness distribution be used to alter the design?**

1.3 Method

The work takes off from previous work done at the faculty with the linear shell models developed in the thesis of Puria Safari and Sara Almstedt [6]. There, a linear implementation for Kirchhoff-Love shells was developed using MATLAB.

The work will initially do a quick review of shell structures as an input to how this problem could be approached. Further it will investigate the theory and implementation needed to solve the large displacements for the shells. This will be done in two steps. An initial step will deal with shells condensed into two dimensions, thus turning them into 2d beam elements. Through this various solution procedures can be studied along with their suitability for the intended design task. This will also serve as a testing ground for the control of the process. In the subsequent step, the knowledge from the beam case will inform the implementation of the tool for the shell case.

The development of the final tool will, due to time limitations remain with MATLAB as the framework for numerical calculations. The beam will solely be developed in MATLAB. Given the ease it offers in debugging and variable overviews etc it makes a good framework to test implementations. As the time frame for the this is to short to reprogram the existing shell code along with the nonlinear procedures, the design tool will use MATLAB remotely. The interface will be programmed in visual studio for the commercial package Rhino, and MATLAB will be called externally when calculations are needed. This is not an ideal process, but as MATLAB provides this framework, it will be the easiest, and not a too poor way.

The final outcome will be a tool along with a design approach. However, the application of the tool will span into wider areas where large displacement IGA can be useful. The design case provides merely one example and there are most likely lots of other applications for this.

1.4 Limitations

The scope of the project is quite large and therefore some limitations are necessary. The first limitation, is that the thesis will focus solely one the application of IGA. Further, as the design case feature very thin shells only Kirchhoff-Love kinematics will be considered.

Secondly, given the segmented nature of the shells, as will be seen later, they will be considered as single patches and multi patch continuity will not be studied. The main focus of the thesis will be put on forming flat, single NURBS patches into a curved form.

The process and integration is the main goal, and aspects related to analysis, such as selection of integration schemes and such will not be elaborated on. Same goes for accuracy comparing to regular FEA.

Here it is referred to studies other studies such as [6].

Given the focus on numerical methods and geometry, material effects as nonlinear behaviour or creep and similar effects will also be neglected. The processes will therefore be limited to applications concerned with small strains which allows for the use of linear material behaviour according to St. Venant theory to be used. Also, when using more complex stiffness distributions, one needs to make sure the integration scheme can handle it. This thesis will more investigate potential means of controlling deformation as a design approach, and leave details of convergence and selection of integration schemes etc out. Convergence will be checked simply by refining the mesh or increasing the polynomial degree.

1.5 Structure of the Report

The report is structures in the following way. First, an introduction to the field of shells structures is presented. This serves as overview of the design of shell structures and focus is put on the geometric description of shells. Further the following chapter provides an introduction to NURBS as a geometric description together with the fundamental aspects of isogeometric analysis. This thesis assumes that the reader is already familiar with traditional FE-methods, and the chapter translates the concepts of FEM to IGA using NURBS as a geometry base. Further a chapter concerning the theory applied and used to solve the problems are stated. The theory chapter is mainly concerned with general relationships, discoupled from NURBS, and could thus be applied with different geometric descriptions. On that note a chapter follows on the computer implementation and solution procedures associated with the theory and aims. Chapter 6 then introduces the developed framework for integrated design making use of the previous chapter which is followed by a chapter applying the framework to a set of basic problems.

2

Design of Shell Structures

With the outset of thesis established, the first step is a study on the critical aspects of designing shells. This will serve as a starting point for the processes laid out further on. This chapter will be an introduction into the discourse of the design of shell structures. It will cover some basic principles of designing curved support structures along with some considerations. It will first give some examples of shell structures and further advance into the field of the application of bending active structures.

Eventually it will end up with the proposed system that the methods developed later in the thesis aims to create. The general focus is on the process of finding or defining the shape of a shell, and not so much on the analysis and assessment of the shell.

2.1 The General Perspective

Shells are found in many applications. It's found in the bodies of cars and planes, in the case that holds your glasses or as civil engineering structures. However, the general meaning of what is meant with a *shell* might not be clear to the uninitated reader. Therefore, the first question we need approach, might seem elementary, but is a necessary one. That is the question of:

- *What is a shell structure?*

Most people are familiar with the word in an everyday sense, however, within the field of civil engineering structures it might not be as clear. The word in itself might direct thought towards the hard and thin outer surface of eggs, which actually is a good representation of the word.
A description of shells is given by Chris Williams in his chapter *What is a shell?* found in [8], where he defines it as:

> "A *shell* is a structure defined by a curved surface. It is thin in the direction perpendicular to the surface, but there is no absolute rule as to how thin it has to be. It might be curved two directions, like a dome or cooling tower, or it may be cylindrical and curve only in one direction."

However, as the name *shell*, implies something rather rigid, the definition will omit structures such as tensile membranes which match the definition above. The shell is therefore defined as a form-passive structure that carries load through membrane forces (and bending) where the load transfer is made possible through the form of the shell. Since the form of a shell is instrumental to the load bearing behavior, special considerations need to be included in the description of what the form looks like. This, the process of finding a suitable geometry of a shell, is what the upcoming chapter will be about.

Through out the history of shell design various methods have been employed to approach this problem. Either mathematical models of curved surfaces, to physical processes to more recent digital methods. The following chapter will give a brief walk through of these methods, related to some examples.

2.2 Geometric Description of Shells

The following chapter will elaborate on what a form defining process is by looking at some precedent structures and the way their respective geometries were defined.

In Shell Structures for Architecture [8], three different means of defining a geometry for a shell structure is given.

The different categories are as follows:

Mathematical Geometrical or analytical shells are directly described by mathematical formulas. They are chosen for their suitability to perform further analytical calculations on and fabrication purposes.

Form Found These include the natural hanging chains or cloths of Gaudi, Otto or Isler together with "strained" shells that feature bending stresses.

Freeform Free-curved or sculptural shells are generated without taking into consideration structural performance. If they are shaped digitally, then they are often described by higher degree polynomials (eg. NURBS)

It can be seen that some shells operate on a geometric precedent or imposed form, where as some are derived from physical properties. A common term circulating the design of shell structures (which also features in the title of this thesis) is the term *form finding*.

Sigrid Adriaenssens describes form finding as [8]:

> *Form Finding* is a forward process in which parameters are explicitly/directly controlled to find an "optimal geometry" of a structure which is in static equilibrium with design loading.

As can be read from the quote above, a form finding process often refers to a process which, given a certain boundary or body and a set of constraints finds a suitable form for a curved structure, meaning that geometry is the final outcome of this process. It can therefore be seen that this process is concerned with both the design aspects as well as analysis embedding physical properties. Therefore this process is an interesting target for the application of an integrated framework for design and analysis, as form finding often means that analysis is design. In the scope of this thesis the term form finding, will differ slightly from that proposed by Sigrid Adrianssens, and refer to the process of finding the deformation of a piece of material with respect to a set of prescribed boundary conditions.

However, all shells are not traced from a reference geometry and boundaries, and the following chapters will provide an overview as input for the potential application of IGA in our context.

2.2.1 Mathematical

The mathematical definition of a form means a function or mathematically derived typology was used. Two sets of shells can be found that follow this methodology. The main procedure however come from the application of a preset form known to possess required geometrical properties and the reason for this can be seen in the consideration for fabrication constraints.

The two different classes described here will be called *typological* and *free form*. The typological means the application of a known type, whereas the custom is composed of a specific mathematical description to fit a specific case.

2.2.1.1 Typological

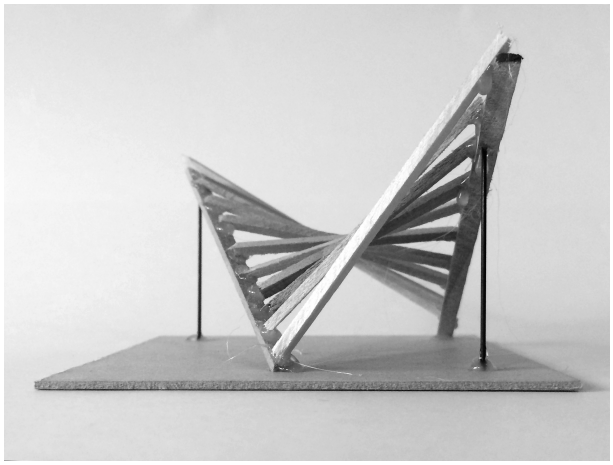


Figure 2.1: Example of a doubly curved geometry constructed from straight segments

The application of typological shells are the use of known forms as hypars and paraboloids or other known and easily described. A famous hypar designer was Felix Candela [10]. The central aspect of using this approach is the simplicity of construction, using for example surfaces that can be constructed from straight lines, or calculability in the age before finite element methods.

2.2.1.2 Custom

Figure 2.2: Roof of Savill gardens (right) and Construction of Weald and Downland (left)

Other examples of using more custom functions to describe a non-standard form is the roof for the great court over the british museum or the roof for the building in savill garden. In both of these cases a surface was described using mathematical functions [11]. The grid distribution was then relaxed onto the targer surface. In this type of formfinding, the overall geometry is not the aim, but explicitly model, instead it's the discretization or distribution of material that is *found*.

Thus, the geometry is not related to the equilibrium of a specific loadcase, but sufficient stiffness of the surface is made sure to be embedded in the imposed geometry. In the savill garden the relaxed grid corresponds to laths meant to be formed from initially straight ones.

2.2.2 Form found

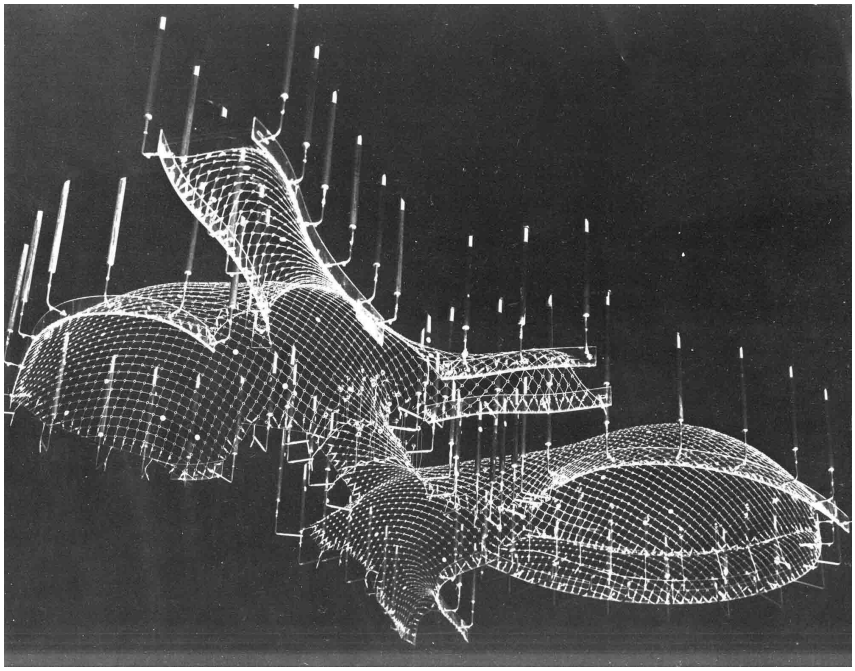


Figure 2.3: Physical formfinding of the mannheim multihalle [33]

Using static equilibrium with respect to a design loading is the process that Sigrid Adriansens refer to with her definition of the word form finding. This means the models done by Heinz Isler were physical cloths were hung under their own weight to find a perfect equilibrium shape. The same approach has been carried out by, among other, Antonio Gaudi and Frei Otto. Figure 2.3 shows the form finding of the Mannheim Multihalle.

The general systematic of this approach starts from a reference body, a set of boundary conditions along with a design load. The body will then adjust to find a static equilibrium with the applied load, thus, for the specific load case, a perfect compression structure.

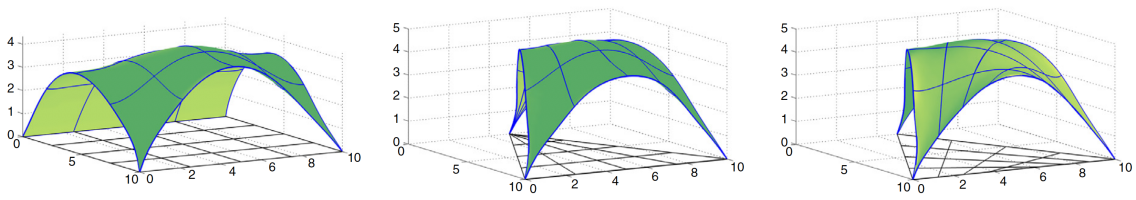


Figure 2.4: Form Fiding with IGA (Vedad Alic) Image from [19]

Recently this procedure has been applied digitally trough, either the force density method or dynamic relaxation of spring based systems. This has also been explored using NURBS and IGA as the numerical tools, which can be seen in the work by Andrea Alexandersson [18] and Vedad Alic [19]. Images show some of the implementations from the work of Vedad Alic.

2.2.3 Freeform

Figure 2.5: Meiso No Mori - Freeform Shell [Toyo Ito/Mutsuro Sasaki] [34]

Some shells are also designed using a free form sculptural surface. Theses may be shaped completely in disregard of any structural or physcial constraints. This is often based on a n architiecturally modeled surface.

However, manipulations can be made to negotiate the surface and it's structural performance. An example of this is the shell roof for the Meiso no Mori funeral hall. An initial design surface was modeled. On this the structural engineer (Mutsuro Sasaki) used a process called sensitivity analysis. This means adjusting the form of the roof in an attempt to reduce the strain energy due to bending. In this case an input surface is used as a starting point for mapping. [8]

2.3 Bending Active Structures

2.3.1 Background

With new developments in simulation and computational tools, new potential approaches emerge as a means to find the form of shell structures. One of these fields is the field of bending active structures. As form finding procedures for traditional shells often are concerned with an equilibrium body w.r.t an external load, bending active structures are also required to be in a state of equilibrium with itself. These structures need the use of advanced numerical procedures to have their behaviour predicted to make sure that the final geometry intended is the one that the deformed pieces will form. The physical properties and material distributions role in the process is essential, making today's structures often be computed through the use of advanced digital simulations and the integration of material behaviour within the form-finding process [29].

The broader term *bending active* applies to a wide range of structures that employ bending as a form giving and stabilization process [4]. This often implies large deformation of initially planar elements, thus enabling the use of simple fabrication by integrating the construction process into the design. These systems make up structures systems emerging from the combination of curved elements [13]. While the traditional engineering approach often is to limit the amount of bending in structures, this type instead uses bending as a construction tool.

Since these structure are so inherently interdependent on analysis as a means of design, they will serve as a base problem for the application of IGA.

2.3.2 Elastically Bent Shells

Figure 2.6: Elastic Segmented Shell, Jan Brütting (University of Stuttgart) Image taken from [28]

Elastic bending has been used as a means of designing structures. One famous example is the Mannheim multihalle. The form was found using hanging chain models, but the final geometry was designed from deforming initially straight laths. Previous examples such as the Weald and Downland and Savill Gardens also used active bending as the construction process. However, these pose examples of gridshells. This thesis will focus on the bending of sheets.

The bending of plates has also been studied. Especially at the ITKE and ICD at the university of stuttgart. Their research pavillions from 2012 and 2016 both emply elastic bending of plywood as a form giving principle [14] [2]. Further Figure 2.6 shows an elastically bent segmented shell from the thesis work of Jan Brütting. Here the elastic properties of the pieces was adjusted to make the deformations fit the target geometry [28]. The planar strips were informed by the intended curvature distribution of the target geometry.

2.3.3 Geometric Considerations

A problem with the deformation of planar sheets is the fact that the modes of deformation are very limited. Only single curved geometries are possible. Gauss showed with his theorema egregium that it is impossible to change the Gaussian curvature of any geometry without stretching the middle plane [8] [4]. The Gaussian curvature is given as the product of the two principal curvatures.

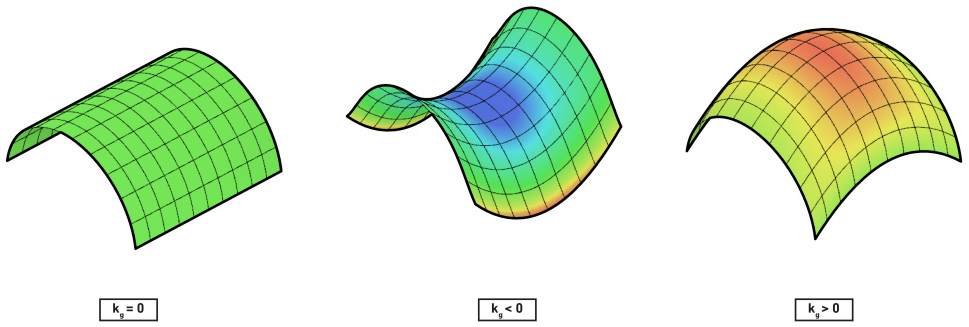


Figure 2.7: Curvature of a set of geometries with zero, negative and positive Gaussian curvature

This means that there only exists two isometries to the plane sheet [4]. An isometry is a form that posses the same lengths. In this context an isometry refers to a deformed body that has not altered any lengths on the body.

The isometries of the plane are:

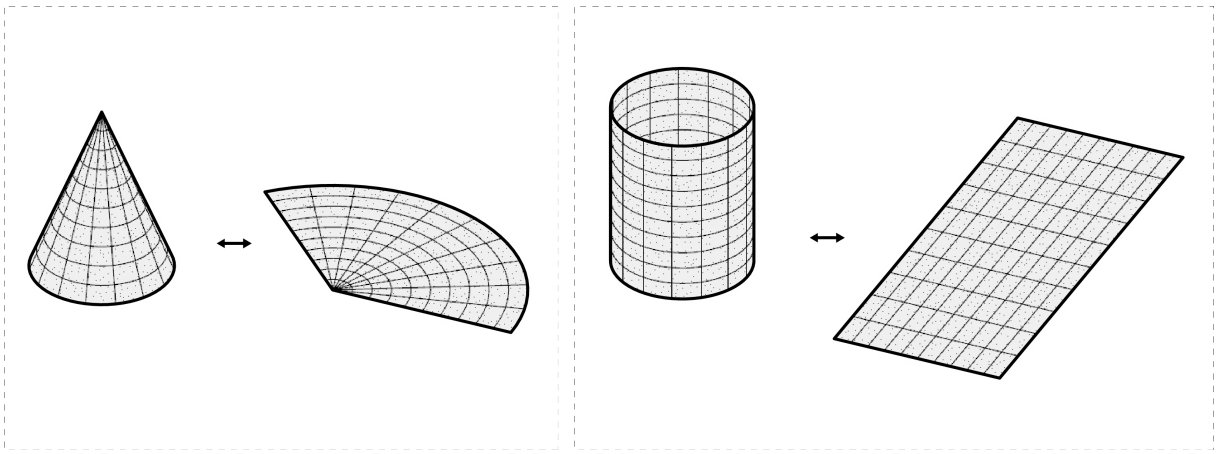


Figure 2.8: Isometries of the plane

However, as these are purely mathematical operations, in a physical object there will always be some stretching and twisting possible, making one able to negotiate the purerly mathematical aspects. However, in order to find a realistic form and not having the process possibly fail, one needs to keep in mind these deformation modes. Still, this illustrates the need for a segmented shell, when dealing with bent sheets, as the modes of deformation are so limited.

2.4 Design Case

2.4.1 Background

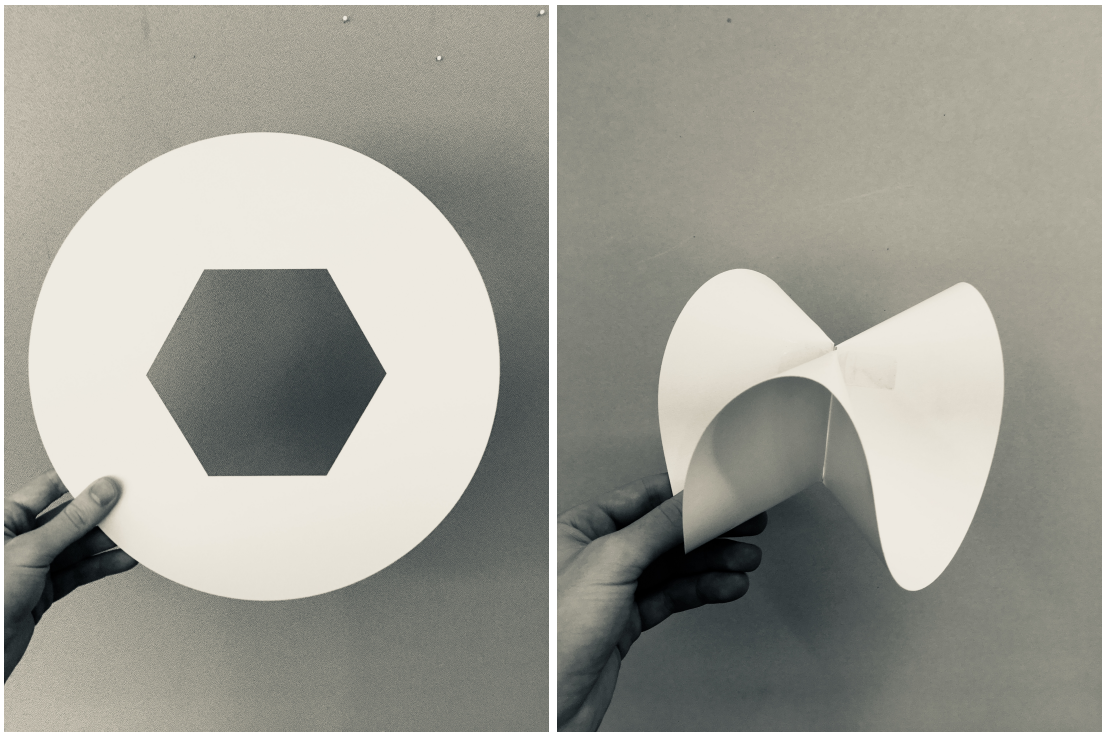


Figure 2.9: Example of Stiffening using Curves and Creases

Given the geometric constraints the proposed shell will be restrained to work with single curved pieces. The design case for this study is based on the logic of feigning double curvature through a combination of discrete creases and smooth curves. As the geometry only bends into a single curved object, see isometries, the connection of sheets at angles will form a discrete second curvature. However, some torsion will occur, which will need to meet the strength of the material. Figure 2.9 shows an example of a simple geometry being turned in to a rather complex, stiff 3-dimensional shape. For the approach how to find the form of something like this digitally, inspiration will be drawn from the previously presented examples. In general two overarching characteristics can be found:

- One were the geometry is sought based on a set of constraints and a load case (like the hanging chain models.) The difference is that the load case here is not fixed boundaries and a design loading, but instead a set of prescribed displacements.
- A second where the geometry is given, and a thickness distribution is mapped onto the target geometry, like the Savill Garden or Weald and Downland gridshells. The material distribution in this case however, will not be a grid but instead a thickness distribution associated with the target geometry and a reference configuration.

2.4.2 Intentions

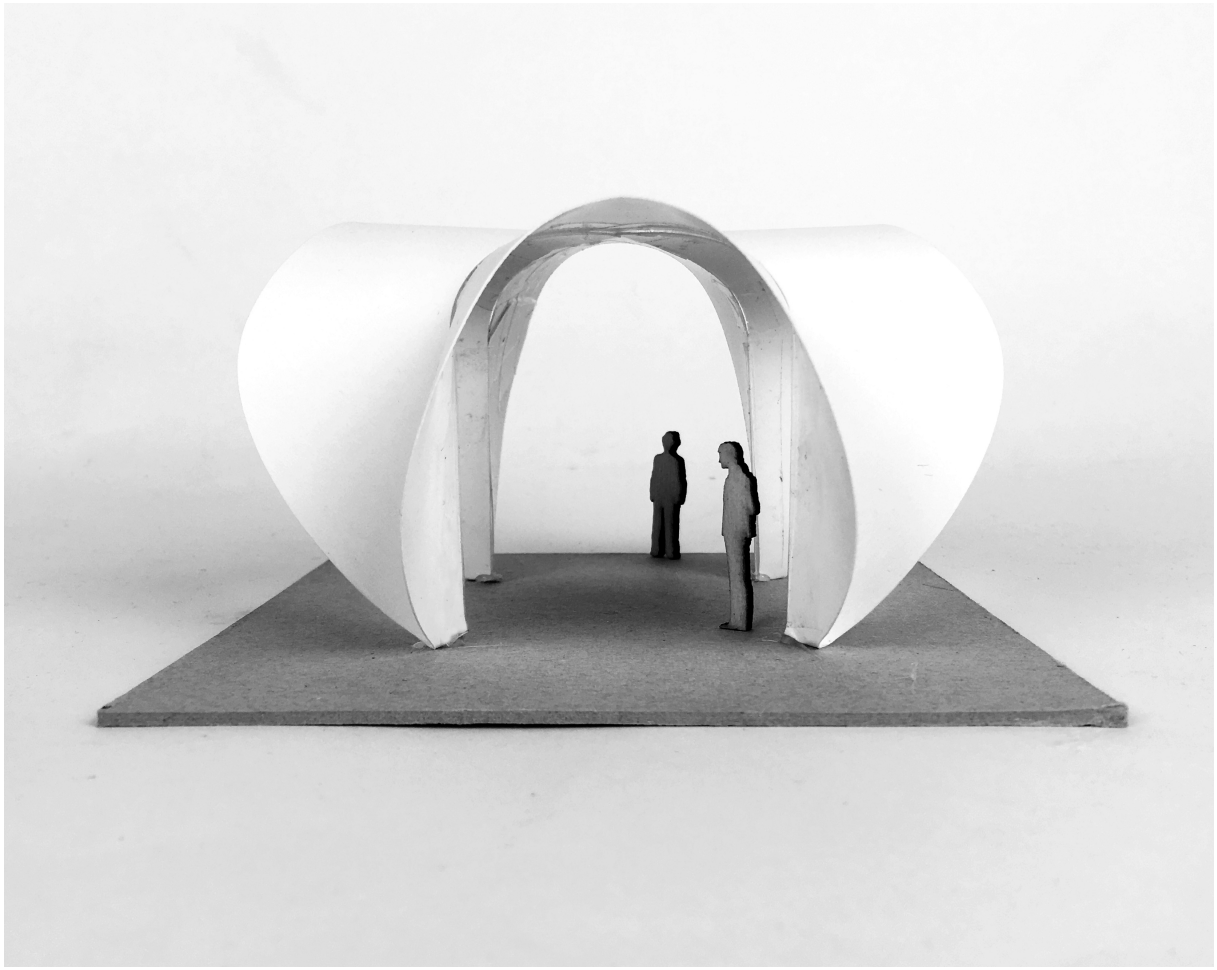


Figure 2.10: Design Case which the process will be targeting

In Figure 2.10 a shell design based on the previous deformation mode pictured. This shell is constructed from the assembly of four bent pieces of paper and is doubly symmetrical. The model depicted is primarily found using physical modeling. This case will serve as the target for the implementation of the Isogeometric Formfinding.

In general, two different approaches will be undertaken, following from the categorisation on the previous page. The two approaches will be refered to as a *forward prediction*, and *backwards tracing*.

With the *forward prediction* method, in this context, we mean a process where a form is found given an reference body and a set of boundary conditions (at the intersections and the ground). This means that the resulting geometry is the unknown in the process, which will then have to be tweaked to find a satifying geometry given demands on appearance and structural performance.

With the *backwards tracing* method, a reference will be used both for the deformed and undeformed geometry. Similar to some of the process before, a thickness distribution will then be mapped onto the piece to make sure make sure the desired geometry is in equilibrium.

An overview of this is given on the next page in Figure 2.11. The forward prediction process starts with step 1 and traces step 2. In the *backwards tracing* process a reference geometry is given (-1) which then maps a thickness distribution to the undeformed geometry. This can then be used to trace a final geometry reassembling the target input.

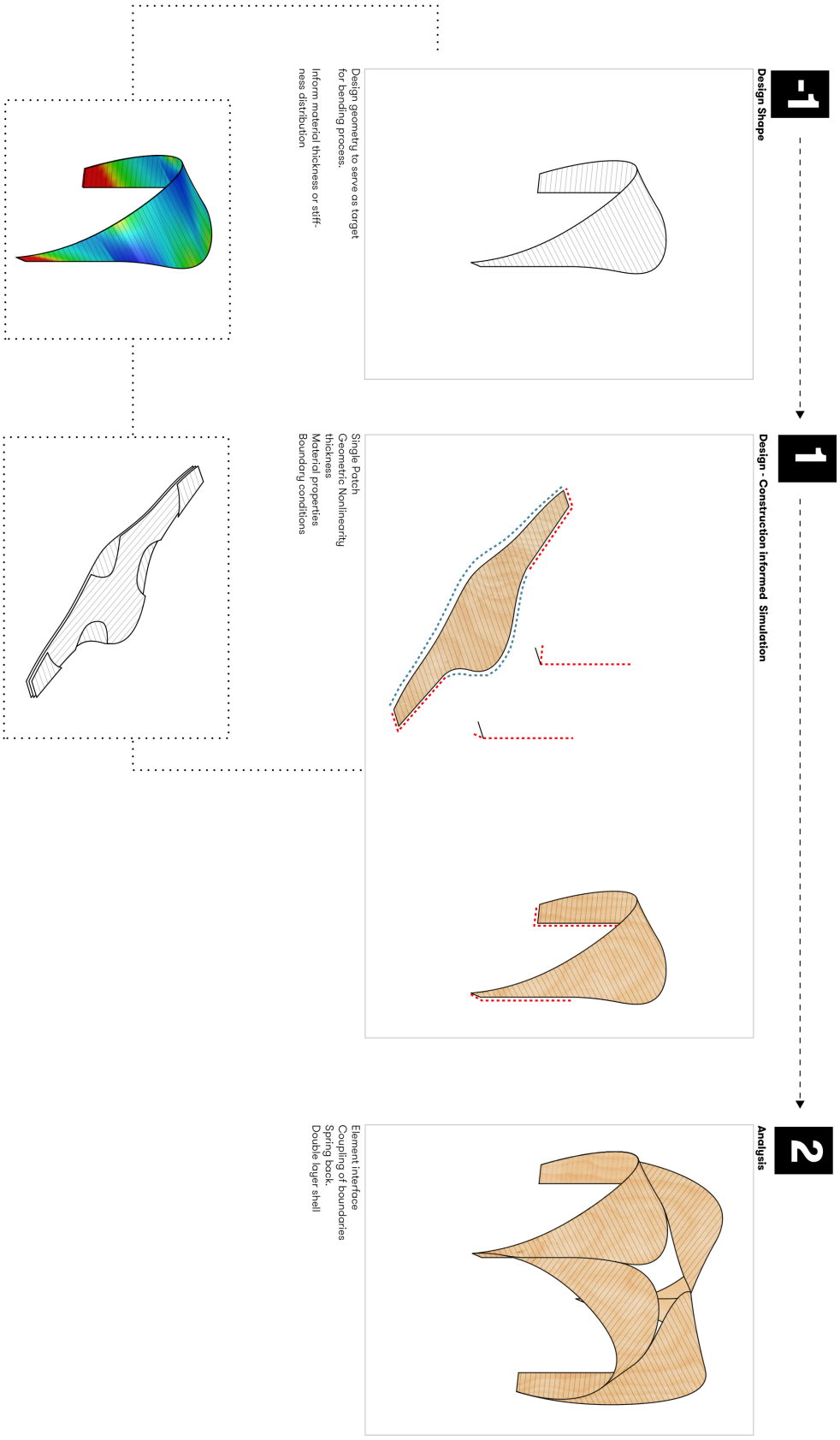


Figure 2.11: Stages of Forming

3

Fundamentals of Isogeometric Analysis

This chapter serves as an introduction to the Isogeometric Analaysis procedures. It covers both the underlying geometry description of NURBS along with how this basis can be used for finite element analysis. This description will be rather brief and additional concepts will can be found in among others [22] [1] [6].

3.1 Basics of NURBS

3.1.1 Background

Within the field of digital design, one often distinguishes two main modeling forms, namely surface or polygon modeling. Mesh modeling lie as the basis for the analysis procedures of today and are based on representing form as an assembly of polygons. This differs from NURBS based surface modeling where one works with interpolated, smooth surface. Figure 3.1 shows some examples of the two descriptions.

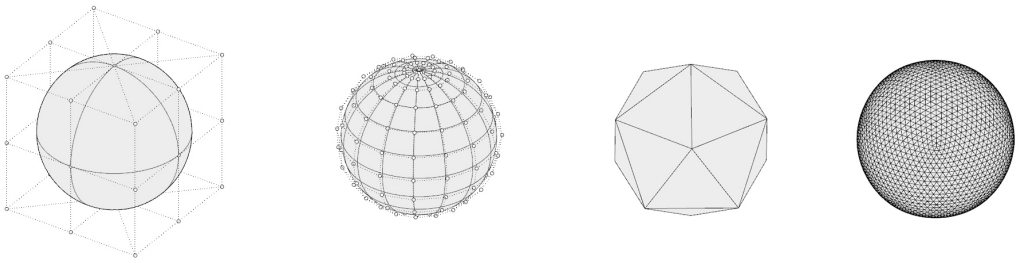


Figure 3.1: Geometric Descriptions showing a NURBS refinement on the left and a polygonal mesh refinement on the right

Surface Descriptions
Often curves and surfaces are represented in geometric modeling as either implicit equations or parametric functions. An example is the curve I(x, y) = 0 in the xy-plane. This equation describes an implicit relationship between the x and y coordinates of the points lying on the curve. An example is the circle of unit radius centered at the origin.

NURBS curves are instead paramteric curves. These are parameterized in a domain. The coordinates then follow from the value of that parameterand they could be conceived as a particle moving through time, where time is the parameter.

3.1.2 NURBS definition

NURBS stands for **N**on-**U**niform **R**ational **B**-**S**pline and is an extension of the B-spline concept. The widespread success of NURBS comes down to a number of factors. One of them being multiplicity as NURBS is able to analytically represent both fundamental geometries as the circle exactly, while

simultaneously being able to represent free form shapes as a car body [22]. NURBS are based on a parametric definition that draws from B-splines and bezier curves.

Components of a Nurbs Definitions
A geometric description relies on a certain set of parameters. The NURBS description consists of the following entities:

[Knots] The knot vector is an increasing ($\xi_{n-1} \leq \xi_n \leq \xi_{n+1}$) number of coordinates within the domain of the parameter space. This list of number controls the distribution of the basis functions over the domain.

[Control Points] The control points is a list of geometric points in physical space through which the curve or surface is interpolated.

[Weights] In NURBS, opposed to B-splines, each point is associated with a certain weight. The greater the weight for a point, the larger the effect on the geometry. Weights where introduced mainly to be able to construct exact representations of for example circles. [22]

[Degree] The degree means the interpolatory degree of a curve. This affects the structure of the knot vector and the smoothness of the curve. The larger the degree the more control points are active in a knot span.

3.1.2.1 Knot vector

The knot vector is given as:

$$\Xi = \{\xi_0, \xi_1, ...\xi_{n+p+1}\} \tag{3.1}$$

A knot vector splits the domain in to a set of knot spans. These can be categorized into zero- and nonzero knot spans, where the nonzero knot spans are those that are spaced along the curve. At each knot one basis function becomes active and an other becomes deactivated. A basis function N_i spans from ξ_i to $\xi_i + p + 1$. Each knotspan therefore have $p + 1$ active points. Each knot can occur multiple times, which will affect the continuity of the geometry. The first and last knot always have multiplicity p+1. The amount of active control points in each span. Knots can be uniform or non uniformly distributed
Knot vector can be open or closed. If it is open the last points are always fully interpolated.

3.1.2.2 B-Splines

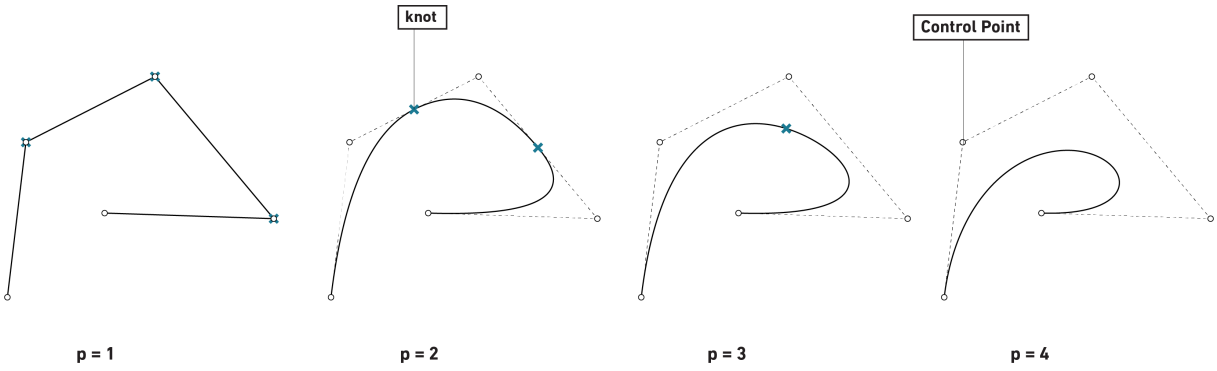


Figure 3.2: Degree elevation from 1 to 4 for an identical set of control points

The mother of the NURBS curve is the B-spline. We will start with a description of the B-spline which then will be further extended into the full NURBS description.
A B-spline curve or surface is based on the interpolation of the aforementioned control points using a set of basis functions, one for each point. The basis function here refers to a similar polynomial as is found in the galerkin discretizations in classical FEA.

A B-spline curve is given as follows:

$$\boldsymbol{C}(\xi) = \sum N^i(\xi)\mathbf{P}^i \tag{3.2}$$

where N is the basis function and **P** is the corresponding control point

The table below shows the knot vectors associated with the curves shown in Figure 3.2. They all have the same amount of points and therefore the same amount of basis functions. However, the amount of nonzero knot spans reduce with each degree elevation. For five points a degree higher than four is impossible. It can be seen that the smoothness increases with each degree elevation. This is due to the order of the basis functions. As the degree increases, more and more points get activated at the same time. In the linear curve, each knot span is affected by two points. In the last p=4 curve the whole curve is one single span, meaning that all points are active over the whole curve, with varying effect. The table below shows the knot vectors for the four examples.

p	Ξ
1	[0 0 0.25 0.5 0.75 1 1]
2	[0 0 0 0.33 0.66 1 1 1]
3	[0 0 0 0 0.5 1 1 1 1]
4	[0 0 0 0 0 1 1 1 1 1]

Table 3.1: Knot vectors for the curves in Figure 3.2

Basis Functions
The basis functions are polynomials that give the influence of each point at a given parameter. Each control point in a curve are associated with a corresponding basis function. The distribution of basis functions along the domain are given by the knot vector and the degree. An example some single span basis functions are given in Figure 3.3.

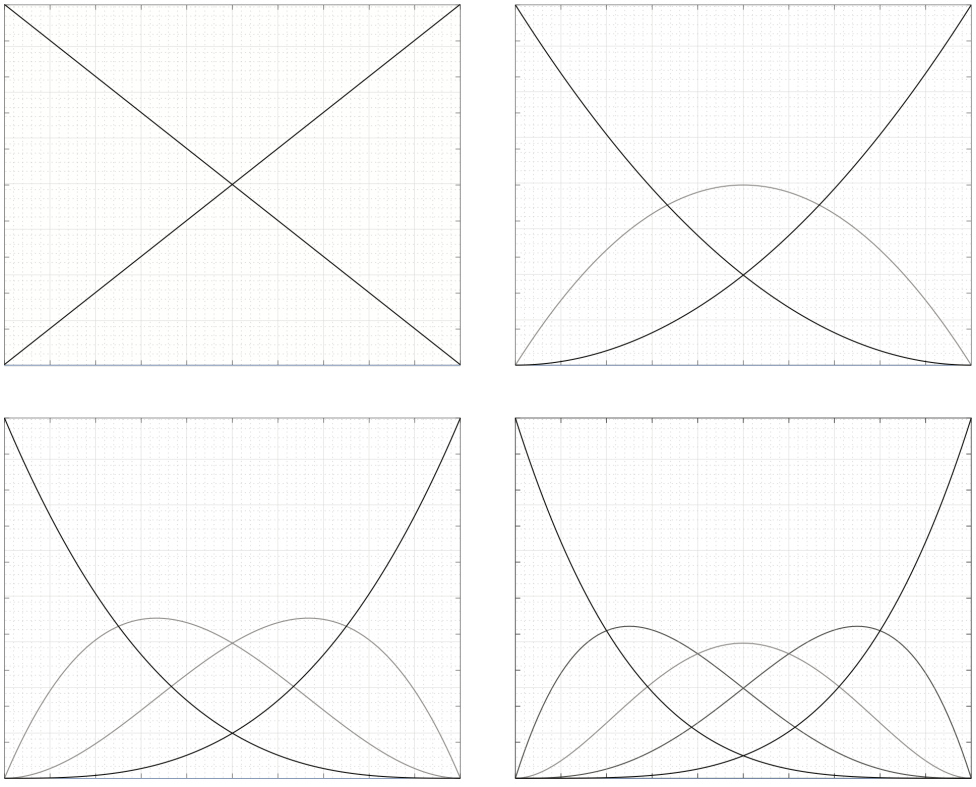


Figure 3.3: Example of basis functions for one single knotspan

Given a knot vector and a polynomial degree, a Cox-deBoor recursive formula is used to compute the basis functions. In order to compute the basis functions for an interpolation of degree p one needs the basis functions for *p-1*. Therefore, the definition starts with the piecewise constant functions of the 0th degree functions.
In each knot span there are *p+1* active basis functions.

$$N_{i,0}(\xi) = \begin{cases} 1 & if\ \xi_i \leq \xi \leq \xi_i \\ 0 & otherwise \end{cases} \tag{3.3}$$

Which states that the basis function is equal to 1 in the corresponding knot span, otherwise it is zero.

From this the basis for $p \leq 1$ interpolation can be found as:

$$N_{i,p} = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{3.4}$$

The derivative of a parametric curve can be found as (in 2d)

$$\mathbf{C}'(\xi) = \begin{bmatrix} x_{1'\xi} \\ x_{2'\xi} \end{bmatrix} \tag{3.5}$$

Thus, the slope of a curve can be found through the derivative of the basis functions.

3.1.2.3 NURBS Basis

The NURBS basis extends the B-spline definition to a 4D space. Each point now consists of $P_i[x_i, y_i, z_i, w_i]$ where w denotes the weights. The NURBS curve is the projection of this onto 3D space, meaning it can be dealt with similarly to the previously mentioned B-Splines. The defintion of the NURBS basis functions is given by:

$$R_i^p = \frac{N_{i,p}(\xi) w_i}{\sum_{j=1}^{n} N_{j,p}(\xi) w_j} \tag{3.6}$$

Giving the function for a NURBS curve as:

$$\mathbf{C}(\xi) = R_i^p(\xi) \mathbf{P}_i \tag{3.7}$$

The main motivation for extending B-splines into NURBS is the possibility to represent exact analytical geometries such as circles. The weight alteres the influence of specific points- Figure shows a 2nd degree curve, one where the weights are modified to match a perfect circle. The simple second order polynomial will not have aconstant curvature, and will therefore devaite from the perfect circle, which can be seen in Figure 3.4.

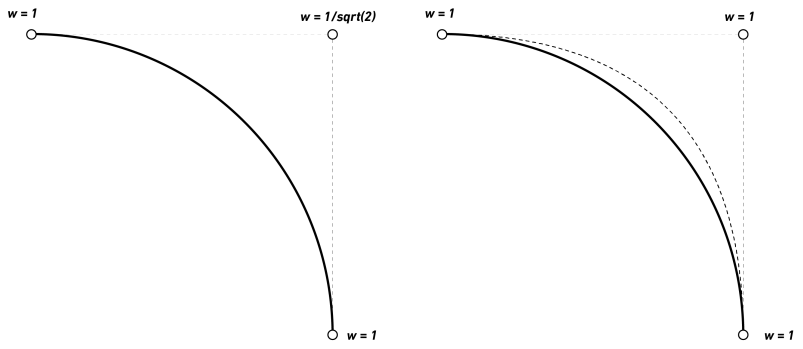


Figure 3.4: NURBS with weights describe perfect quarter circle (Left) B-Spline with constant weights comparing to perfect circle dashe(Left)

3.1.3 NURBS Surfaces

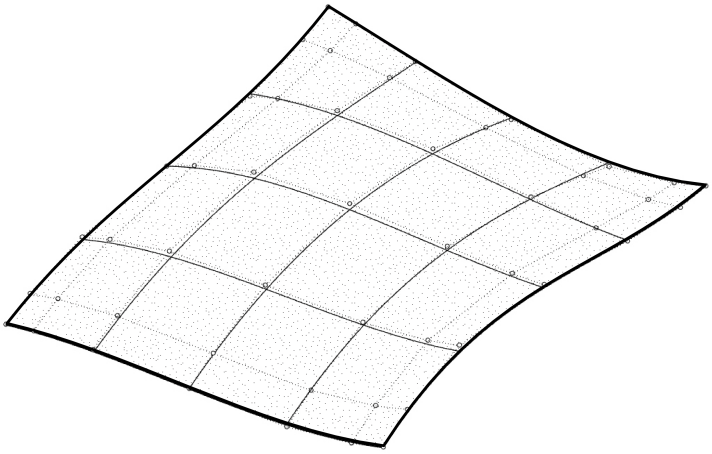


Figure 3.5: Example of a NURBS surface

The definition for a surface, which is the relevant geometry for this thesis, has the same logic as the curve, but extended to a grid. Each point is then associated with 2 basis functions, one in each direction. The defintion is as follows.

$$S(\xi_1, \xi_2) = \sum_{i=1}^{n} \sum_{i=1}^{n} N_{i,p}(\xi_1) M_{j,q}(\xi_2) \mathbf{P} i, j \tag{3.8}$$

where

- **S** is the surface
- **N** and **M** are the basis functions in the two directions
- i,j are the indexes of the basis functions.
- p,q are the interpolation degrees for the two directions.
- **P** is a point in the control point net.

It can be seen that the underlying base for any NURBS surface always is a 2 dimensional grid of points. However, the amount of points can vary in each direction and they can have independent interpolation degrees. A NURBS model is mainly composed of several surfaces, which will be referred to as patches. Within a patch the continuity is given as p-1, but as the endpoints are interpolated, the continuity of basis functions will not follow over patch intersections.

3.2 Isogeometric FE Formulation

Now the fundamentals of NURBS have been covered. The next step is to clarify how this logic is implemented in numerical analysis procedures. What follows is very brief and mainly serves to introduce the reader to some important concepts and a certain vocabulary. A deeper description can be found in [1]

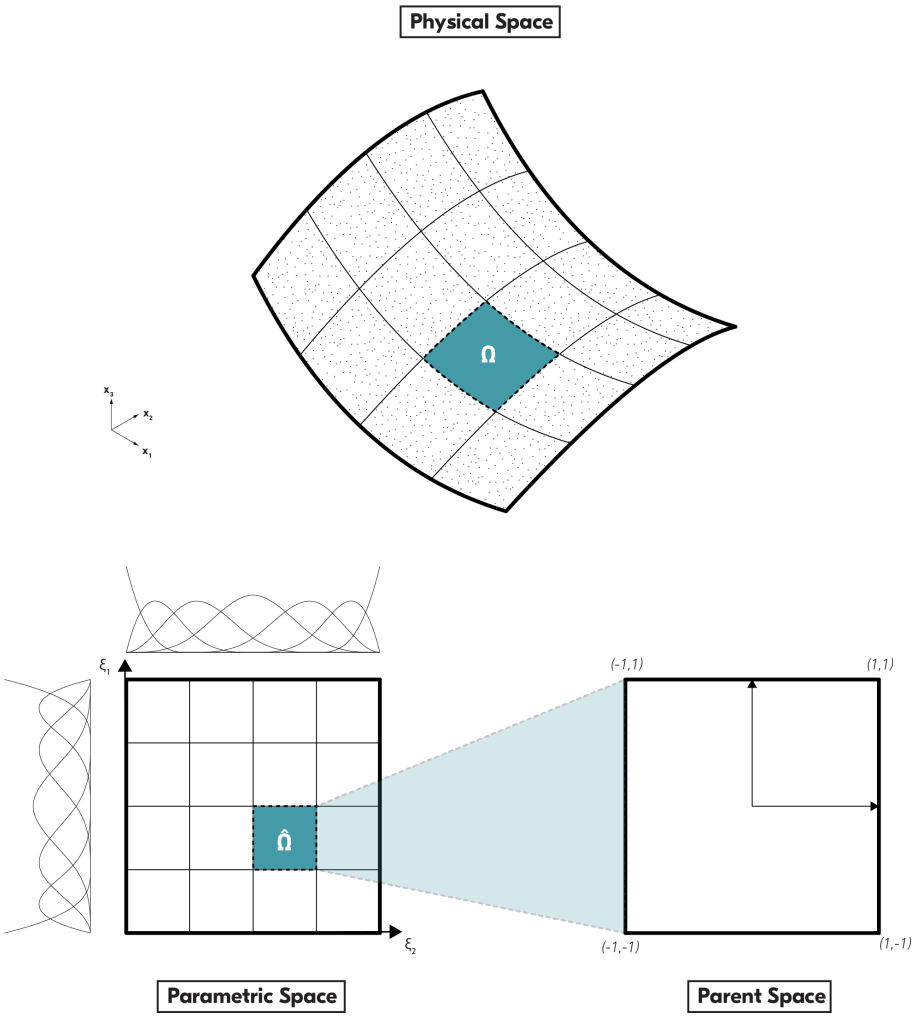


Figure 3.6: The different analysis spaces (Physical, parametric and parent)

3.2.1 IGA compared to FEA

Most of the process of IGA is similar to the procedures of FEM. The main difference is the geometry basis. The most important difference lies in the importance of the NURBS patch as opposed to the polygonal mesh most often used in standard FEA practise.

In FEA using meshes as geometry the basis functions are local to the element. A set of elements then compose a mesh describing the geometry of the object subject to analysis. By adapting the isoparametric concept, the basis functions are defined on a parent element, which is then mapped to the physical element. Integration is performed on the parent element. This gives two spaces, the physical and the parent. With IGA, and the notion of the patch, another space emerges, here labeled parametric space. An IGA patch is a system of elements, based on the knot vector, where each non-zero knot span is an element. The basis functions are instead defined on this combination of elements, the parameter space keep track of where on the patch each element is. This corresponds to the ξ_1, ξ_2 parameterisation of the patch. An additional mapping is then used from the element in parametric space to the normalised parent space of the element.

Figure 3.6 shows an example patch, consisting of four non-zero knotspans in each direction. The basis functions are shown besides it. I can be seen that the basis functions are continuous over element borders. However, this continuity is not true over element patches, as the endpoints are interpolated. Therefore, analysis of multi-patch systems are a complication for the application of IGA. This thesis will therefore only be concerned with single patch surfaces.

It should be noted that the definition is rotation free. This means that no rotational degrees of freedom are needed and curvatures are instead computed directly which is made possible through the continuity between the elements.
The use of IGA give an advantage in some concerns when dealing with some element types. A comparison between the regular basis functions used in FEA compared to IGA is shown in Figure 3.7. This show a one dimensional span (for example a bar element) consisting of four elements using quadratic interpolation polynomials. Each element has three points active on each element. However, as can be seen the FEA only has three local bases, defined for each element. The amount of nodes are 9 for the FEM mesh and 6 for the IGA patch. This graph shows the difference in continuity over the elements.

3.2.2 Relation to shell theory

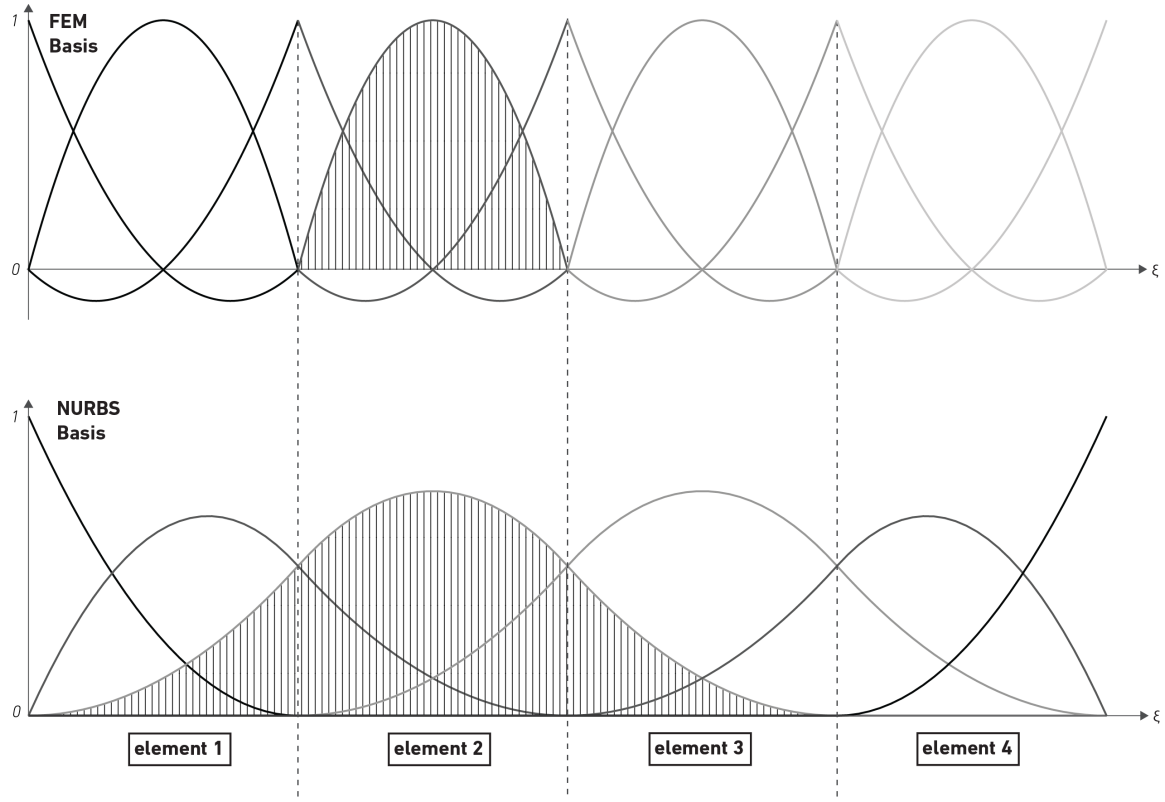


Figure 3.7: Example of element partitioning of Basis functions

When it comes to using NURBS function as a basis for shell analysis one aspect should be stated. The analysis of thin shells is preferably done with Kirchhoff Love theory, as will be expanded upon in the following chapter. This requires the derivatives of the basis functions.

However, given the basis of FEA as only c0 continuous the use of KL shell theory has been limited. It proved easier to use thick shell theory (Mindlin-Reissner) in combination with classical FE procedures. The application of NURBS enables higher continuity, which means that the derivatives necessary for KL theory are existing,

Kirchoff-Love shells are not suitable to FE discretization. Instead Mindlin Reissner theory became the main theory implemented in general FE packages. [7]

4

Nonlinear Formulations for Beams and Shells

As the aims of this thesis is to solve the form of deformed shell structures, a formulation for how this behaviour is understood and modeled is needed. This chapter will introduce the reader to the general theory of nonlinear structural mechanics that that is needed to solve the problem stated. It will not be concerned with the implementation aspects, but solely with the underlying principles and assumptions regarding the behaviour of thin shells and beams undergoing large displacements

We will go through the basics of geometrical non-linearity along with a brief review of relevant mathematical concepts. From that follows an introduction to the formulations used for the nonlinear behaviour along with an in depth derivation of a nonlinear beam in 2D space, followed by the same principles applied to a shell in 3D.

4.1 Nonlinear Analysis

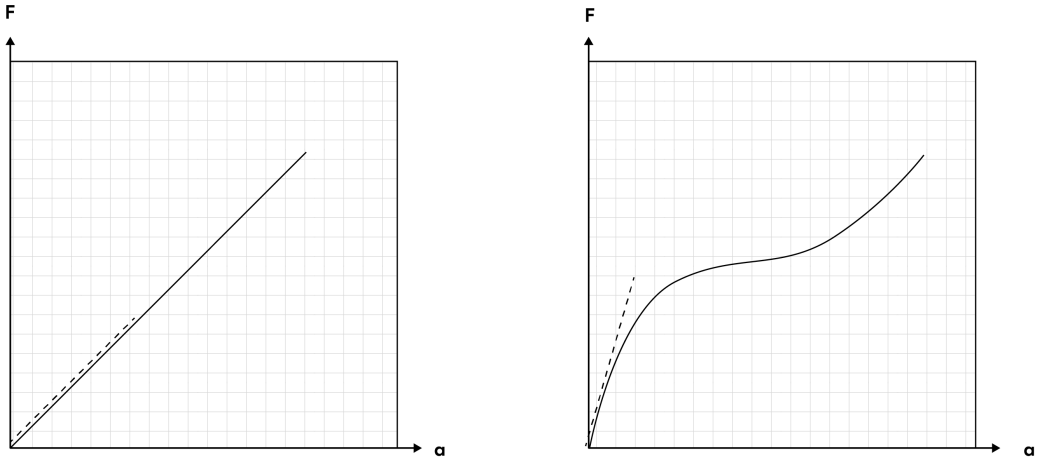


Figure 4.1: Example of a linear vs a nonlinear relationship between force *(F)* and displacement *(a)*

4.1.1 Preliminaries

Structural mechanics is, for the most part, concerned with finding the response of a structure to a certain load for some boundary conditions. In finite element methods the system of equations is most often concerned with the relation between the applied load and displacements through the stiffness. This relationship is often written as:

$$\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f}_{ext} \tag{4.1}$$

where K is the stiffness matrix, a is the displacement vector and F is the vector of external forces. This describes the behaviour pictured on the left in Figure 4.1.

The general assumption for most structures is that the displacements and strains are relatively small, meaning that this relationship is linear. However, if the displacements are large, making the structure reconfigure itself, or the material behaviour is nonlinear, the response of the structure will not be linear anymore, meaning that equation (4.1) is no longer enough.

The main difference is that the tangent stiffness is dependent on the displacements, as can be seen on the right in Figure 4.1. This leads to a system were the solution is dependent on the solution.

$$\boldsymbol{f}_{int}(\boldsymbol{a}) = \boldsymbol{f}_{ext} \tag{4.2}$$

As the system behaviour changes with displacements, the solution becomes more complicated than a linear analysis. The problem description now needs to account for the equilibrium of the deformed body as well, which is neglected in the linear formulation.

As mentioned, there are several potential sources of nonlinearities in a system, and the solution approach will depend on the source. However, in general, one can split the sources of nonlinearity in to two categories, which are:

- *Geometric nonlinearity*
- *Material (or physical) nonlinearity*

With geomertic nonlinearity one refers to the fact that the geometric configuration will be change with the solution. In geometric linearity, the deformed configuration is assumed to be the same as the undeformed. With a geometrically nonlinear case this is no longer true, thus one needs to separate the formulation into an intial configuration and a deformed, and ensuring equilibrium on the deformed geometry.

Material nonlinearity instead occurs on the material level and describe a nonlinearity in the stress-strain relations of a material, (e.g plasticity).

Bathe gives an overview of some nonlinear formulations and their definition, which are given in the table below.

Classification of Nonlinear analyses

TYPE OF ANALYSIS	DESCRIPTION	TYPICAL FORMULATION USED	STRESS AND STRAIN MEASURES
Materially Nonlinear Only	Infinitesimnal displacements and strains, the stress-strain relationships are nonlinear	Materially Nonlinear Only (MNO)	Engineering stress Engineering strain
Large Displacements Large Rotations Small Strains	Displacements and rotations are large, but fiber extensions and angle changes between fibres are small, the stress strain relation may be either linear or nonlinear	Total Lagrangian (TL)	Second Piola Kirchhoff stress Green Lagrange strain
		Updated Lagrangian (UL)	Cauchy stress Alamansi Strain
Large Displacements Large Rotations Large Strains	Fiber extensions and angle changes between fibres are large, fiber displacements and rotations may also be large, the stress-strain relation may be either linear or non-linear.	Total Lagrangian (TL)	2nd Piola Kirchhoff Stress Green-Lagrange Strain
		Updated Lagrangian (UL)	Cauchy Stress Logarithmic Strain

Figure 4.2: Classification of Nonlinear analyses redrawn after Bathe [25]

TIME AND TIMESTEPS
As the properties of the system changes, the solution of a nonlinear system needs to trace the behaviour as a sequence. To describe subsequent steps in the process a variable t representing time is often introduced, and will so be in the derivations of this thesis. However, this time is not to be confused with actual time as the solutions of nonlinear static problems neglect inertia behaviour and assumes a static response at all stages. The structure does therefore not move but the time is simply a pseudo time used to denote the different stages of nonlinear analysis. The displacements are solved in the step from time t to $t + \Delta t$.

In the derivations three generalized time steps will be used:

time 0 Refers to the initial step and reference configuration.

time t Refers to a known step at any time.

time t + Δt Refers to the next and unknown step.

In this thesis, both a static and a pseudo dynamic approach will be studied. For reference see chapter 5, Nonlinear solutions techniques.

4.1.2 Geometrically Nonlinear Analysis

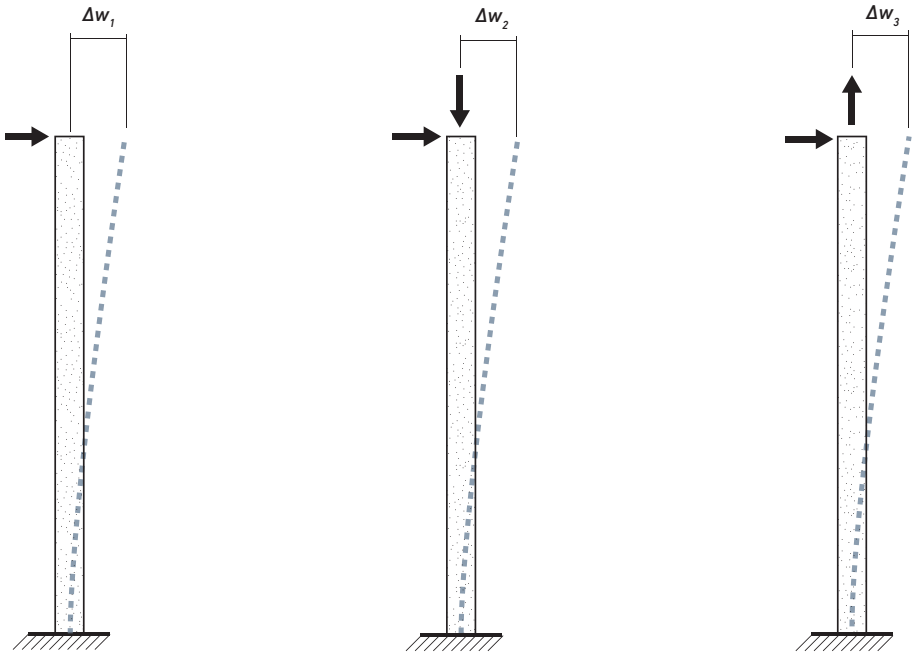


Figure 4.3: P-Δ effect - Example of a geometrically nonlinear problem

This thesis will only be concerned with geometrical non-linearities, assuming large displacements but a linear stress strain relationship. A distinction should be made between large strains and large displacements, which are not necessarily connected. A material can undergo large deformation while still only having small length changes.

As previously mentioned, geometrical nonlinearity means accounting for the equilibrium on the deformed geometry. This is often split in two categories, 2nd or 3rd order effects. In 2nd order analysis, the equilibrium is considered on the deformed geometry, however, it's still restricted to small displacements. An example of this is the $P - \Delta$ effect illustrated in Figure 4.3. When equilibrium is considered on the undeformed geometry, the compressive or tensile load will not affect the horisontal displacement as they are orthogonal. However, considering the equilibrium on the deformed geometry will result in a contribution of the vertical load making

$$\Delta w_3 < \Delta w_1 < \Delta w_2$$

As this study will analyse significant changes in geometry, a complete analysis is needed. The following chapters will provide the formulations. A key feature is to make sure to properly account for rigid body modes as they should not add any strain energy to the system.

4.1.3 Nonlinear Formulation

The following chapter will provide the general formulations required to describe a body undergoing arbitrarily large displacements. It will first describe the formulation of the equilibrium using the virtual work and the relevant measures needed to accurately trace the equilibrium and establish a framework for the nonlinear elements.

4.1.3.1 Virtual Work

The first step is to formulate the conditions for equilibrium. Compared to linear solutions which find a deformation based on the input geometry the nonlinear formulation need to be valid at any point in the analysis. It will therefore be formulated as an arbitrary step from a time t to an unknown step at $t + \Delta t$ as mentioned earlier.

To achieve this, the principle of virtual work, or more specifically, of virtual displacements will be used. [27] [23] This states that the difference between the external and internal work on a structure should be the zero under an arbitrary virtual displacement δu.

$$\delta W = W_{ext} - W_{int} = 0 \tag{4.3}$$

$$\delta W = \frac{\partial W}{\partial u} \Delta u = 0 \tag{4.4}$$

By linearizing this equation the following expression can be obtained

$$\frac{\partial W}{\partial u_r} + \frac{\partial^2 W}{\partial u_r \partial u_s} \Delta u = 0 \tag{4.5}$$

The first derivative of the work w.r.t a displacement variable gives the resultant force vector $\mathbf{R}$, which is given as:

$$R_r = \frac{\partial W_{ext}}{\partial u_r} - \frac{\partial W_{int}}{\partial u_r} = (\mathbf{f}_{ext} - \mathbf{f}_{int})_r \tag{4.6}$$

Further, the tangent stiffness $\mathbf{K}_{rs}$ is given by the second derivative of equation (4.3). In the context of this study, the load is assumed to be displacement independent, which means that the effect from the external load is constant w.r.t to the displacement. The stiffness then becomes:

$$\mathbf{K}_{rs} = \frac{\partial^2 W_{int}}{\partial u_r \partial u_r} \tag{4.7}$$

Putting this back into (4.5) the nonlinear system can written as:

$$\mathbf{K}^t \Delta u = \mathbf{f}_{ext}^{t+\Delta t} - \mathbf{f}_{int}^{t} \tag{4.8}$$

The nonlinear equilibrium is now linearised according to the Newton-Raphson method. A step using the tangent stiffness matrix can here be seen to correspond to a change or an out of balance for $\Delta \mathbf{f} = \mathbf{f}_{int} + \mathbf{f}_{ext}$ *(if no difference between internal and external forces, the system is in equilibrium and* $\Delta u = 0$*).*

4.1.3.2 Configurations, Stress and Mapping

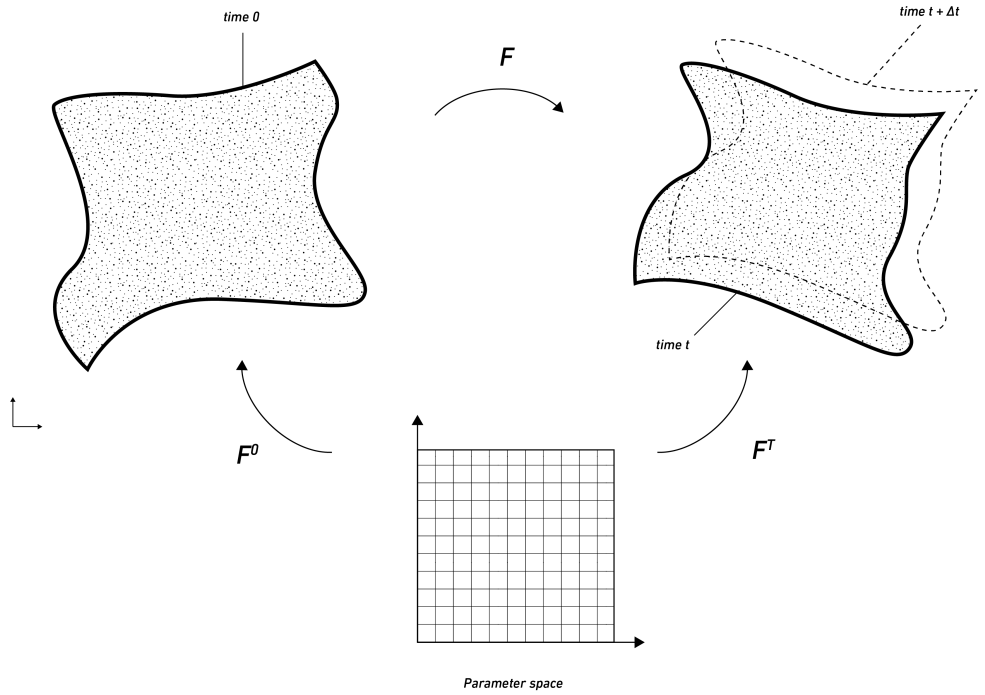


Figure 4.4: Reference Bodies and corresponding mappings

The following segment will provide a glossary of configurations and mapping required to keep track of the relationships and set up the framework for how the behaviour is traced.

From the previous parts of the thesis we have established a set of spaces or configurations. More specifically the physical and parameter spaces have been explained in Ch. [3.1], and subsequently the different configurations for the nonlinear formulation (0,t and $t + \Delta t$) have been defined. These are depicted in Figure 4.4.

Further a way to relate these different domains to each other is necessary. Therefore a set of mappings are introduced. These are $\boldsymbol{\varphi}_0(\xi_1, \xi_2)$, $\boldsymbol{\varphi}_t(\xi_1, \xi_2)$ and $\mathbf{F}$. $\boldsymbol{\varphi}_0$ and $\boldsymbol{\varphi}_t$ map a point in the parametric space to the reference and current configuration respectively. $\mathbf{F}$ denotes the deformation gradient and is used to map from deformed to undeformed condition. The deformation gradient relates the length of a material fiber in the reference configuration to the same in the current configuration.

The deformation gradient is defined as follows.

$$\mathbf{F} = \frac{\partial \boldsymbol{x}_i^t}{\partial \boldsymbol{x}_j^0} = \frac{\partial \boldsymbol{\varphi}_t}{\partial \xi_i} \cdot \left[\frac{\partial \boldsymbol{\varphi}_0}{\partial \xi_j} \right]^{-1} \tag{4.9}$$

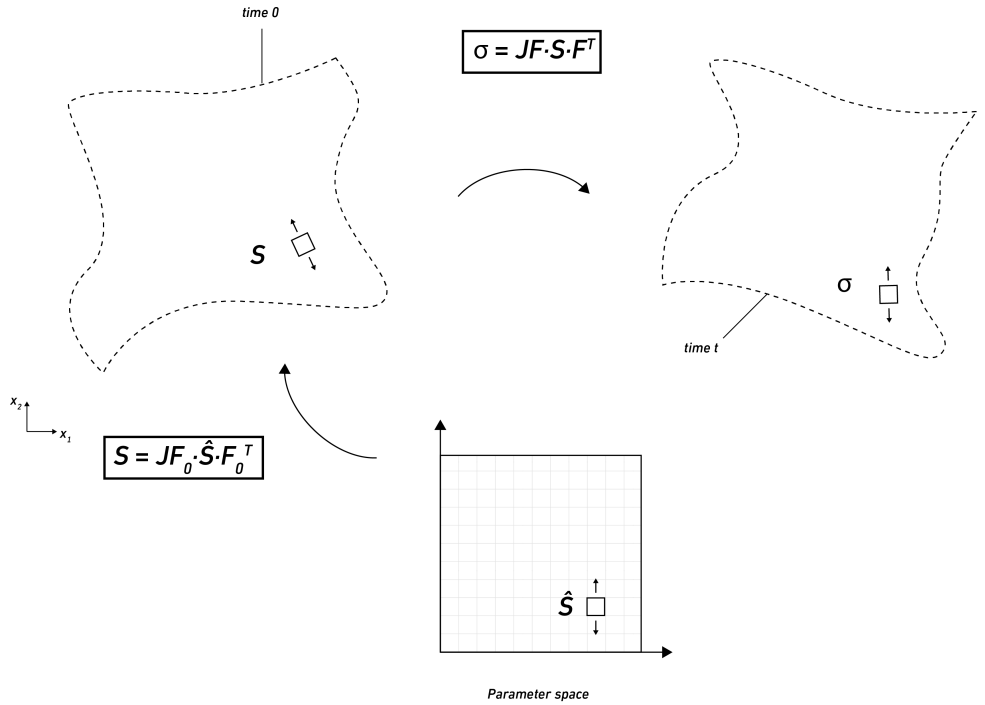


Figure 4.5: Stress measures and their point of action and corresponding mappings

As can be seen in the table of Bathe, there exist a range of different stress measures. For our case (Large displacements, small strains) two stress measures are relevant, namely, the 2nd Piola Kirchhoff (PK2) stress and the Cauchy stress. The difference between these stress measures is crucial to the formulation of the nonlinear behaviour. In Figure 4.5 three different stresses are depicted. Where S is the PK2 stress, σ is the Cauchy stress and $\hat{S}$ is the PK2 stress expressed with reference to the parameter space.

The cauchy stress is often referred to as the *true stress tensor*. [reference]. This means that this tensor represents the state in the actual configuration. It relates the forces in the current configuration to the areas in the current configuration.

The 2nd Piola-Kirchhoff stress tensor is the projection of the cauchy stress back onto the reference configuration through the deformation gradient. It is a stress that lacks any direct physical interpretation but plays a key role in nonlinear computaion.
The 2nd Piola Kirchhoff stress is related to the cauchy stress by the mapping [26]:

$$\mathbf{S} = J^{-1}\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-T} \tag{4.10}$$

where

- $\boldsymbol{\sigma}$ is the Cauchy Stress tensor
- $\mathbf{S}$ is the 2nd Piola Kirchoff Stress tensor
- $\mathbf{F}$ is the deformation gradient between deformed and undeformed configuration.
- J = det($\mathbf{F}$) defines the relative volume between the configurations.

From mass continuity J also gives the density relation between the deformed and undeformed configurations

$$J = \frac{\rho_t}{\rho_0} \tag{4.11}$$

4.1.3.3 Lagrangian Formulations

One of the key differences between linear and nonlinear is the cause of the nonlinear effects which requires the application of the appropriate stress and strain measures. [25]

When dealing with large displacements one would prefer to work in the material coordinates, as opposed to describing stress and strain in the global coordinate system. One of the key properties should be that the deformation is not producing any strain energy under rigid body displacements, but should be completely invariant to the position in space.

As can be seen in table, two different formulations are commonly used when dealing with large displacements, namely *Total Lagrangian* (TL) or *Updated Lagrangian* (UL).

The main difference between the two lies in where the stresses are considered. In the updated Lagrangian the stresses are computed in the current configuration, resulting in the use of the *true stress* of the cauchy stress-tensor. Integrations are therfore performed on the deformed body at time *t*. In the total Lagrangian formulation, however, the stresses are instead *pulled back* to the reference configuration and the integration is performed on the reference body, leading to the use of the PK2 stress measure. Here a brief review of the relevant measures will be considered, for a more extensive review please see [25].

Considering the linear equation derived in the previous section, the equilibrium equation is given in the Total Lagrangian description as:

$$\Delta \mathbf{f}^{t+\Delta t} = \mathbf{f}_{ext}^{t+\Delta t} - \int_{\Omega_0} (\mathbf{S}^{t+\Delta t} : \delta \mathbf{E}) d\Omega_0 \tag{4.12}$$

where

- $\mathbf{S}$ is the 2nd Piola Kirchoff Stress Tensor
- $\mathbf{E}$ is the Green Lagrange strain tensor
- $\mathbf{\Omega}_0$ is the body in the reference configuration

And in the Updated Lagrangian description as:

$$\Delta \mathbf{f}^{t+\Delta t} = \mathbf{f}_{ext}^{t+\Delta t} - \int_{\Omega_{t+\Delta t}} (\boldsymbol{\sigma}^{t+\Delta t} : \delta \boldsymbol{e}) d\Omega_{t+\Delta t} \tag{4.13}$$

where

- $\boldsymbol{\sigma}$ is the Cauchy Stress tensor
- $\mathbf{e}$ is the Alamansi strain tensor
- $\mathbf{\Omega}_{t+\Delta t}$ is the body in the current configuration

As mentioned the PK-2 stress lacks physical interpretation, but it is the energetically conjugate tensor to the Green-Lagrange strain. The 2nd Piola Kirchhoff stress is derived by the differentiation of the strain energy w.r.t the Green-Lagrange strain-tensor. [23].

$$\mathbf{S} = \frac{\partial W_{int}}{\partial \boldsymbol{E}} \tag{4.14}$$

By using the total lagrangian formulation, the integration can then be performed on the reference geometry, which will be done in the present work.

4.2 Brief Review of Differential Geometry

When dealing with NURBS as a geometry basis, one deals with the representation of parameterized smooth objects. The field of differential geometry is concerned with the study of curved objects in space. It will therefore provide some concepts and tools needed in order to be able to compute the necessary quantities, regarding stresses and strains in smooth continuous bodies. The chapter follows from what is presented in [8] in the Appendix on the same topic by C.Williams.
Further more it proves interesting insights regarding the plane isometries presented in Chapter [2.3]

4.2.1 Co- and Contravariant Base Vectors

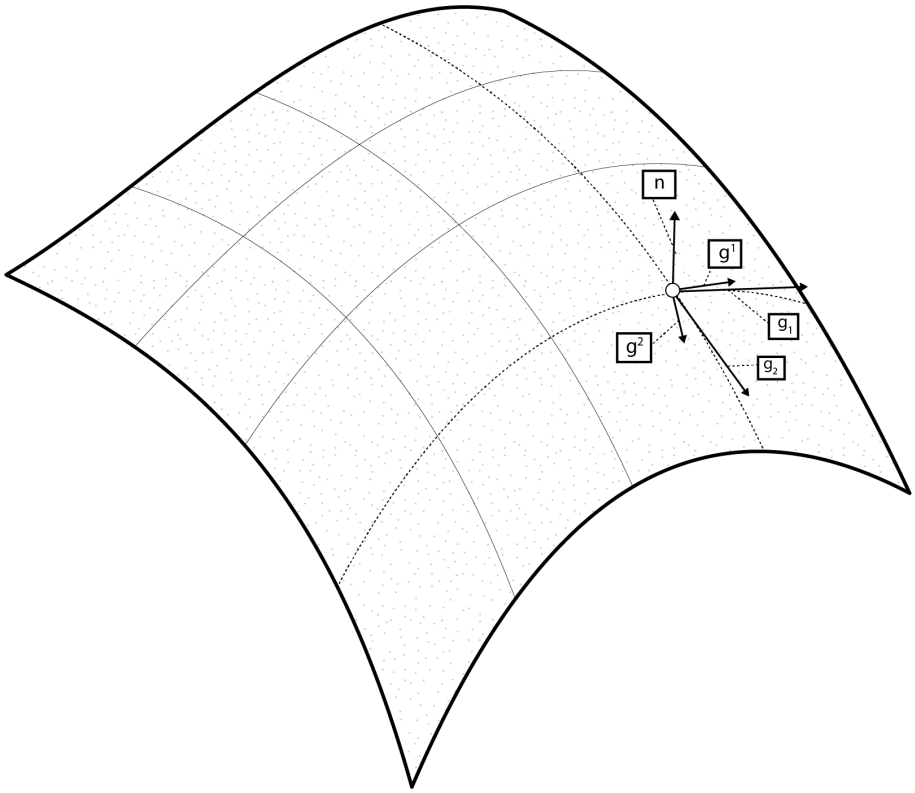


Figure 4.6: Geometry of an arbitrary shell body

As a first step a general parameterisation of a curved body in three dimensional space is needed.
The body of a surface can be parameterized as:

$$\mathbf{X}(\theta_1, \theta_2) = x(\theta_1, \theta_2)\mathbf{e}_1 + y(\theta_1, \theta_2)\mathbf{e}_2 + z(\theta_1, \theta_2)\mathbf{e}_3 \tag{4.15}$$

where

- $\mathbf{X}$ denotes a point on the body of the surface
- $\mathbf{e}_i$ are the euclidian base vectors [i = 1,2,3]
- θ_1, θ_2 are the surface parameters

When dealing with quantities related to a curvilinear coordinate system, a set of vectors are necessary to treat these quantities. For the theories presented will refer to two sets of vectors needed to deal with this type of geometry.
In figure a set of vectors are drawn. These show the covariant base vectors $[g_1, g_2]$ along with the contravariant base vectors $[g^1, g^2]$ and the normal vector $[n]$
The *covariant base vectors* are given by the differentiation of the parameterization w.r.t a parameterization variable

$$\mathbf{g}_i = \frac{\partial \mathbf{X}}{\partial \theta_i} \tag{4.16}$$

with the normal vector $\boldsymbol{n}$ given by:

$$\mathbf{n} = \frac{\mathbf{g}_1 \times \mathbf{g}_2}{|\mathbf{g}_1 \times \mathbf{g}_2|} \tag{4.17}$$

The covariant base vectors can be used to define a local cartesian basis at an arbitrary point on the shell body. This is given as: (the hat refers to a local coordinate system)

$$\hat{\boldsymbol{e}}_1 = \frac{\mathbf{g}_1}{|\mathbf{g}_1|}$$

$$\hat{\boldsymbol{e}}_2 = \frac{\mathbf{g}_2 - (\mathbf{g}_2 \cdot \mathbf{g}_1)\mathbf{g}_1}{||\mathbf{g}_2 - (\mathbf{g}_2 \cdot \mathbf{g}_1)\mathbf{g}_1||}$$

$$\hat{\boldsymbol{e}}_3 = \mathbf{n}$$

However, as the base vectors g_i are curvilinear, and therefore are not necessarily perpendicular, another set of base vectors are useful, namely, the contravariant base vectors g^i. The essential property of the contravariant vectors are that they are perpendicular to their respective contravariant vector. Further they also lie in the tangent plane of g_i and are perpendicular to n.
They therefore possess the following relationship:

$$\mathbf{g}_i \cdot \mathbf{g}^j = \delta_{ij} \tag{4.18}$$

$$\mathbf{g}_i \cdot \mathbf{n} = \mathbf{g}^i \cdot \mathbf{n} = 0 \tag{4.19}$$

The magnitude of the contravariant vectors are coupled to that of the covariant vectors through (4.18). The reason for the need of these vectors lies in the fact that when computing dot products between curvilinear base vectors creates spurious off diagonal terms, which is circumvented by using a combination of co and contravariant bases.

4.2.2 First and Second Fundamental Form of Surfaces

The basevectors can be used to compute length and curvature measurements on a parameterized surface. This is done through what is referred to as the metric tensor (often denoted in covariant basis as g_{ij}) which is defined by the corresponding basevectors as:

$$g_{ij} = \mathbf{g}_i \cdot \mathbf{g}_j \tag{4.20}$$

This equation is referred to as the *first fundamental form of surfaces* and it gives us the possibility to compute lengths and changes in lengths on a continuous surface.

Further, the contravariant components of said tensor can be found using the contravariant base vectors as:

$$g^{ij} = \mathbf{g}^i \cdot \mathbf{g}^j \tag{4.21}$$

the co and contravariant descriptions of this tensor is related by:

$$[g^{ij}] = [g_{ij}]^{-1} \tag{4.22}$$

In a similar way, the *second fundamental form of surfaces* is used to describe the curvature and subsequent changes in curvature on a surface.
It is defined as covariant coefficients:

$$b_{ij} = \mathbf{g}_{i'j} \cdot \mathbf{n} \tag{4.23}$$

or as contravariant:

$$b^{ij} = \mathbf{g}^i_{\ j} \cdot \mathbf{n} \tag{4.24}$$

For a proof of these equations see [8] or [4].

4.2.3 Basis Changes

Further on a need to map between different bases will be necessary to compute strain energies on shell bodies.
The first and second fundamental form of surfaces give relationships that relate to lengths and curvature on a continuous body, and are necessary for the definition of strain concepts.
Given two different bases we now need to switch a set of coefficients from one to the other:

$$\alpha_{ij}\mathbf{a}^i \otimes \mathbf{a}^j = b_{ij}\mathbf{b}^i \otimes \mathbf{b}^j \tag{4.25}$$

Kiendl [27] presents the following transformation rules between co and contravariant bases.

$$b_{kl} = \alpha_{ij}(\mathbf{b}_k \cdot \mathbf{a}^i)(\mathbf{a}^j \cdot \mathbf{b}_l) \tag{4.26}$$

$$b^{kl} = \alpha_{ij}(\mathbf{b}^k \cdot \mathbf{a}^i)(\mathbf{a}^j \cdot \mathbf{b}^l) \tag{4.27}$$

$$b_{kl} = \alpha^{ij}(\mathbf{b}_k \cdot \mathbf{a}_i)(\mathbf{a}_j \cdot \mathbf{b}_l) \tag{4.28}$$

$$b^{kl} = \alpha^{ij}(\mathbf{b}^k \cdot \mathbf{a}_i)(\mathbf{a}_j \cdot \mathbf{b}^l) \tag{4.29}$$

4.3 Nonlinear Element formulations

The main element this study is concerned with is the shell element. However, as an intermediate step a reduced shell model will be derived in 2D. This will be equivalent of a beam in 2D space, but it's logic will be similar. As the beam is considered an elaborate pre-step to the shell model, it will be derived using the necessary methods of the shell continuum. Therefore, there might exist smarter, simpler or more suitable ways to derive the beam model, however as the beam is not the end goal will will here treat it as a continuum model condensed into a 1D element.
All elements will be derived using the Euler-Bernoulli (or Kirchhoff-Love) hypothesis that couples the orientation of the section to be perpendicular to the center line, thus neglecting any shear deformations.

4.3.1 A Geometrically Exact Beam Formulation

The following chapter derives the necessary relationship for the analysis of a nonlinear smooth parameterized beam.

4.3.1.1 Beam Geometry

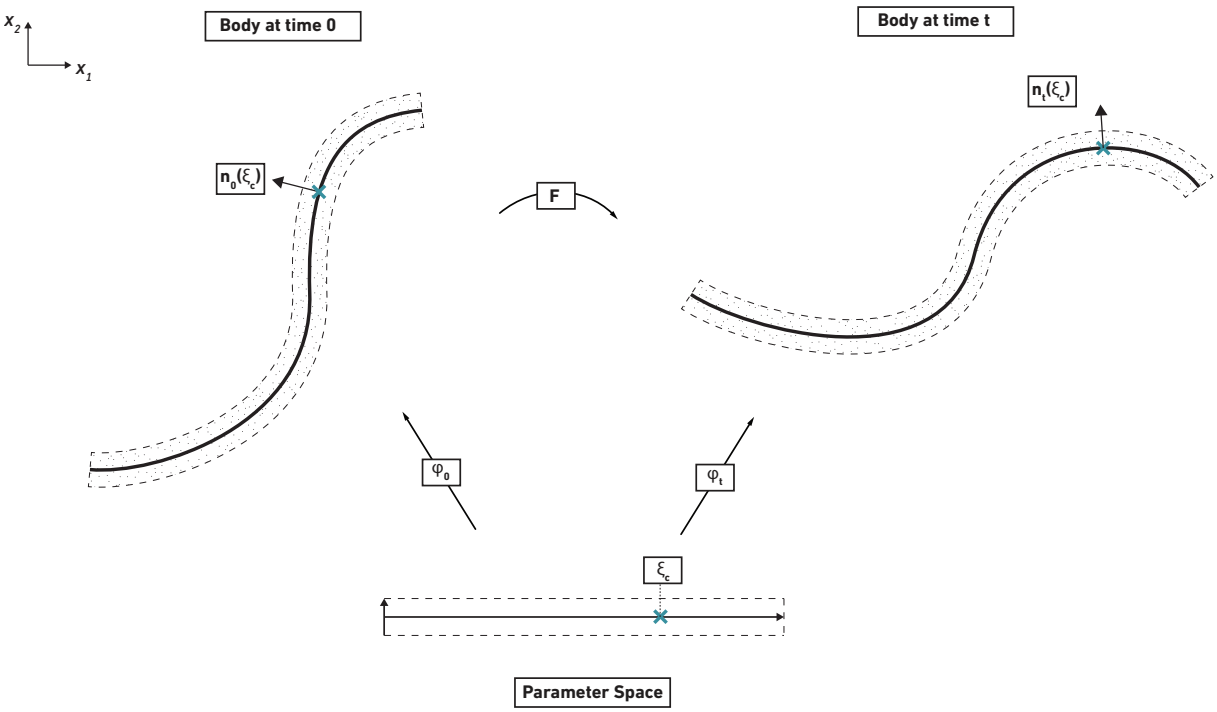


Figure 4.7: Beam Geometry and definitions

A point $\varphi(\xi_1)$ on an arbitrarily shaped beam body Ω can be parameterized as,

$$\boldsymbol{X}(\xi_1, \xi_3) = \tilde{\boldsymbol{\varphi}}(\xi_1) + \xi_3 h \mathbf{n}(\xi_1) \tag{4.30}$$

where,

$\tilde{\boldsymbol{\varphi}}$ denotes the position of the center line.
$\boldsymbol{n}(\xi_1)$ is given by

$$\frac{e_3 \times \tilde{\boldsymbol{\varphi}}'_{\xi_1}}{|\tilde{\boldsymbol{\varphi}}'_{\xi_1}|}$$

4.3.1.2 Large Displacement Kinematics

Given the description of the geometry, the kinematics describe the changes in the geometric configuration. The following derivation will use the total Lagrangian formulation meaning that the configuration is always referred to it's initial state at t_0, as opposed to the updated Lagrangian formulation where the previous step is used as a reference for the internal forces. Through this, the integration will be preformed on the initial geometry assuming that lengths and cross section properties never change A^0 "is approximately" A^t

Consider the two configurations in Figure 4.7 where $\varphi^0(\xi_1)$ denotes the reference configuration and $\varphi^t(\xi_1)$ denotes the geometry at an arbitrary time **t**

The displacement vectors at time t is given by,

$$\mathbf{u}^t = \boldsymbol{\varphi}^t - \boldsymbol{\varphi}^0 \tag{4.31}$$

The deformation gradient gives the mapping between deformed and undeformed configuration and can be described as.

$$\mathbf{F} = \frac{\partial \boldsymbol{x}^t}{\partial \boldsymbol{x}^0} = \frac{\partial \boldsymbol{\varphi}^t}{\partial \boldsymbol{\xi}} \cdot \left[\frac{\partial \boldsymbol{\varphi}^0}{\partial \boldsymbol{\xi}}\right]^{-1} \tag{4.32}$$

This corresponds to the ratio between a material fiber in the reference configuration compared to the updated one.

In large displacement kinematics, one must make sure that the rigid body movements does not affect the energy in the system, but only the actual length changes, regardless of position. There exists a multitude of strain measures, and the one chosen in this study is the Green Lagrange strain tensor. That tensor is given by the deformation gradient through,

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I}) \tag{4.33}$$

The strain measure can be rewritten in the following way,

$$\hat{E} = \mathbf{F^0}^T\mathbf{E}\mathbf{F}^0 = \frac{1}{2}(\mathbf{F}^{tT}\mathbf{F}^t - \mathbf{F^0}^T\mathbf{F}^0) \tag{4.34}$$

The strain measure $\hat{E}$ is the strain mapped onto the parametric space. It will therefore not relate to the euclidean units. See Figure 4.5. $\mathbf{F^t}$ representes the lengths at time **t** and $\mathbf{F^0}$ represents the lengths at time **0**. These are given by tensors in the following way,

$$\mathbf{F}^t = \frac{\partial \boldsymbol{\varphi}^t}{\partial \boldsymbol{\xi}} = \frac{\partial \boldsymbol{\varphi}^t}{\partial \xi_1} \otimes \boldsymbol{e}_1^\xi + \frac{\partial \boldsymbol{\varphi}^t}{\partial \xi_3} \otimes \boldsymbol{e}_3^\xi \tag{4.35}$$

This describes a two dimensional strain state in the beam. However, as this study is mainly interested in the study of very slender profiles, the only term to be considered will be F_{11} which corresponds to the longitudinal strain of the beam, meaning the strain perpendicular to the center line together with the coupling terms will not be considered .

The term F_{11} of F can be expanded into,

$$\mathbf{F}_{11}^t = \tilde{\boldsymbol{\varphi}}'^t_{\xi_1} + \xi_3 h \boldsymbol{n}^t \tag{4.36}$$

Giving

$$(\mathbf{F}^{tT}\mathbf{F}^t)_{11} = \tilde{\boldsymbol{\varphi}}'^t_{\xi_1} \cdot \tilde{\boldsymbol{\varphi}}'^t_{\xi_1} + 2\xi_3 h \boldsymbol{n}'^t_{\xi_1} \cdot \tilde{\boldsymbol{\varphi}}'^t_{\xi_1} + \xi_3^2 h^2 \boldsymbol{n}'^t_{\xi_1} \cdot \boldsymbol{n}'^t_{\xi_1} \tag{4.37}$$

For slender beams, as are considered in this thesis, the last quadratic term can be neglected as $\xi_3^2 h^2 \boldsymbol{n}^t << \tilde{\boldsymbol{\varphi}}^t_{\prime\xi_1}$ [reference]

The strain term can then be formulated as

$$\hat{E}_{11} = \frac{1}{2}(\boldsymbol{\varphi}^t_{\prime\xi_1} \cdot \boldsymbol{\varphi}^t_{\prime\xi_1} - \boldsymbol{\varphi}^0_{\xi_1} \cdot \boldsymbol{\varphi}^0_{\xi_1}) + \frac{\xi_3 h}{a_\xi}(\boldsymbol{\varphi}^0_{\prime\xi_1\xi_1} \cdot \boldsymbol{n}^0 - \boldsymbol{\varphi}^t_{\prime\xi_1\xi_1} \cdot \boldsymbol{n}^t) \tag{4.38}$$

The strain term described above is not related to normalized standard units but are described in parametric space.
To map these to a local, orthogonal basis expressed in standard units (m) we need to find E so that:

$$E_{ij}\tilde{\boldsymbol{e}}^i_0 \otimes \tilde{\boldsymbol{e}}^j_0 = \hat{E}_{ij}\mathbf{a}^i_0 \otimes \mathbf{a}^j_0 \tag{4.39}$$

where:
- E and $\hat{E}$ are the strains in normalized and non-normalized units.
- $\tilde{\boldsymbol{e}}_{\boldsymbol{i}}$ denotes a local cartesian basis
- $\mathbf{a}_i$ denotes the base vectors $\tilde{\boldsymbol{\varphi}}_{\prime\xi_i}$
- *i = {1,3}*

With the local cartesian defined as in equation [ref]. For this the equations presented in [ref] are used, with the local cartesian defined as in equation [ref]. As both bases are orthogonal and we only are concerned with the first component, the equation reduces to:

$$E_{11} = \hat{E}_{11}(\tilde{\boldsymbol{e}}^0_1 \cdot \boldsymbol{a}^1_0)(\boldsymbol{a}^1_0 \cdot \tilde{\boldsymbol{e}}^0_1) \tag{4.40}$$

or:

$$E_{11} = \hat{E}_{11}\frac{1}{\boldsymbol{a}^0_1 \cdot \boldsymbol{a}^0_1} \tag{4.41}$$

The mapping refers the length in the deformed configuration to the length of a material fiber in the initial configuration. For an initially straight beam, the lengths are constant through the thickness of the beam. However, if the beam is intially curved this is no longer the case. The effect of this has shown to be minimal [reference] and for this study, this initial variation is neglected and the whole section is assumed to have the same initial length (which is the length at the centerline.)

By adopting the notation for the base vectors as $\boldsymbol{a}_{\xi_1}$ and $\boldsymbol{a}_3$, the metric coefficients can be described as follows:

In the reference configuration

$$A_\xi = \boldsymbol{a}^0_{\xi_1} \cdot \boldsymbol{a}^0_{\xi_1}$$
$$K_\xi = \boldsymbol{a}^0_{\xi'_1\xi_1} \cdot \boldsymbol{a}^0_3$$

and in the deformed configuration

$$a_\xi = \boldsymbol{a}^t_{\xi_1} \cdot \boldsymbol{a}^t_{\xi_1}$$
$$\kappa_\xi = \boldsymbol{a}^t_{\xi'_1\xi_1} \cdot \boldsymbol{a}^t_3$$

we get the final expression for the strain as,

$$E_{11} = \frac{1}{2\tilde{A}_\xi}(a_\xi - A_\xi) + \frac{\xi_3}{\tilde{A}_\xi}(K_\xi - k_\xi) \tag{4.42}$$

It can be observed the the strain is split into one constant part, which represents the strain of the middle surface, and one linear part which is related to the change in curvature resulting from bending.

$$\epsilon_{11} = \epsilon_{\xi_1} + \xi_3\kappa_\xi \tag{4.43}$$

4.3.1.3 Constitutive Relations

For the stress-strain relationship the previously derived Green-Lagrange strain measure will be used in combination with the 2nd Piola-Kirchhoff stress measure.
They relate in the following way:

$$\boldsymbol{S} = \boldsymbol{C} : \boldsymbol{E} \tag{4.44}$$

In the context of this study, the strains will be assumed to be rather small, as no significantly large strain should be applied to the sheets as they are formed. For this purpose, the stress-strain relationship adapted is the one of St. Venant which assumes a linear relationship between stress and strain, thus, the Hooke tensor will be used.

The three dimensional body of the beam will be reduced into a one dimensional stress state. It will therefore be assumed that the stresses S_{22} and S_{33} will be zero.
The stresses in the beam can then be written on voight notation as:

$$\mathbf{S}^{ij} = \begin{bmatrix} S^{11} \\ S^{22} \\ S^{33} \end{bmatrix} = \begin{bmatrix} S^{11} \\ 0 \\ 0 \end{bmatrix} \tag{4.45}$$

The stress in the 1st direction is then computed by:

$$S^{11} = \sum_{ij} \mathbf{C}^{11ij} \mathbf{E}_{ij} \tag{4.46}$$

Given our one dimensional body, only one stress and one strain measure exists. The stress is given by condensing the constitutive relationship:

$$\begin{bmatrix} S^{11} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} C^{1111} & C^{1122} & C^{1133} \\ C^{2211} & C^{2222} & C^{2233} \\ C^{3311} & C^{3322} & C^{3333} \end{bmatrix} \begin{bmatrix} E_{11} \\ E_{22} \\ E_{33} \end{bmatrix} \tag{4.47}$$

by first finding the unknown strains E_{22} and E_{33} as:

$$\begin{bmatrix} E_{22} \\ E_{33} \end{bmatrix} = \begin{bmatrix} C^{2222} & C^{2233} \\ C^{3322} & C^{3333} \end{bmatrix}^{-1} \begin{bmatrix} C^{2211} \\ C^{3311} \end{bmatrix} E_{11} \tag{4.48}$$

and subsequently finding the condensed relationship for the first stress as:

$$S^{11} = \left[C^{1111} - \begin{bmatrix} C^{1122} & C^{1133} \end{bmatrix} \begin{bmatrix} C^{2222} & C^{2233} \\ C^{3322} & C^{3333} \end{bmatrix}^{-1} \begin{bmatrix} C^{2211} \\ C^{3311} \end{bmatrix} \right] E_{11} \tag{4.49}$$

However, this expression reduces to the commonly known Young's modulus, giving:

$$S^{11} = E_{young} E_{11} \tag{4.50}$$

However, the strain is dependent on the ξ_3 parameter. The stress-strain relations can be expanded into the two terms present in equation.
Decomposed into bending and axial force, this becomes:

$$\boldsymbol{S} = \begin{bmatrix} \sigma_n \\ \sigma_b \end{bmatrix} = E_{young} \begin{bmatrix} \epsilon_\xi \\ \xi_3 \kappa_\xi \end{bmatrix} = E_{young} \boldsymbol{E} \tag{4.51}$$

4.3.1.4 Internal Work

From chapter [4.1.3] and equation (4.3) the equilibrium conditions are given. The definition of the internal work is given as:

$$\delta W_{int} = \int_{\Omega_0} \boldsymbol{S} : \delta \boldsymbol{E} d\Omega_0 \tag{4.52}$$

Using the relations in [4.51] his can be expanded into

$$\delta W_{int} = \int_{\Omega_0} (\sigma_n(\delta\epsilon_\xi) + \sigma_b \xi_3(\delta\kappa_\xi)) d\Omega_0 \tag{4.53}$$

The next step is to reduce the body equation into an integration only over the length of the beam. Undet the assumption that the Youngs's modulus is constant over the cross section a preintegration can be performed. The results is familiar from beam theory, giving the final expression:

$$\delta W_{int} = \int_{L_0} (\epsilon_{\xi_1} EA(\delta\epsilon_{\xi_1}) + \kappa_{\xi_1} \mathrm{EI}(\delta\kappa_{\xi_1})) \mathrm{dL}_0 \tag{4.54}$$

The internal force is given by the first derivative of the virtual work equation w.r.t a displacement variable. With the above expression it can then be found as:

$$(f_{int})_r = \int_{L_0} (\epsilon_{\xi_1} \mathrm{EA} \frac{\delta\epsilon_{\xi_1}}{\partial u_r} + \kappa_{\xi_1} \mathrm{EI} \frac{\delta\kappa_{\xi_1}}{\partial u_r}) dL_0 \tag{4.55}$$

And subsequently the tangent Stiffness Matrix can be found as:

$$K_{rs} = \int_{L_0} (\frac{\partial\epsilon_{\xi_1}}{\partial u_s} \mathrm{EA} \frac{\delta\epsilon_{\xi_1}}{\partial u_r} + \epsilon_{\xi_1} EA \frac{\partial^2\epsilon_{\xi_1}}{\partial u_r \partial u_s} + \frac{\partial\kappa_{\xi_1}}{\partial u_s} \mathrm{EI} \frac{\delta\kappa_{\xi_1}}{\partial u_r} + \kappa_{\xi_1} EI \frac{\partial^2\kappa_{\xi_1}}{\partial u_r \partial u_s}) dL_0 \tag{4.56}$$

These equations can then be used to solve the equation (4.3) presented earlier.

4.3.2 Kirchhoff-Love Shell element

The equations for the shell element will correspond to those derived for the beam and will be based on the Kirchhoff-Love hypothesis.
The derivation is based on that of [27], and for a more elaborate version, further reading can be found there.

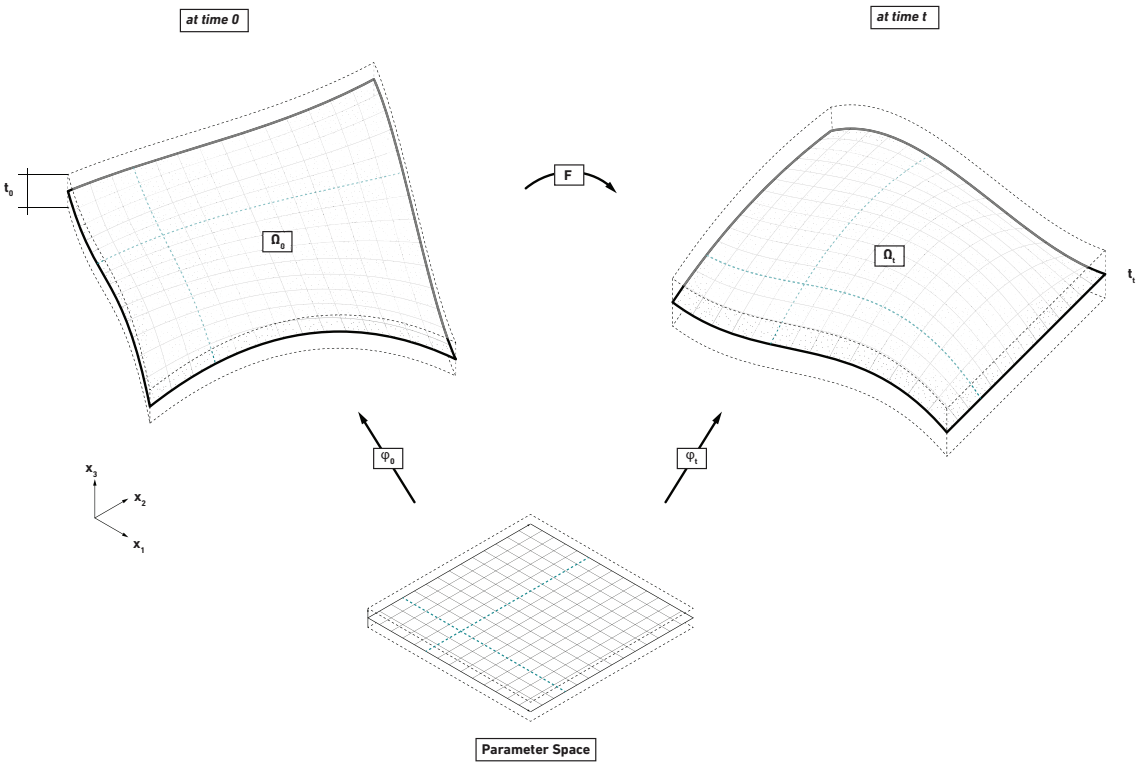


Figure 4.8: Shell geometry and mappings

4.3.2.1 Shell Geometry

The shell geometry follow from what was introduced in chapter about differential geometry. The shell body is parameterized like the beam, by considering a 3d dimensional body, described by a 2-dimensional body and coupling the thickness to the normal of the middle plane.
The body of the shell is parameterized by the midsurface and a thickness parameter. This is in line with the Kirchhoff-Love hypothesis.

A point $\boldsymbol{X}(\xi_1, \xi_2, \xi_3)$ on the shell is parameterized as:

$$\boldsymbol{X}(\xi_1, \xi_2, \xi_3) = \boldsymbol{\varphi}(\xi_1, \xi_2) + \frac{t}{2}\boldsymbol{n}\xi_3 \tag{4.57}$$

where:

- $\boldsymbol{X}$ *is a point in global coordinates {x,y,z}*
- $\boldsymbol{\varphi}$ *denotes a point on the middle surface of the shell.*
- t *is the thickness of the shell*
- $\boldsymbol{n}$ *is the normal vector of the middle surface.*

4.3.2.2 Large Displacement Kinematics

Just as the for the beam, the displacement is given by the difference between the reference configuration $\boldsymbol{\varphi}^0$ and the deformed configuration $\boldsymbol{\varphi}^t$ as:

$$\mathbf{u}^t = \boldsymbol{\varphi}^t - \boldsymbol{\varphi}^0 \tag{4.58}$$

The Green-Lagrange strains will be used for the strain descriptions. The choice of basis is important in the shell model, as opposed to the beam model, where only one strain direction exists. In the shell various coordinate systems could be used to describe the strains on the shell, as will be seen later. For now, the parameter directions of the shell (ξ_1, ξ_2) will be used, meaning that the undeformed *covariant* base vectors will be used, giving the Green-Lagrange strain defined as:

$$\hat{\boldsymbol{E}} = \frac{1}{2}(\boldsymbol{F}^T \cdot \boldsymbol{F} - \boldsymbol{I}) = \hat{E}_{ij}\boldsymbol{g}_0^i \otimes \boldsymbol{g}_0^j \tag{4.59}$$

The deformation gradient $\mathbf{F}$ is defined as in the previous chapter.
However, as we are working under the hypothesis of Kirchhoff-Love, the Green Lagrange strains can be reduced to the mid plane.

$$\hat{\boldsymbol{E}} = \frac{1}{2}(\boldsymbol{F}^T \cdot \boldsymbol{F} - \boldsymbol{I}) = \hat{E}_{\alpha\beta}\boldsymbol{g}_0^\alpha \otimes \boldsymbol{g}_0^\beta \tag{4.60}$$

Introducing the new basevectors of the center plane:

$\boldsymbol{a}_{\prime\alpha} = \boldsymbol{\varphi}_{\prime\alpha}(\xi_1, \xi_2)$ with $\alpha = \{1, 2\}$

instread of

$\boldsymbol{g}_{\prime i} = \mathbf{X}_{\prime i}(\xi_1, \xi_2, \xi_3)$ with $i = \{1, 2, 3\}$

The metric coefficients of the midsurface can be written as:

$$\alpha_{\alpha\beta} = \boldsymbol{a}_\alpha \cdot \boldsymbol{a}_\beta \tag{4.61}$$

$$\beta_{\alpha\beta} = \boldsymbol{a}_{\alpha'\beta} \cdot \mathbf{n} \tag{4.62}$$

Similarly to the steps in chapter 4.3.1 the strain components can be found as [27]:

$$\hat{E}_{\alpha\beta} = \frac{1}{2}(a^t_{\alpha\beta} - a^0_{\alpha\beta}) + t\xi_3(\beta^0_{\alpha\beta} - \beta^t_{\alpha\beta}) \tag{4.63}$$

where a_{ij} and κ_{ij} are the metric coefficients of the first and second fundamental form of surfaces in the reference and current configuration. The strain E_{ij} is a 2:nd order tensor. The first term (E_{11}) is the strain term derived in chapter 4.3.1, which is now expanded with a second direction along with two coupling terms (E_{12}, E_{21}) referring to shear strains.

$$\hat{E}_{\alpha\beta} = \frac{1}{2}\epsilon_{\alpha\beta} + t\xi_3\kappa_{\alpha\beta} \tag{4.64}$$

4.3.2.3 Constitutive Relations

For the shell model, the St. Venant theory will be used. This means that the constitutive relations are much like the beam given by the elastic Hooke tensor
According to the Total Lagrangian description presented in chapter 4.1.3 the PK2 stress are is used. The stress is defined in the parameteric space as (see Figure 4.5):

$$\hat{\boldsymbol{S}} = \boldsymbol{C} : \hat{\boldsymbol{E}} \tag{4.65}$$

or:

$$\hat{S}^{ij} = C^{ijkl}\hat{E}_{kl} \tag{4.66}$$

$$\hat{\boldsymbol{S}} = \hat{S}^{ij}\boldsymbol{g}_i^0 \otimes \boldsymbol{g}_j^0 \tag{4.67}$$

However, when dealing with this type of relationship, it will relate to physical properties as Young's Modulus and length measures. It is therefore desirable to set up this in the physical space. There for we need to instead work with the relationship:

$$\boldsymbol{S} = \boldsymbol{C} : \boldsymbol{E} \tag{4.68}$$

Which exist in physical space and relate to physical units [mm, N, etc].
This means mapping the strain from $\hat{\boldsymbol{E}}$ to $\boldsymbol{E}$ where $\boldsymbol{E}$ is given by:

$$\boldsymbol{E} = \hat{E}_{\alpha\beta}\tilde{\boldsymbol{g}}_0^{\alpha} \otimes \tilde{\boldsymbol{g}}_0^{\beta} = E_{ij}\tilde{\boldsymbol{e}}_0^{i} \otimes \tilde{\boldsymbol{e}}_0^{j} \tag{4.69}$$

where:

- E^{ij} denotes the strain tensor expressed in Euclidean units [m]
- $\tilde{\boldsymbol{e}}_i^0$ denotes a local Cartesian basis on the reference configuration.

This can be done using the term of the deformation gradient referring to the reference configuration ($\boldsymbol{F}^0$) or by using the basis changes presented in chapter 4.2.3.

However, as we work with plane stress conditions we reduce the conditions to the plane conditions:

$$\boldsymbol{E} = E_{\alpha\beta}\tilde{\boldsymbol{e}}_0^{\alpha} \otimes \tilde{\boldsymbol{e}}_0^{\beta} \tag{4.70}$$

Through this the constitutive model can be defined for a local cartesian basis as:

$$S^{\alpha\beta} = C^{\alpha\beta\gamma\delta}E_{\gamma\delta} \tag{4.71}$$

$$\boldsymbol{S} = S^{\alpha\beta}\boldsymbol{e}_{\alpha}^0 \otimes \boldsymbol{e}_{\beta}^0 \tag{4.72}$$

With $C^{\alpha\beta\gamma\delta}$ being defined in physical coordinates, meaning that conventional models for plane stress can be used. By using Voight notation the stress-strain can be written as

$$\begin{bmatrix} S^{11} \\ S^{22} \\ S^{12} \end{bmatrix} = \mathbf{C} \begin{bmatrix} E^{11} \\ E^{22} \\ 2E^{12} \end{bmatrix} \tag{4.73}$$

From equation (4.64) it can be observed that, the stress is composed of two terms where one is constant and refers to the normal stress, and the second is linearly dependent on the thickness of the shell which refers to the bending stress. However, in the context of shell mechanics, it is common practice to work with membrane forces $[\frac{N}{m}]$ instead of stresses $[\frac{N}{m^2}]$.

Through integration over the thickness, the membrane forces can be expressed as:

$$M^{\alpha\beta} = \int_{-t/2}^{t/2} S^{\alpha\beta} d\xi_3 = \int_{-t/2}^{t/2} C^{\alpha\beta\gamma\delta}E_{\gamma\delta} d\xi_3 \tag{4.74}$$

It can be observed that the strain term consists of one constant and one linearly dependent w.r.t the height parameter. This means that the membrane force can be split into a normal and bending term as:

$$M^{\alpha\beta} = n^{\alpha\beta} + m^{\alpha\beta} \tag{4.75}$$

The integration for each term yields:

$$n^{\alpha\beta} = C^{\alpha\beta\gamma\delta}t\epsilon_{\gamma\delta} \tag{4.76}$$

$$m^{\alpha\beta} = C^{\alpha\beta\gamma\delta}\frac{t^3}{12}\kappa_{\gamma\delta}d \tag{4.77}$$

which can be written in Voight notation as:

$$\begin{bmatrix} n^{11} \\ n^{22} \\ n^{12} \end{bmatrix} = t\mathbf{C} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ 2\epsilon_{12} \end{bmatrix} \tag{4.78}$$

$$\begin{bmatrix} m^{11} \\ m^{22} \\ m^{12} \end{bmatrix} = \frac{t^3}{12}\mathbf{C} \begin{bmatrix} \kappa_{11} \\ \kappa_{22} \\ 2\kappa_{12} \end{bmatrix} \tag{4.79}$$

Note that with the initial strains defined in an ortho-normal basis in physical space, the C matrix is straightforwardly defined. A further elaboration on what material models used in this thesis will be explored in the following chapter on Extended Material Properties.
The most basic C matrix is given by the plane stress matrix:

$$\boldsymbol{C} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{4.80}$$

4.3.2.4 Equilibrium

A derivation for the equilibrium equations can be found in [6]. Here, only the key equations will be stated. The formulation of the internal work is given by:

$$\delta W_{int} = \int_{\Omega_0} \mathbf{S} : \delta\mathbf{E} \; d\Omega_0 \tag{4.81}$$

$$\delta W_{int} = \int_{A_0} (\mathbf{n} : \delta\boldsymbol{\epsilon} + \mathbf{m} : \delta\boldsymbol{\kappa}) dA \tag{4.82}$$

In the same manner as the beam chapter, the final internal forces can then be found as:

$$(f_{int})_r = \int_{A_0} (\mathbf{n} : \frac{\partial(\delta\boldsymbol{\epsilon})}{\partial u_r} + \mathbf{m} : \frac{\partial(\delta\boldsymbol{\kappa})}{\partial u_r}) dA \tag{4.83}$$

4.4 Extended Material Properties

To be able to inform the equilibrium geometry and to control it, a set of extended material properties are needed. The following segment will describe the material profile used in this thesis. The main approach is to modify the stiffness of the piece as a means of adjusting the equilibrium geometry.
The stiffness of the patch relates to two things, the cross section and the material model. In this case, the material will always be kept linear elastic and the modification will be done on the thickness (cross section).

- *Non-uniform thickness distribution*
- *Orthotropic material model*

4.4.1 Thickness Mapping

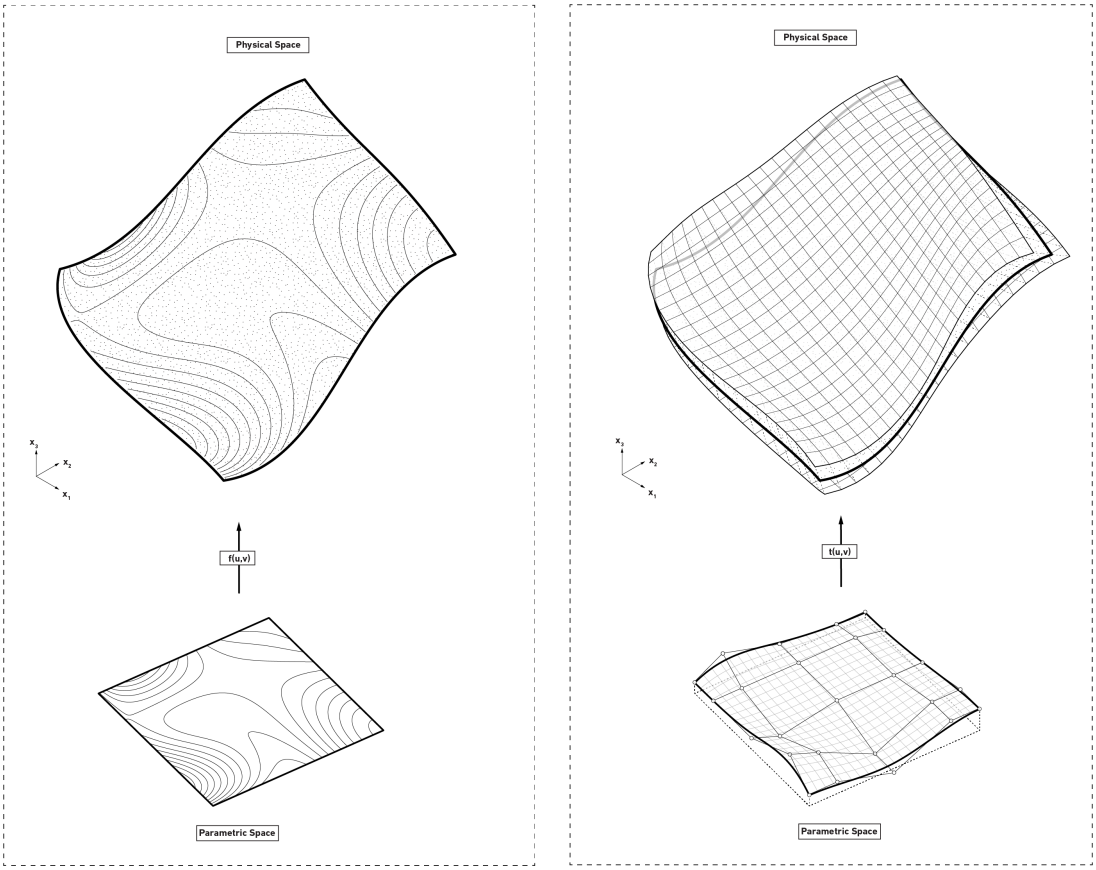


Figure 4.9: Mapping a function onto a body using the parameterisation

The thickness will be controlled by mapping a function w.r.t to the parameter space. An arbitrary function can easily be mapped to the physical space through the underlying parameterisation. For the purpose of this study, an idependent NURBS map is used to manipulate the thickness field of the patch. The reason for this is to get a simple control over geometry in a 3D modeling environment. In theory any interpolation and complexity can be mapped, but one should beware with the integration scheme and parameterisation of the geometry to make sure the proper behaviour is captured.

Through this the thickness parameter is no longer constant, thus turning the internal force relationships into:

$$n(\xi_1, \xi_2) = \mathbf{C}t(\xi_1, \xi_2)\epsilon(\xi_1, \xi_2) \tag{4.84}$$

$$m(\xi_1, \xi_2) = \mathbf{C}\frac{t(\xi_1, \xi_2)^3}{12}\epsilon(\xi_1, \xi_2) \tag{4.85}$$

4.5 Boundary Conditions

4.5.1 Lagrange Multipliers

In isogeometric analysis boundary conditions can not, in general, simply be imposed on the degrees of freedom, as these are not always interpolated. Instead one can constrain the values of certain parameters using LaGrange multipliers, thus constraining the movement of the control points to keep a certain value. The system of equations structural mechanics mainly is concerned with is on the form

$$\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f} \tag{4.86}$$

However, we now also have a set of prescribed conditions for our solution $\boldsymbol{a}$
This could be formulated as a second set of equations describing those constraints.

$$\boldsymbol{B}\boldsymbol{a} = \boldsymbol{u_c} \tag{4.87}$$

where $\boldsymbol{u_c}$ describes the values of the desired constraints. (Not to be confused with the displacement vector)

The general concept essentially means that we find the solution to the equation with respect to a certain constraint. The method for enforcing these are by adding a set of constraint equations to the system of equations thus expanding,

$$\mathbf{K}\Delta\boldsymbol{a} = \boldsymbol{f}_{ext} - \boldsymbol{f}_{int}$$

into,

$$\begin{bmatrix} \mathbf{K} & \boldsymbol{B}^T \\ \boldsymbol{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{a} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{ext} \\ \boldsymbol{u}_c \end{bmatrix} - \begin{bmatrix} \boldsymbol{f}_{int} \\ \mathbf{0} \end{bmatrix}$$

$\boldsymbol{B}$ is the matrix that defines the constraints, by relating the values of the DOFS to each other. The values of $\boldsymbol{\lambda}$ only serve to enforce the values of $\boldsymbol{u_c}$ and can be discarded in the solution of the final displacements. It should be noted that the values of the Lagrange multipliers correspond to the equivalent nodal load vector needed to enforce the constraints.

4.5.2 NURBS based constraints

Enforcing Lagrange based constraints using the NURBS logic of the geometry definition is a rather straight forward process.

In 1D the process is as follows:
A point $\boldsymbol{P_c}$ on the curve can be found through

$$\boldsymbol{P}_c = \sum N^i(\xi_c)\hat{\boldsymbol{x}}^i \tag{4.88}$$

Similarly, the displacement vector of a point anywhere on the curve can be found through,

$$\boldsymbol{U}_c = \sum N^i(\xi_c)\hat{\boldsymbol{a}}^i \tag{4.89}$$

To prescribe the value $\boldsymbol{U}_c$ at the arbitrary parameter ξ_c on the curve, the constraints on the control points are given by the evaluation of the basis functions at that parameter.
This means that:

$$\boldsymbol{B} = N^i(\xi_c) \tag{4.90}$$

It's known from the chapter about NURBS that the basis functions will have *p-1* non-zero values in the knotspan that ξ_c is found in.

For example:
If we want to prescribe the value of the y displacement at parameter ξ_c found in knot span k, we get the following constraint:

$$\begin{bmatrix} N^k(\xi_c) & ... & N^{k+p-1}(\xi_c) \end{bmatrix} \begin{bmatrix} a_y^k \\ ... \\ a_y^{k+p-1} \end{bmatrix} = u_c \tag{4.91}$$

The rest of the entries in the $\boldsymbol{B}$ matrix will be zero, as the values of the rest of the basis functions at parameter ξ_c are zero.

4.6 Stress Output

Recalling Figure 4.5 the main stress used for the calculations is not a true stress. Therefore, when analysis is said and done, and one is interested in the results, the stresses (PK-2 - $\mathbf{S}$) of the shell model need to be mapped to the physically relevant ones (Cauchy - $\boldsymbol{\sigma}$).

The mapping is defined using the deformation gradient $\mathbf{F}$ as:

$$\boldsymbol{\sigma} = J^{-1}\mathbf{F}^T\mathbf{S}\mathbf{F} \tag{4.92}$$

The jacobian J denotes the relation in density (or area) between reference and deformed configuration. It can be calculated as the determinant of the deformation gradient.

This can be done using the basis changes in ch. 4.2.3 along with the scalefactor. This would mean mapping the stress from the basis on reference configuration to the corresponding basis defined on the Deformed configuration.

4.7 Membrane Locking

A problem associated with this nonlinear formulation is membrane locking. Different approaches can be undertaken to overcome this problem. In this thesis it will be overcome by raising the polynomial degree of the NURBS basis as done in [6]. A reference study can be found in Appendix I.

5

Computer Implementation

The following chapter will discuss the means of implementing and solving the equations presented in the following chapter using NURBS. It should be noted that the theories derived has no direct connection to NURBS and could be solved using different geometrical basis. Therefore, the logic of discretising the equations using a NURBS basis will be discussed.

Different solutions schemes for solving the non-linear system will be described using path following techniques along with dynamic relaxation.

5.1 Discretisation using NURBS

The following section will describe the implementation of NURBS as a base for discretizing the formulations in the previous chapter.
For the shell, the discretization has been left out, but the reader is instead referred to [27].

5.1.1 Beam Element

The location of a point on the curve, can be described using NURBS as

$$\boldsymbol{\varphi}^t(\xi_1) = \sum N^i(\xi_1)\hat{\boldsymbol{x}}^{i^t} = \sum N^i(\xi_1)(\hat{\boldsymbol{x}}^{i^0} + \hat{\boldsymbol{u}}^{i^t}) \tag{5.1}$$

Through this the base vectors can straightforwardly be found as

$$\mathbf{a}^t_{\xi_1} = \boldsymbol{\varphi}^t_{,\xi_1} = N^i_{,\xi_1}(\hat{\boldsymbol{x}}^{i^0} + \hat{\boldsymbol{u}}^{i^t}) \tag{5.2}$$

For the computation of the stiffness matrix and internal force vector, given in equations (4.55) and (4.56) the derivatives of the strain w.r.t the displacements are needed.

These can be found as:

$$\epsilon_{\xi' r} = \frac{1}{2}(\alpha^t_{,r} - \alpha^0_{,r}) \tag{5.3}$$

and

$$\kappa_{\xi' r} = (\beta^0_{,r} - \beta^t_{,r}) \tag{5.4}$$

The derivatives of the metric coefficients are found from the derivatives of the base vectors. They are given as:

$$\mathbf{a}^t_{\xi_1' r} = \sum_i N^i_{,\xi}(\hat{\boldsymbol{x}}^{i^0} + \hat{\boldsymbol{u}}^{i^t})_{,r} \tag{5.5}$$

As the initial positions are invariant to the displacements, it can be written as:

$$\mathbf{a}^t_{\xi' r} = \sum_i N^i_{,\xi}\hat{\boldsymbol{u}}^{i^t}_{,r} \tag{5.6}$$

Here it should be noted that the derivative of the displacement vector w.r.t to the displacements assumes the following logic:

$$\mathbf{u}_{k' r} = \delta_{kr} = \begin{cases} 1 & if\ k = r \\ 0 & otherwise \end{cases} \tag{5.7}$$

Further, as the displacement vector $\mathbf{u}$ is a set of constant values, the second derivative w.r.t the displacements become:

$$\mathbf{u}_{\prime rs} = 0 \tag{5.8}$$

Given the Kirchhoff-Love hypothesis, the normal vector is coupled to the tangent vector, as they always are perpendicular. Thus, the derivative $\mathbf{n}^t_{\prime r}$ can be found directly from the derivative of the base vector. Given the derivatives of the base vectors, we can compute the derivatives of the metric coefficients.

$$\alpha^t_{\prime r} = (\mathbf{a}^t_\xi \cdot \mathbf{a}^t_\xi)_{\prime r} \tag{5.9}$$

$$\alpha^0_{\prime r} = 0 \tag{5.10}$$

$$\kappa^t_{\prime r} = -(\mathbf{a}^t_{\xi\prime\xi} \cdot \mathbf{n}^t_\xi)_{\prime r} \tag{5.11}$$

$$\kappa^0_{\prime r} = 0 \tag{5.12}$$

Which gives the expressions for the first derivatives of the strains as:

$$\frac{\partial \epsilon_\xi}{\partial u_r} = \frac{1}{2}(\mathbf{a}^t_\xi \cdot \mathbf{a}^t_\xi)_{\prime r} = \mathbf{a}^t_\xi \cdot \mathbf{a}^t_{\xi\prime r} \tag{5.13}$$

and

$$\frac{\partial \kappa_\xi}{\partial u_r} = -(\mathbf{a}^t_{\xi\prime\xi} \cdot \mathbf{n}^t_\xi)_{\prime r} = -(\mathbf{a}^t_{\xi\prime\xi\prime r} \cdot \mathbf{n}^t + \mathbf{a}^t_{\xi\prime\xi} \cdot \mathbf{n}^t_{\prime r}) \tag{5.14}$$

Further, the second derivatives can be found as:

$$\frac{\partial^2 \epsilon_\xi}{\partial u_r \partial u_s} = \frac{1}{2}(\mathbf{a}^t_\xi \cdot \mathbf{a}^t_\xi)_{\prime rs} \tag{5.15}$$

$$\frac{\partial^2 \kappa_\xi}{\partial u_r \partial u_s} = -(\mathbf{a}^t_{\xi\prime\xi} \cdot \mathbf{n}^t_\xi)_{\prime rs} \tag{5.16}$$

Using the chain rule and remembering equation (5.8) the expressions for the second derivatives becomes:

$$\frac{\partial^2 \epsilon_\xi}{\partial u_r \partial u_s} = \mathbf{a}^t_{\xi\prime r} \cdot \mathbf{a}^t_{\xi\prime s} \tag{5.17}$$

$$\frac{\partial^2 \kappa_\xi}{\partial u_r \partial u_s} = \mathbf{a}^t_{\xi\prime\xi\prime r} \cdot \mathbf{n}^t_{\prime s} + \mathbf{a}^t_{\xi\prime\xi\prime s} \cdot \mathbf{n}^t_{\prime r} \tag{5.18}$$

By dividing the stiffness matrix into a linear (L) and nonlinear (NL) part and further into a membrane (m) and bending (b) part, the following parts of the tangent stiffness can be found.

$$\mathbf{K}^L_m = \int_{\Omega^0} \frac{\partial \epsilon_\xi}{\partial u_r} EA \frac{\partial(\delta\epsilon_\xi)}{\partial u_s} d\Omega_0 = \int_{\Omega^0} \mathbf{B}^{L^T}_m EA \mathbf{B}^L_m d\Omega_0 \tag{5.19}$$

$$\mathbf{K}^L_b = \int_{\Omega^0} \frac{\partial \kappa_\xi}{\partial u_r} EI \frac{\partial(\delta\kappa_\xi)}{\partial u_s} d\Omega_0 = \int_{\Omega^0} \mathbf{B}^{L^T}_b EI \mathbf{B}^L_b d\Omega_0 \tag{5.20}$$

$$\mathbf{K}^{NL}_m = \int_{\Omega^0} \epsilon_\xi EA \frac{\partial^2(\delta\epsilon_\xi)}{\partial u_r \partial u_s} d\Omega_0 = \int_{\Omega^0} \epsilon_\xi EA \mathbf{B}^{NL}_m d\Omega_0 \tag{5.21}$$

$$\mathbf{K}^{NL}_b = \int_{\Omega^0} \kappa_\xi EI \frac{\partial^2(\delta\kappa_\xi)}{\partial u_r \partial u_s} d\Omega_0 = \int_{\Omega^0} \kappa_\xi EI \mathbf{B}^{NL}_b d\Omega_0 \tag{5.22}$$

The B-matrices are defined as follows.

$$\mathbf{B}_m^L = \begin{bmatrix} N^1_{\prime\xi} a^{x_1} & N^1_{\prime\xi} a^{x_2} & N^2_{\prime\xi} a^{x_1} & N^2_{\prime\xi} a^{x_2} & ... & N^n_{\prime\xi} a^{x_1} & N^n_{\prime\xi} a^{x_2} \end{bmatrix} \tag{5.23}$$

$$\mathbf{B}_b^L = \begin{bmatrix} -N^1_{\prime\xi\xi} n^{x_1} + N^1_{\prime\xi} a^{x_2}_{\prime\xi} & -N^1_{\prime\xi\xi} n^{x_2} + N^1_{\prime\xi} a^{x_1}_{\prime\xi} & ... & -N^n_{\prime\xi\xi} n^{x_1} + N^n_{\prime\xi} a^{x_2}_{\prime\xi} & -N^n_{\prime\xi\xi} n^{x_2} + N^n_{\prime\xi} a^{x_2}_{\prime\xi} \end{bmatrix} \tag{5.24}$$

$$\mathbf{B}_m^{NL} = \mathbf{B}_{m-part}^{NL^T} \mathbf{B}_{m-part}^{NL} \tag{5.25}$$

$$\mathbf{B}_{m-part}^{NL} = \begin{bmatrix} N^1_{\prime\xi} & 0 & N^2_{\prime\xi} & 0 & ... & N^n_{\prime\xi} & 0 \\ 0 & N^1_{\prime\xi} & 0 & N^2_{\prime\xi} & ... & 0 & N^n_{\prime\xi} \end{bmatrix} \tag{5.26}$$

$$\mathbf{B}_b^{NL} = \mathbf{B}_{b-part}^{NL^T} + \mathbf{B}_{b-part}^{NL} \tag{5.27}$$

$$\mathbf{B}_{b-part}^{NL} = \mathbf{B}_{b-part1}^{NL} \mathbf{B}_{b-part2}^{NL} \tag{5.28}$$

with

$$\mathbf{B}_{b-part1}^{NL} = \begin{bmatrix} N^1_{\prime\xi\xi} & 0 & N^2_{\prime\xi\xi} & 0 & ... & N^n_{\prime\xi\xi} & 0 \\ 0 & N^1_{\prime\xi\xi} & 0 & N^2_{\prime\xi\xi} & ... & 0 & N^n_{\prime\xi\xi} \end{bmatrix} \tag{5.29}$$

$$\mathbf{B}_{b-part2}^{NL} = \begin{bmatrix} 0 & -N^1_{\prime\xi} & 0 & -N^2_{\prime\xi} & ... & 0 & -N^n_{\prime\xi} \\ -N^1_{\prime\xi} & 0 & -N^2_{\prime\xi} & 0 & ... & N^n_{\prime\xi} & 0 \end{bmatrix} \tag{5.30}$$

Through this, the tangent stiffness matrix can be obtained.

$$\mathbf{K} = \mathbf{K}_m^L + \mathbf{K}_b^L + \mathbf{K}_m^{NL} + \mathbf{K}_b^{NL} \tag{5.31}$$

The same matrices can also be used to compute the internal force vector as:

$$\boldsymbol{f}_{int} = \int_{\Omega^0} \epsilon EI \mathbf{B}_m^{L^T} + \kappa EI \mathbf{B}_b^{L^T} d\Omega_0 \tag{5.32}$$

5.2 Nonlinear Solution Procedures

Up until now, the relationships describing the behaviour of beams and shells has been described. In the previous section, it was described how to discretize these relationships and equations using NURBS. The following chapter will therefore focus on the aspects one has to consider when solving these types of systems. As stated in chapter (4.1.3), the central relationship that we have to satisfy is that of equilibrium in the end. This means that the internal forces of the body should be in equilibrium with the external ones. The nonlinear solution can not be found directly from the start, but it has to be computed in a series of steps. This chapter will provide a description of two different ways of tracing the nonlinear equilibrium. These are:

- *Stiffness based method (Matrix)*
- *Dynamic Relaxation*

Both of these approaches can be used to solve the same systems, however, their ways of doing it are rather different. The stiffness method refers to the linearized equation (4.8) which is based on a the tangent stiffness matrix. This is an application of the Newton-Raphson method, which is used in many fields of mathematics. This method traces a series of equilibriums until it finds the appropriate one. It should be noted that using this approach, the body never moves, but is always in equilibrium. It simply goes from one to the next, neglecting any inertia effects etc. This is analogous with a load that is applied slowly.

Instead of solving the problem as a set of static equilibrium, the method of dynamic Relaxation will instead treat the movement of structure as it moves into the final equilibrium. This means not solving equilibrium in each step but instead treating the problem as a pseudodynamic problem. Here no stiffness is needed, but instead the out of balance forces will result in accelerations and velocities, thus, moving the body towards it's final state.

5.2.1 Newton Raphson

As previosly mentioned, the equation (4.8) is a linearisation according to the Newton-Raphson procedure. This is a method used to find solutions to equations in many different applications. The central aspect to the newton-Raphson procedure is that a linearized step based on the derivative is used to approximate the function. Taking steps in the gradient (or tangent) direction makes it possible to trace the load-displacement curve of a structure. The stiffness is the relationship between applied load and displacement, which means that the tangent stiffness matrix (the stiffness of a system at time t) is the slope of the load displacement curve.

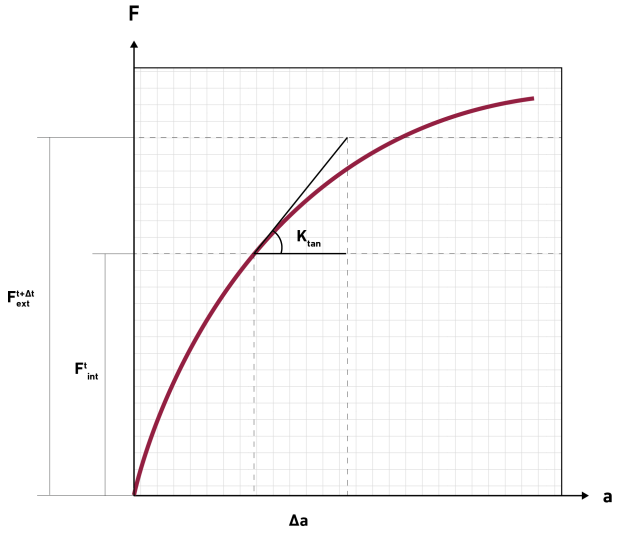


Figure 5.1: Components of a Newton Raphson procedure for a nonlinear equilibrium problem

5.2.1.1 General Solution Procedure

The equation to be solved is described as a step in the load. This will then result in a step in the displacement. In nonlinear FEM this used in combination with an incremental load. The load is incremented in a series of steps and the NR method is used to find the response of the function at each step.

A problem arises as the equilibrium path will deviate from the linear approximation, there will be an associated error with each computation. A simple incremental process will not be precise as it will continuously deviate from the true equilibrium path for each step, thus, very small steps will be necessary to be able to arrive at a good solution. Figure 5.2 shows this deviation from the equilibrium on the left.

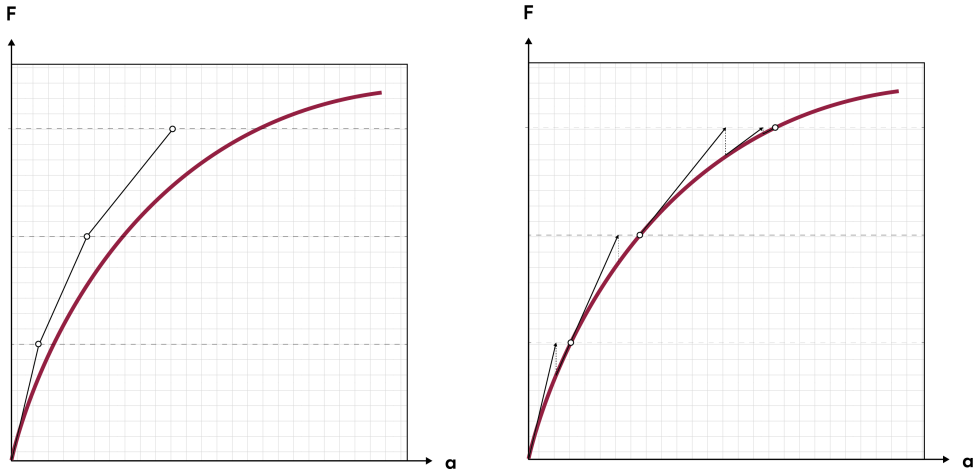


Figure 5.2: No equilibrium iterations (left) With equilibrium iterations (right)

To solve this problem, equilibrium must be found at each load. This can be done by iterating until the error is small before applying the next load increment. As the equilibrium will be considered through iterations, the tangent stiffness is not essential. This means that after the initial step is computed, the difference between the internal and external force (ΔF) is found. The process is then repeated until ΔF is smaller than a certain tolerance. This process is shown in Figure 5.2 on the right. Algorithm 1 describes a general full NR procedure. For a modified process, simply skip step 12.

Algorithm 1 Newton Raphson Procedure

1: **procedure** FULL NR($\boldsymbol{f}_{ext}$)
2: $\boldsymbol{f}_{int} \leftarrow 0$, $\boldsymbol{f}_{ext} \leftarrow 0$, $a \leftarrow 0$
3: $\boldsymbol{f}_{step} \leftarrow$load increment
4: **while** $\boldsymbol{f}_{ext} < \boldsymbol{f}_{max}$ **do** ▷ Loop over increments
5: $\boldsymbol{f}_{ext} = \boldsymbol{f}_{step} * i$ ▷ Current external load
6: <compute $\boldsymbol{K}_{tan}$> ▷ Tangent stiffness matrix
7: Solve $\Delta\boldsymbol{a} = \boldsymbol{K}_{tan}^{-1}(\boldsymbol{f}_{ext} - \boldsymbol{f}_{int})$ ▷ Solve displacement step
8: $\boldsymbol{a} += \Delta\boldsymbol{a}$
9: Update $\boldsymbol{f}_{int}$
10: $\Delta\boldsymbol{f} = \boldsymbol{f}_{ext} - \boldsymbol{f}_{int}$
11: **while** $\Delta\boldsymbol{f} > tol$ and $iter < maxIter$ **do**
12: <compute $\boldsymbol{K}$> ▷ Tangent stiffness matrix
13: Solve $\Delta\boldsymbol{a} = \boldsymbol{K}^{-1}(\Delta\boldsymbol{f})$ ▷ Solve displacement step
14: $a += \Delta\boldsymbol{a}$
15: Update $\boldsymbol{f}_{int}$
16: Update $\Delta\boldsymbol{f}$
17: iter++
18: i++
19: **return** $\boldsymbol{a}$

However, as the equilibrium is iterated the stiffness matrix is computed at each step. This could be a time consuming task and, since there is iterations, it is not necessary. Therefore, one does not need to use the exact stiffness at each step. In general three different approaches can be defined, considering the way the stiffness is used in the iteration process. These are:

- *Full Newton Raphson*
- *Modified Newton Raphson*
- *Linear Elastic Newton Raphson*

These are illustrated in Figure 5.3

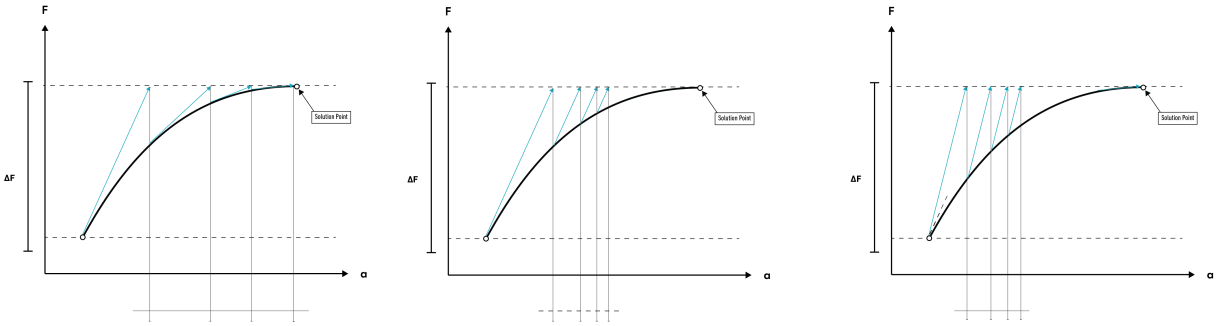


Figure 5.3: Different iteration schemes

One the left, the tangent stiffness is updated both at iteration and increment. This is called a *full* Netwon-Raphson. In the middle graphic, the stiffness is only updated at increment, and kept constant for the iteration. This is called a *modified* scheme. On the right is the extreme case where the linear stiffness is used at all times, referred to as *linear*.

Another aspect of this kind of solution procedure is what variable is used for the increment. Most commonly three types can be defined. These are:

- *Force Control*
- *Displacement Control*
- *Arc-Length control*

Most of what was said before related to force control, and in the coming chapters force and displacement control will be elaborated on further. Arc-Length is left out of this discussion and for the interested reader, other literature is needed.

5.2.1.2 Displacement Control

A problem exists with tracing unknown structures using load levels. One does not always know the peak load of a structure. Further, some structure may experience a negative stiffness at some point in the process at this stage, the load should instead be reduced. Keeping increasing the load at this case will cause the process to diverge, as no equilibrium can be found. Figure 5.4 shows a load -displacement relationship where there exist no equilibrium for the load level.

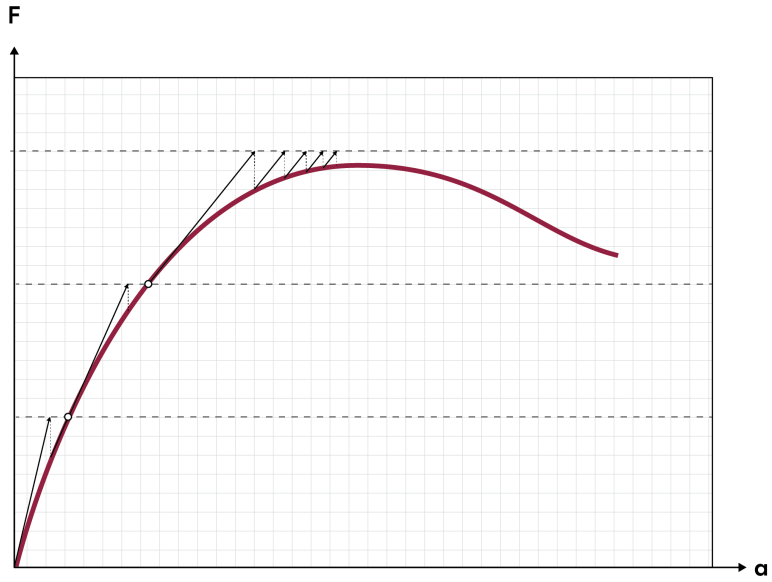


Figure 5.4: Divergent modified Newton Raphson solution

The procedure described above concerns a force controlled iterative procedure. However, in the context of form finding and large displacements as opposed to analysis, the final load is very rarely known. This makes using external load to control a final shape highly unintuitive.

Instead controlling the final positions is much more inline with the actual process of designing the type of bent shells that are the aim of this study. Another advantage of displacement control over load control is that in this we can overcome the previously mentioned peaks. This is illustrated in Figure 5.5

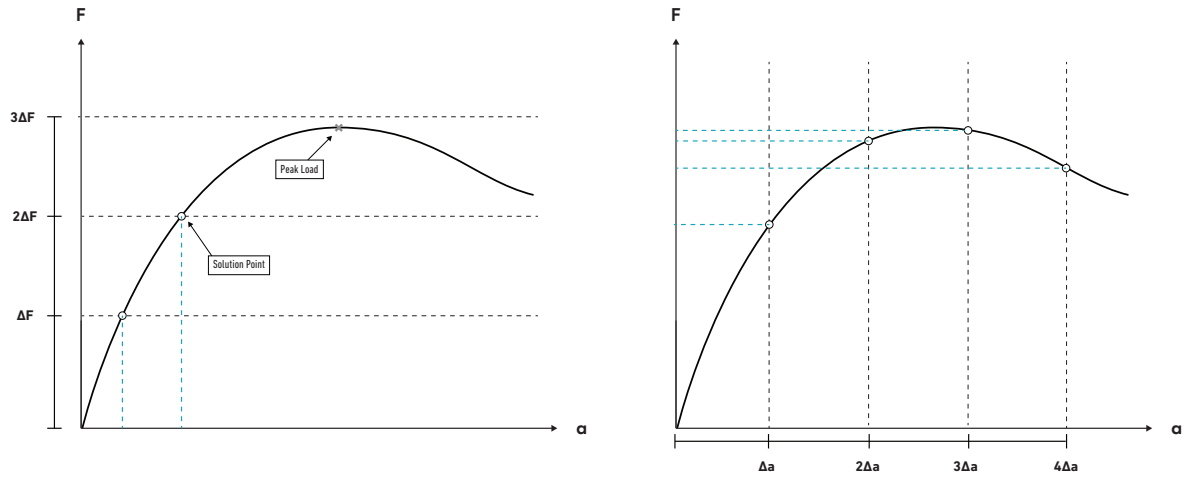


Figure 5.5: Comparison between displacement and force control

The general way of controlling displacement is done through the following.
The following algorithm show the logic of a displacement controlled procedure.

Algorithm 2 Displacement Control

procedure FULL NR($\boldsymbol{f}_{ext}$)
  $\boldsymbol{f}_{int} \leftarrow 0$, $\boldsymbol{f}_{ext} \leftarrow 0$, $\boldsymbol{a} \leftarrow 0$
  $\boldsymbol{a}_{step} \leftarrow$displacement increment
  while *steps* $<$ *maxSteps* **do** ▷ Loop over increments
    $\Delta\boldsymbol{a}_c = \boldsymbol{a}_{step}$ ▷ Prescribed displacement increment
    <compute $\boldsymbol{K}_{tan}$> ▷ Tangent stiffness matrix
    <Partition into $\boldsymbol{K}_{ff}, \boldsymbol{K}_{fc}$ and $\boldsymbol{K}_{cc}$>
    Compute $\boldsymbol{f}_{eq} = \boldsymbol{K}_{fc}\Delta\boldsymbol{a}_c$
    Solve $\Delta\boldsymbol{a}_f = \boldsymbol{K}_{ff}^{-1}(\boldsymbol{f}_{eq} + \boldsymbol{f}_{ext} - \boldsymbol{f}_{int})$ ▷ Solve displacement step
    $\boldsymbol{a}_f += \Delta\boldsymbol{a}_f$
    Update $\boldsymbol{f}_{int}$
    $\Delta\boldsymbol{f} = \boldsymbol{f}_{ext} - \boldsymbol{f}_{int}$
    while $\Delta\boldsymbol{f} >$ *tol* and *iter* $<$ *maxIter* **do**
      <compute K_{tan}> ▷ Tangent stiffness matrix
      Solve $\Delta\boldsymbol{a}_f = \boldsymbol{K}_{ff}^{-1}(\Delta\boldsymbol{f})$ ▷ Solve displacement step
      $\boldsymbol{a} += \Delta\boldsymbol{a}$
      Update $\boldsymbol{f}_{int}$
      Update $\Delta\boldsymbol{f}$
      iter++
    i++
  return $\boldsymbol{a}$

By separating the system of equations

$$\mathbf{K}\Delta\boldsymbol{a} = \boldsymbol{f}_{ext} - \boldsymbol{f}_{int}$$

into,

$$\begin{bmatrix}\mathbf{K}_{ff} & \mathbf{K}_{cf}\\ \mathbf{K}_{fc} & \mathbf{K}_{cc}\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{a}_f\\ \Delta\boldsymbol{a}_c\end{bmatrix} = \begin{bmatrix}\boldsymbol{f}_{ext,f}\\ \boldsymbol{f}_{ext,c}\end{bmatrix} - \begin{bmatrix}\boldsymbol{f}_{int,f}\\ \boldsymbol{f}_{int,c}\end{bmatrix}$$

where the subscript f indicates free degrees of freedom and subscript c denotes constrained degrees of freedom.

If we assume that in the displacement controlled process no external loads are present, the system reduces into,

$$\begin{bmatrix}\mathbf{K}_{ff} & \mathbf{K}_{cf}\\ \mathbf{K}_{fc} & \mathbf{K}_{cc}\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{a}_f\\ \Delta\boldsymbol{a}_c\end{bmatrix} = -\begin{bmatrix}\boldsymbol{f}_{int,f}\\ \boldsymbol{f}_{int,c}\end{bmatrix}$$

From this system of equations the following expression can be obtained,

$$\Delta\boldsymbol{a}_f = -\mathbf{K}_{ff}^{-1}(\boldsymbol{f}_{int,f} + \mathbf{K}_{fc}\Delta\boldsymbol{a}_c) \tag{5.33}$$

which gives the displacements of the free degrees of freedom. The term $\mathbf{K}_{fc}\Delta\boldsymbol{a}_c$ can be interpreted as the equivalent external force vector required to enforce the constraints.

In the subsequent equilibrium iterations the vector of constrained nodes $\Delta\boldsymbol{a}_c$ should be set to zero, as they after the first step have the desired value. This reduces the iterative procedure to,

$$\Delta \boldsymbol{a}_f = -\mathbf{K}_{ff}^{-1} \boldsymbol{f}_{int,f} \tag{5.34}$$

From this it can be seen that displacements are controlled instead of the load increment. The following procedure is still a load controlled w.r.t the out of balance force vector ΔF.

5.2.2 Dynamic Relaxation

Algorithm 3 Dynamic Relaxation Procedure

procedure DYNAMIC RELAXATION($\boldsymbol{f}_{ext}$)
  $\boldsymbol{f}_{int} \leftarrow 0$, $\boldsymbol{f}_{ext} \leftarrow \boldsymbol{f}_{tot}$, $\boldsymbol{a} \leftarrow 0$, $\Delta t \leftarrow timestep$
  $k \leftarrow 0$, $\boldsymbol{V}^t \leftarrow 0$ ▷ kinetic Energy, initial velocity
  while $k > tol$ and $\Delta F_{max} < tol$ **do** ▷ Loop over increments until at rest
    <compute $\boldsymbol{f}_{int}$> ▷ Internal Forces
    $\Delta \boldsymbol{f} = \boldsymbol{f}_{ext} - \boldsymbol{f}_{int}$ ▷ Out of balance force
    $\dot{\boldsymbol{V}}^t = \boldsymbol{M}^{-1} \Delta \boldsymbol{f}$ ▷ Compute Acceleration
    <Update $\boldsymbol{V}^{t+\Delta t}$>
    <compute $\Delta \boldsymbol{a} = \Delta t \boldsymbol{V}^{t+\frac{\Delta t}{2}}$> ▷ Compute Node movement
    $\boldsymbol{a} += \Delta \boldsymbol{a}$
    Update k
    i++
  return a

Dynamic relaxation differs quite significantly from the above mentioned methods. The difference lies in the fact that the stepped methods finds intermediate equilibriums on their way to the final state by successively applying loads or displacements. In dynamic relaxation the full load is applied instantly, and instead of tracing an equilibrium path, it traces the movement of the nodes w.r.t to the applied load.

The fundamental difference is that no stiffness matrix then is needed but the resulting forces on each node will turn into accelerations and velocities that will move the structure. This means that the mass of the system needs to be considered somehow. As the equations formulated for the beams and shells are rotation free, the application of dynamic relaxation is made easier as no moments of inertia has to be included, and the resulting forces on nodes are very straightforward to calculate using the vector of internal forces, as described earlier, and the vector of applied load.

Algorithm 3 describes a typical dynamic relaxation routine. At the heart of the algorithm lies the following equation [20]

$$\mathbf{X}^{t+\Delta t} = \mathbf{X}^t + \Delta t \mathbf{V}^{t+\frac{\Delta t}{2}} \tag{5.35}$$

which corresponds to a forward euler algorithm, approximating the updated positions based on the current positions and the speed. This represent a discrete solution to the equation of motion. The velocities are found from the accelerations, which in turn are given by the residual forces [19].

$$\mathbf{V}^{t+\Delta t} = \mathbf{V}^t + \frac{\dot{\boldsymbol{V}} \Delta t}{2} \tag{5.36}$$

$$\dot{\boldsymbol{V}}^t = \boldsymbol{M}^{-1} \Delta \boldsymbol{f}^t \tag{5.37}$$

where

- $\boldsymbol{M}$ is the Mass matrix
- $\Delta \boldsymbol{f}$ is the residual force vector

Several different approaches can be taken for the definition of the mass matrix [19], however, in this thesis the following simple definition using the identity matrix will be used.

$$\boldsymbol{M} = \rho \boldsymbol{I} \tag{5.38}$$

Further, to reduce the bouncing, a damping coefficient c is introduced on the velocity at each step [21].

5.3 Example Procedures

5.3.1 Geometry - Patch

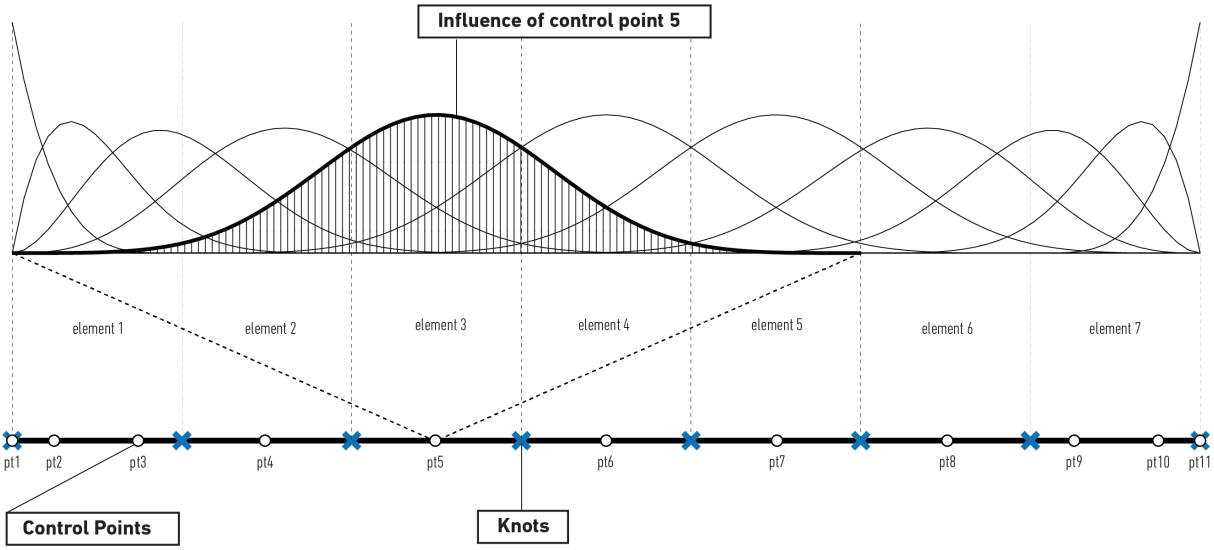


Figure 5.6: IGA patch used for the upcoming truss and beam problems

The following mesh is used for the patches. It's a fourth order polynomial with seven elements.

Points	Degree	Elements	Knots
11	4	7	[0 0 0 0 0 0.1429,0.2857,0.4286 0.5714 0.7143 0.8571 1 1 1 1 1]

Table 5.1: Geometry data for the 1D patch

5.3.2 The von Mises Truss

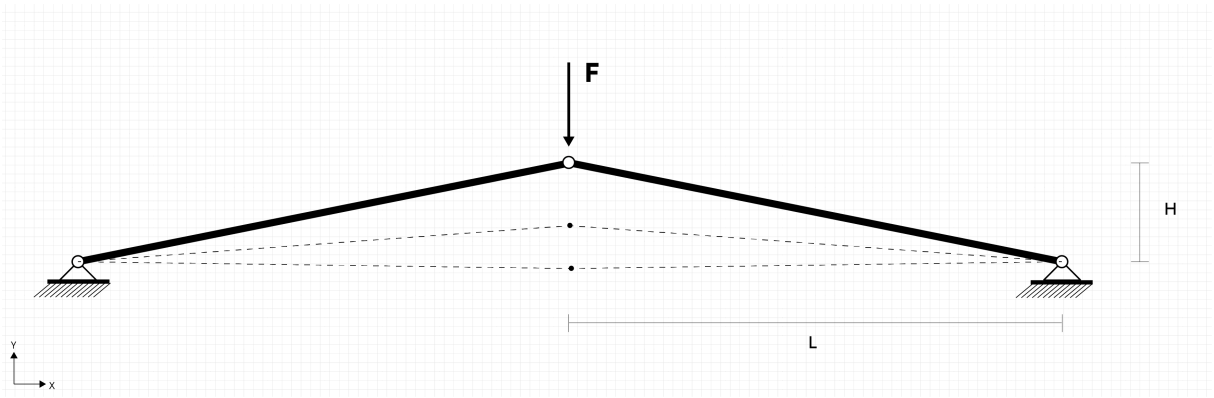


Figure 5.7: Illustration of von Mises Truss

The von Mises truss illustrated above is a classical example in illustrating geometrical non linear phenomena. This will be used as a test case using Newton Raphson as a solution scheme and some effect of the various approaches. This case comes with an analytical solution that can be used to verify the behaviour of the nonlinear solution model.
The analytical solution is given by:

$$F_t = -\text{EA } \sin\Phi_t \frac{l_t - l_0}{l_0} \tag{5.39}$$

where
- F is the force described as a function of the displacement at the peak point
- Φ is the angle between the truss and the x-axis
- l_t and l_0 denotes the length in reference and deformed configuration

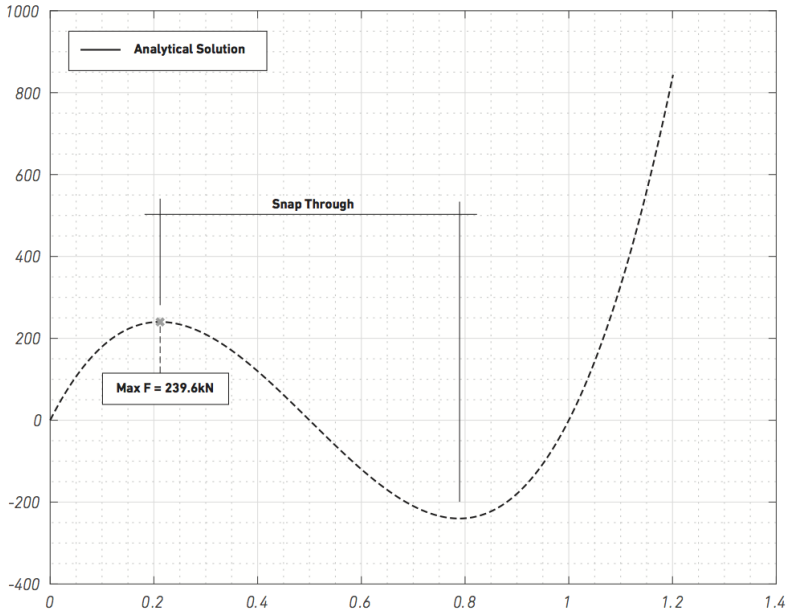


Figure 5.8: Analytical solution to the von Mises Truss

Figure 5.8 shows the analytical solution to the equilibrium of the von Mises truss.

5.3.2.1 Force Control

Solving this problem with force control is not ideal, as can be seen in the analytic solution, as the problem has a peak, and at the snap through the load need to be reduced.
The presented example has a max load of 239.4 kN. Below is a table comparing the performance of the different iteration schemes.

Load [kN]	Linear Elastic	Modified NR	Full NR
40	6	6	3
80	8	6	3
120	12	7	3
160	17	8	3
200	28	11	4
240	>100	>100	>100

Table 5.2: Number of iterations

5.3.2.2 Displacement Control

Switching to displacement control helps overcome the divergence problem found in the previous chapter. The same structure will now be analyzed, but instead controlling the displacement at the top.
Below the traced path of the snap through behaviour can be seen, along with the traced Load-Displacement curve. It should be noted that no iterations were necessary, but the equilibrium was found directly. However, this is due to the simple nature of the problem and this is not necessarily the case for all displacement controlled processes.

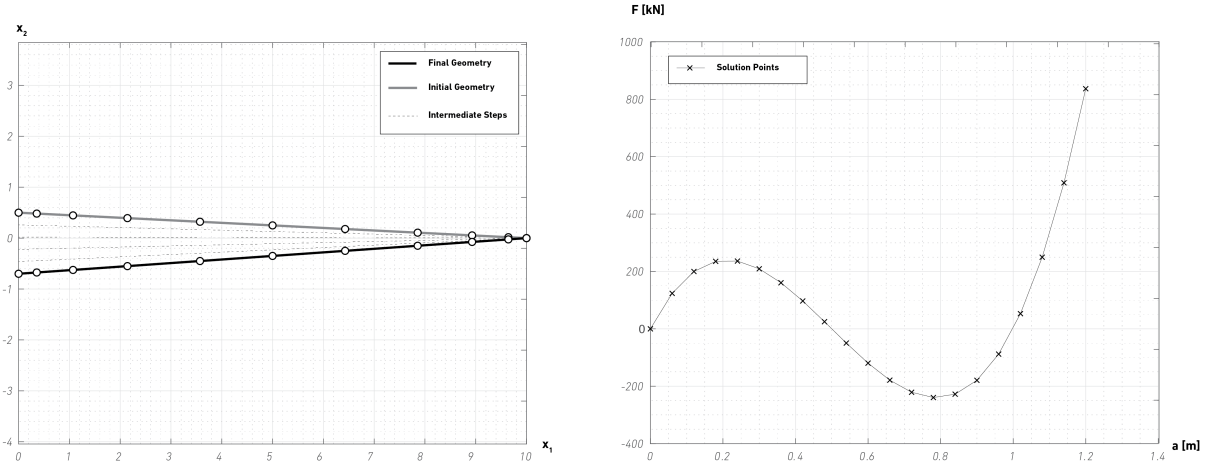


Figure 5.9: Displacements (Left) and Load-Displacement (Right) for the von Mises truss

5.3.3 Spatial Spline

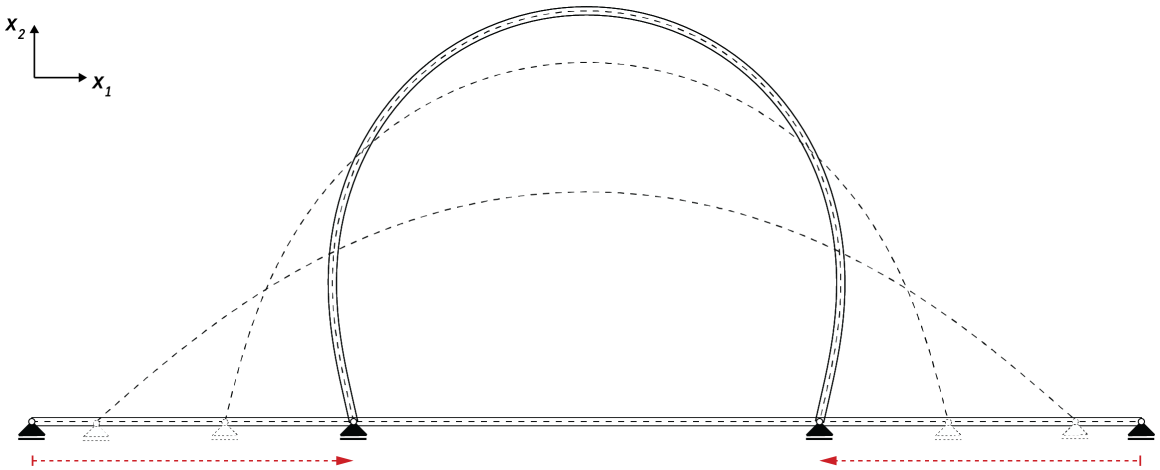


Figure 5.10: Setup for the elastica curve

The following problem will serve as a test case to test the various solution procedures. The problem is to solve the post buckling of a straight beam subjected to forces or translations at the end. This problem is called the elastica curve with a known analytical solutions. This problem combines both axial and bending action which gives a good basis for testing the procedure. The boundary conditions will defined as pinned at the ends.

For this case all three previously presented methods will be examined and their pros and cons will be discussed. The information produced in this case will set up the approach for the design tool in the next case. The mesh used is the same as the one used in the previous case, and the beam is given the following physical properties.

h [m]	**b [m]**	**E [MPa]**	**Length [m]**	ρ
0.08	1	12 000	10	1/1000

Table 5.3: Physical properties of the spline

The parameter ρ indicates the imperfection used. The imperfection means a deviations from the perfectly straight line. The mid point is therefore found at a height of 0.01 [m]. This is to control the buckling procedure and help initiate it as the buckling of a perfectly straight beam is difficult. As mentioned, the beam will be solved using a both a load controlled and a displacement controlled matrix based procedure along with a dynamic relaxation. It will focus on comparing convergence aspects to highlight the robustness of the procedures.

For reference the critical load according to Eulers buckling cases will be used to compare the behaviour of the beam. The beam corresponds to the second buckling case of Euler and the critical load is given by.

$$P_{cr} = \frac{\pi EI}{L^2} \tag{5.40}$$

Which gives a critical load of:

$$Pcr = 50.532kN$$

5.3.3.1 Force Control

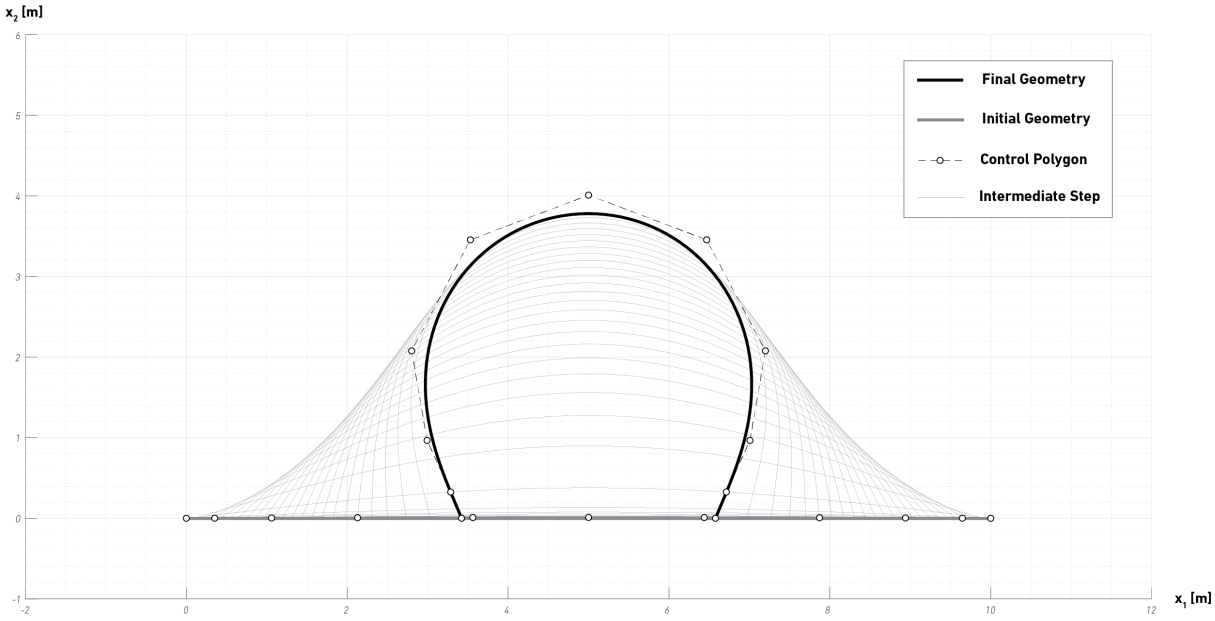


Figure 5.11: Resulting geometry from the load controlled procedure of the Spatial Spline

Setup / Input
Here follows the force controlled procedure. The beam was loaded horisontally at each end, thus pushing the end points towards each other. The folowing procedure traces the behaviour up until a load close to the peak load. For iterations, the nonlinear effects were neglected, as the beam clearly finds itself in two rather constant states of pure axial stiffness and pure bending.

The genral input was as follows.

Peak Load [kN[	Num Steps	Tol	max Iter	Num Gauss Points
116	2000	0.001	100	4

Table 5.4: Analysis setup for the force controlled solution

Results

The final result can be seen in Figure 5.11. It shows the initial geometry along with the final and some of the intermediate steps. What can be seen from this procedure is that the algorithm struggles to overcome the swift change in stiffness that occurs at the point of buckling, even to the extent that 100 iterations is not enough. The process, however, recovers and pick up equilibrium again. It can be seen in Figure 5.13 that as the process approaches thee end, the iterations go up. This is due to the load displacement curve becoming less inclined as it approaches a peak load. The onset of buckling comes very close to the one expected from the Euler case. the slightly premature start is most likely due to the small initial imperfection given to the beam. As can be seen, using a constant load increment, a small step was necessary due to the onset of buckling. This resulted in a lot of steps.

Here, a more sophisticated stiffness matrix might cause numerical instability as the load approaches the buckling load. The process was found more robust neglecting the nonlinear stiffness for iterations.

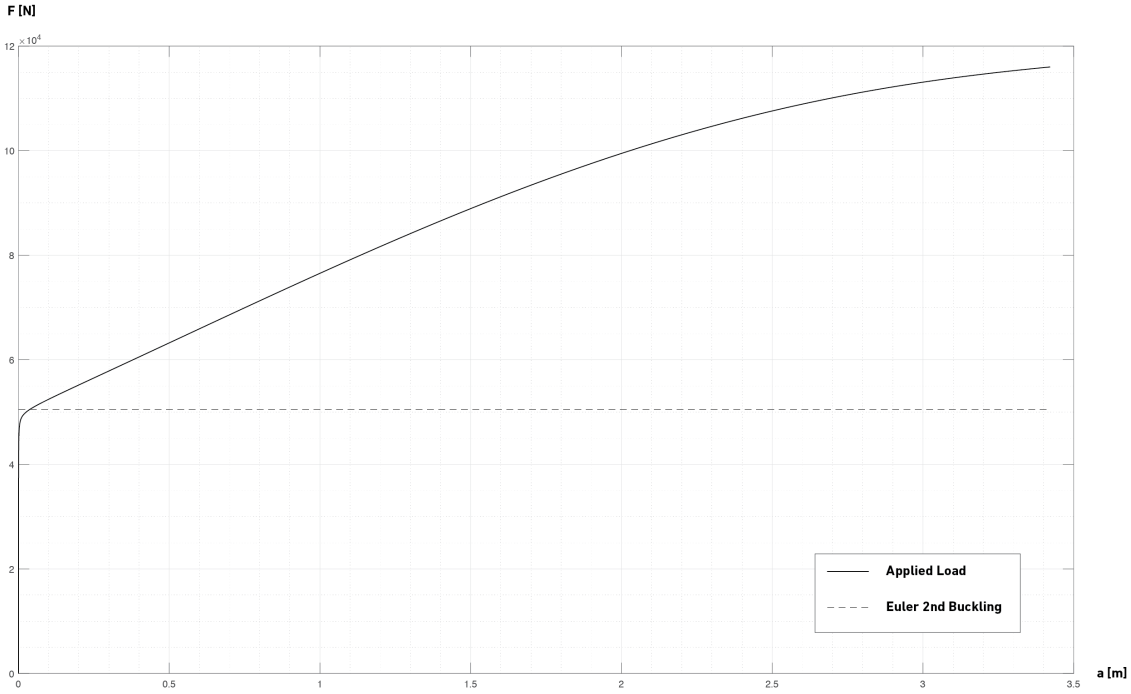


Figure 5.12: Load-Displacement curve for the first DOF for the force controlled process of the spatial spline (Horizontal at node one)

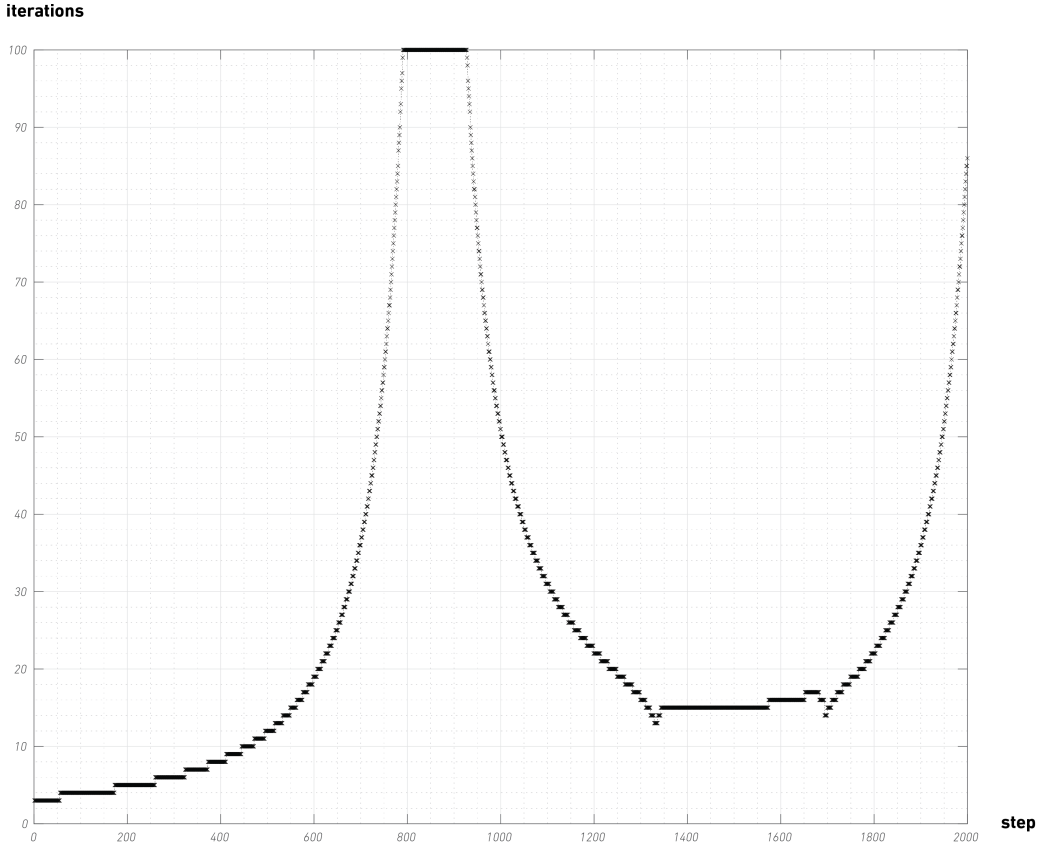


Figure 5.13: Number of equilibrium iterations at each step for the force controlled solution to the spatial spline

5.3.3.2 Dynamic Relaxation

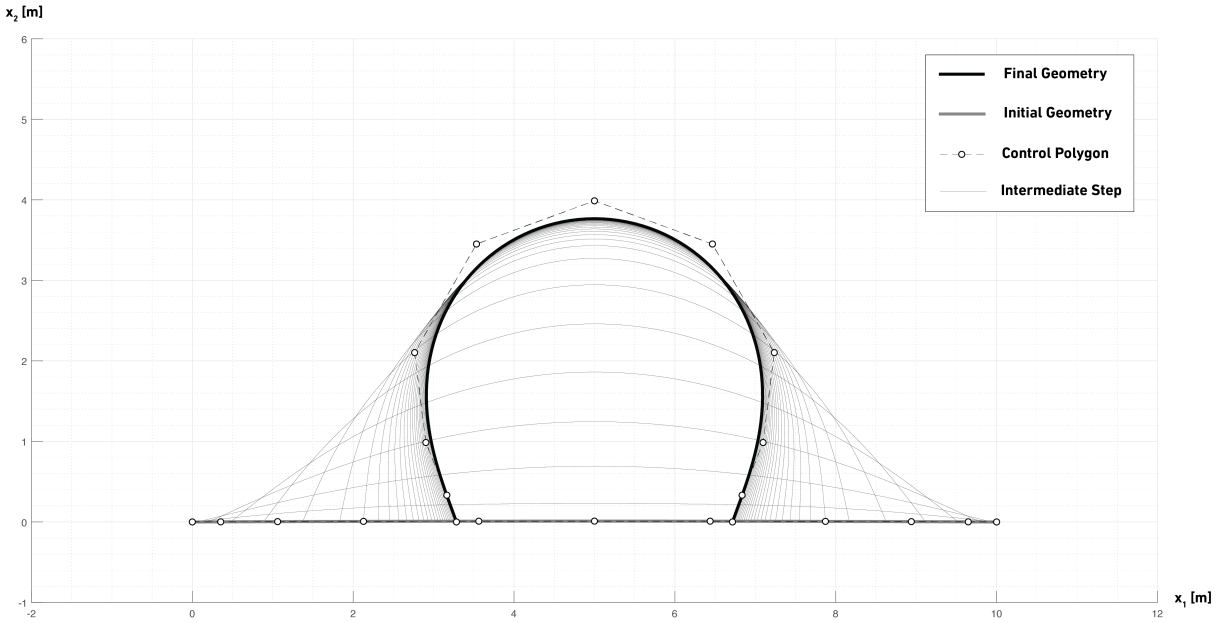


Figure 5.14: Geometry of the deformation from the dynamic relaxation process

Setup / Input
The following describes the input for the dynamic relaxation. The beam was loaded similarly to the previous case except for two things. Firstly, as mentioned, the whole load was applied from the beginning. Secondly, an additional load perpendicular to the beam was used in the first steps of the process. This was to help the buckling as the very small deviation from a straight line made it hard for the buckling to start. This load was to help induce some acceleration. For damping a kinetic damping was used.

In the general input was as follows:

Load [kN[	**Num Steps**	Δt	**c**	ρ	Num Gauss Pts
116	6000	0.0009	0.99	1	4

Table 5.5: Analysis setup for the dynamic relaxation solution

Results
Figure 5.14 shows the final geometry together with the initial and a few of the intermediate steps. It can be seen that it approaches the equilibrium in a different way compared to the previous procedure, with many intermediate steps close to the final. Further the first steps show the application of the perpendicular load as it doesn't follow the form associated with a pure buckling. (Much flatter in the middle).

The dynamic relaxation procedure proved hard to control. Small deviations in time step etc could cause large problems for the convergence. Especially with the stiff region the time step had to be very small. In Figure 5.16 the speed during the process is shown. Before buckling oscillations can be seen, which are due to the large stiffness together with large load, which occurred at the ends. However, as the beam started to buckle, it can be seen that the process started to run smoothly.

However, the convergence was very slow as it approached the equilibrium mode.

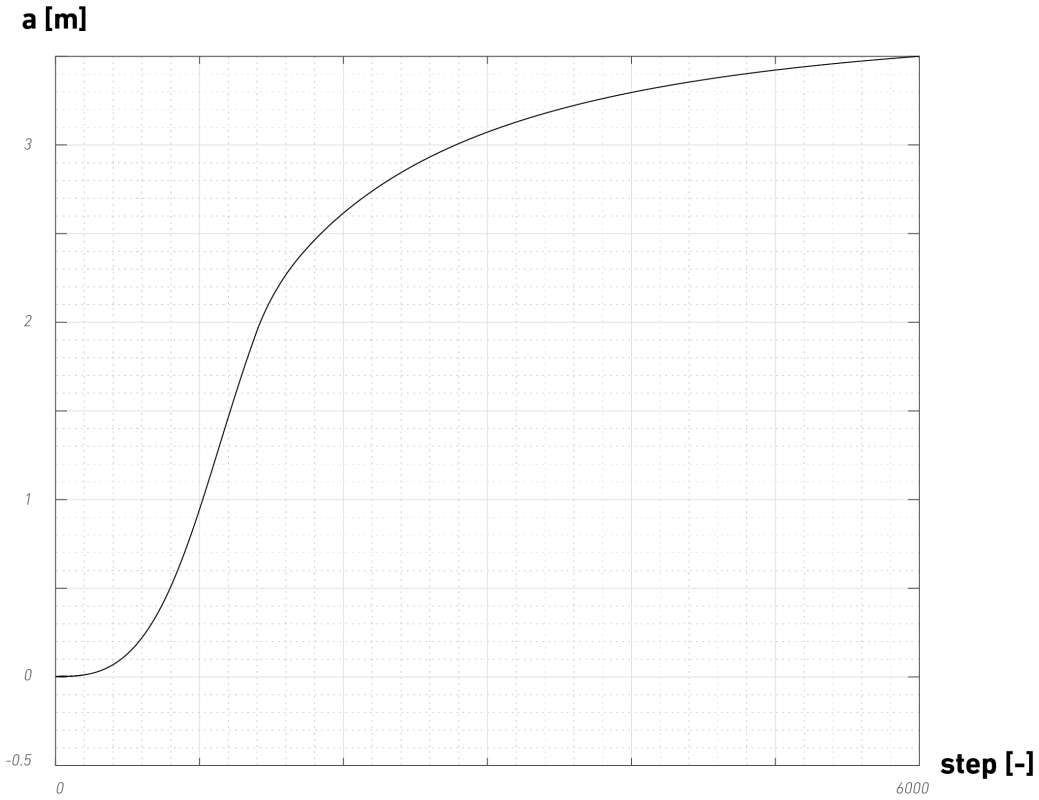


Figure 5.15: Displacment over the time of the process at dof 1 for the dynamic relaxation solution

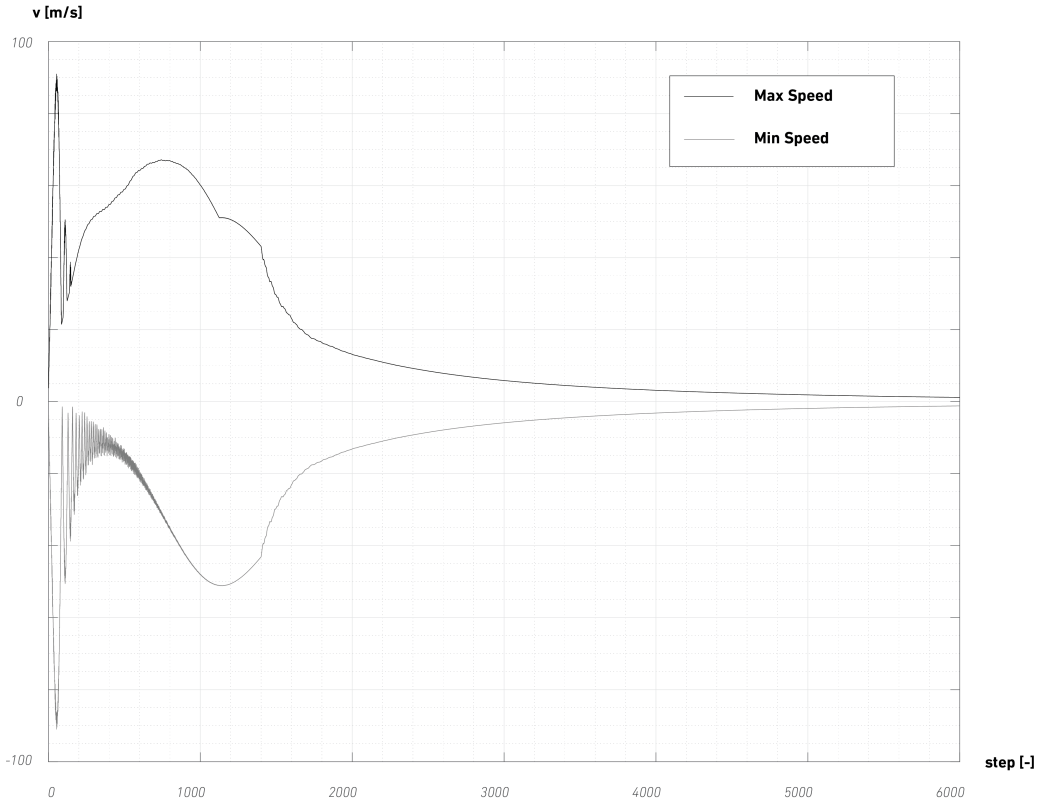


Figure 5.16: Max/Min speed of the model over the time of the process for the dynamic relaxation solution

5.3.3.3 Displacement Control

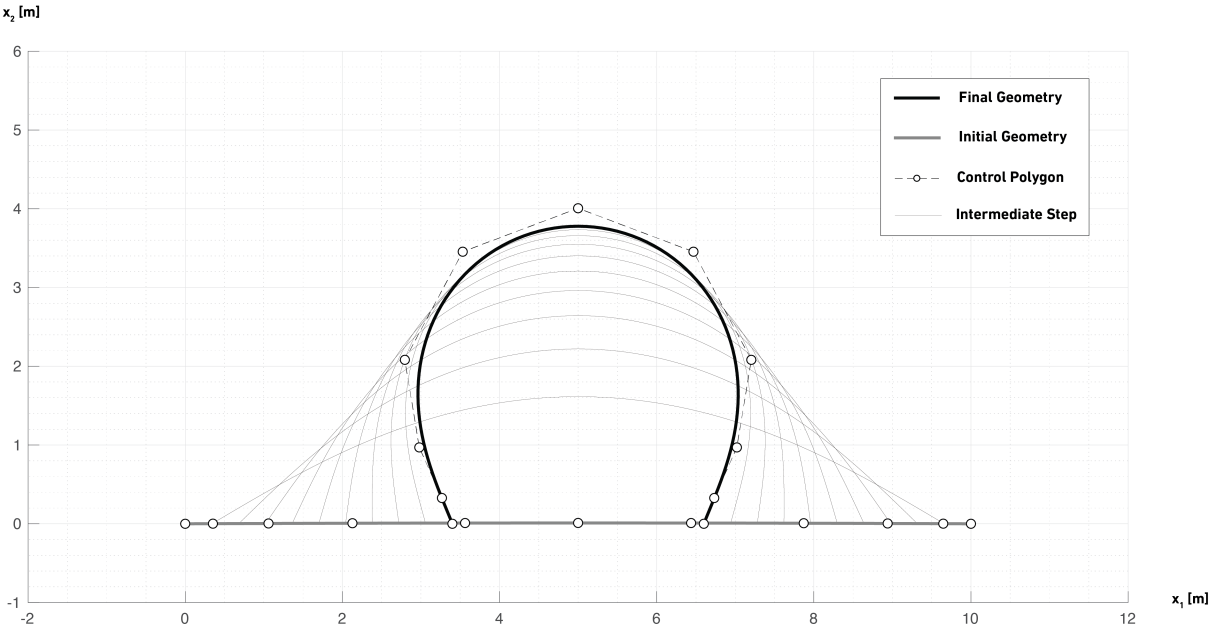


Figure 5.17: Geometry of the spatial spline from displacement controlled process

Setup / Input

The final procedure is similar to the first one, with the difference that the displacements of the end points is controlled instead of the load. The maximum displacement was chosen to correspond, roughly, to the displacement associated with the peak load from the first procedure.

The input was as follows.

Prescribed Displacment [m]	**Num Steps**	**Tolerance**	**Max Iter**	Num Gauss Pts
3.4	100	0.001	100	4

Table 5.6: Analysis setup for the displacement controlled solution

Results

The resulting geometry can be seen in Figure 5.17. The process showed almost no problem with either approaching a max load or at the onset of buckling. This lead to no problem and convergence at only a 100 steps compared to the 2000 needed for load control. The convergence of the process was also rather constant as the amount of iterations needed kept within the range of 8-12 as can be seen in Figure 5.19. The load displacement curve is a bit more coarse in figure 5.18 compared to Figure 5.12 which is due to the decrease in steps. However, the general form follows the same shape and corresponds well to the predicted buckling load.

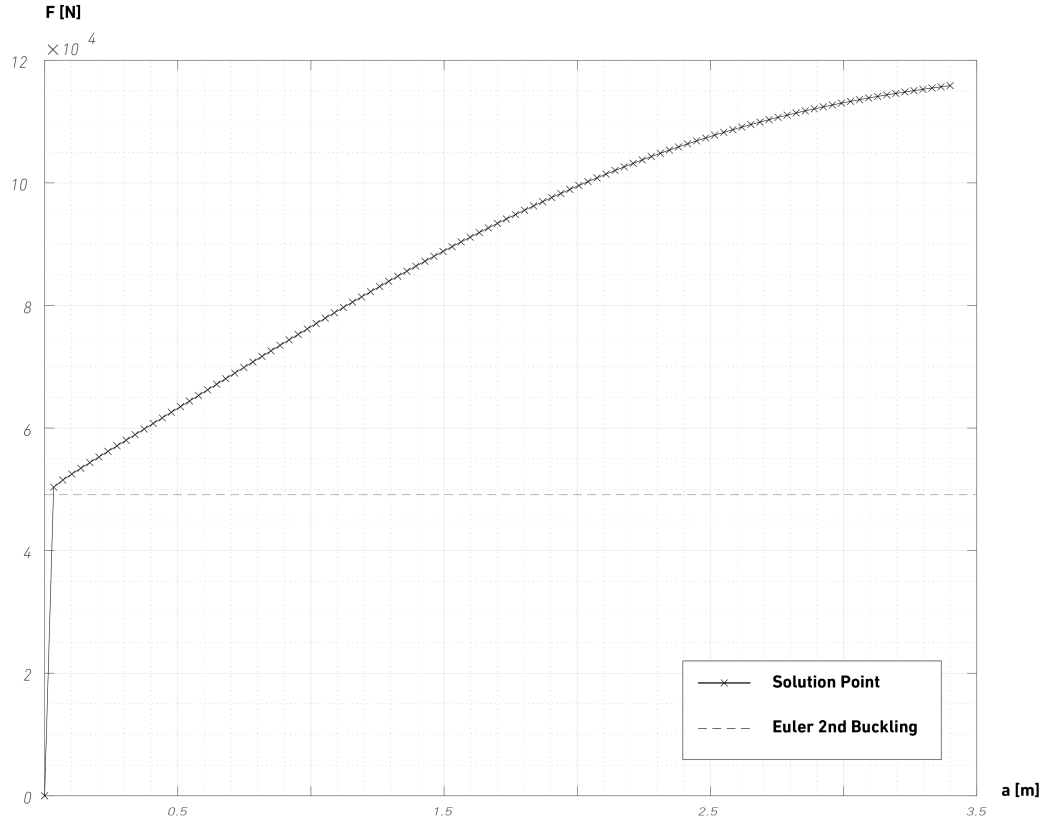


Figure 5.18: Load-Displacement curve for the first DOF for the displacement controlled solution

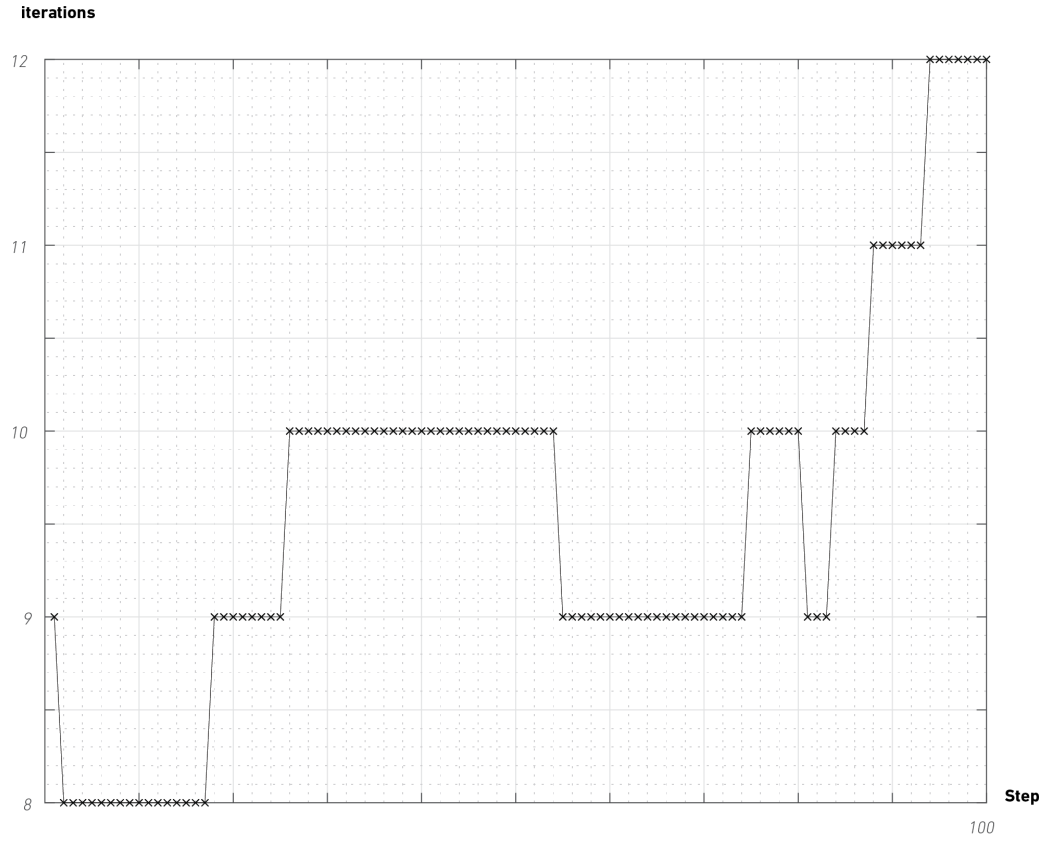


Figure 5.19: Equilibroium iterations at each step for the displacement controlled solution

5.3.4 Sequential Forming

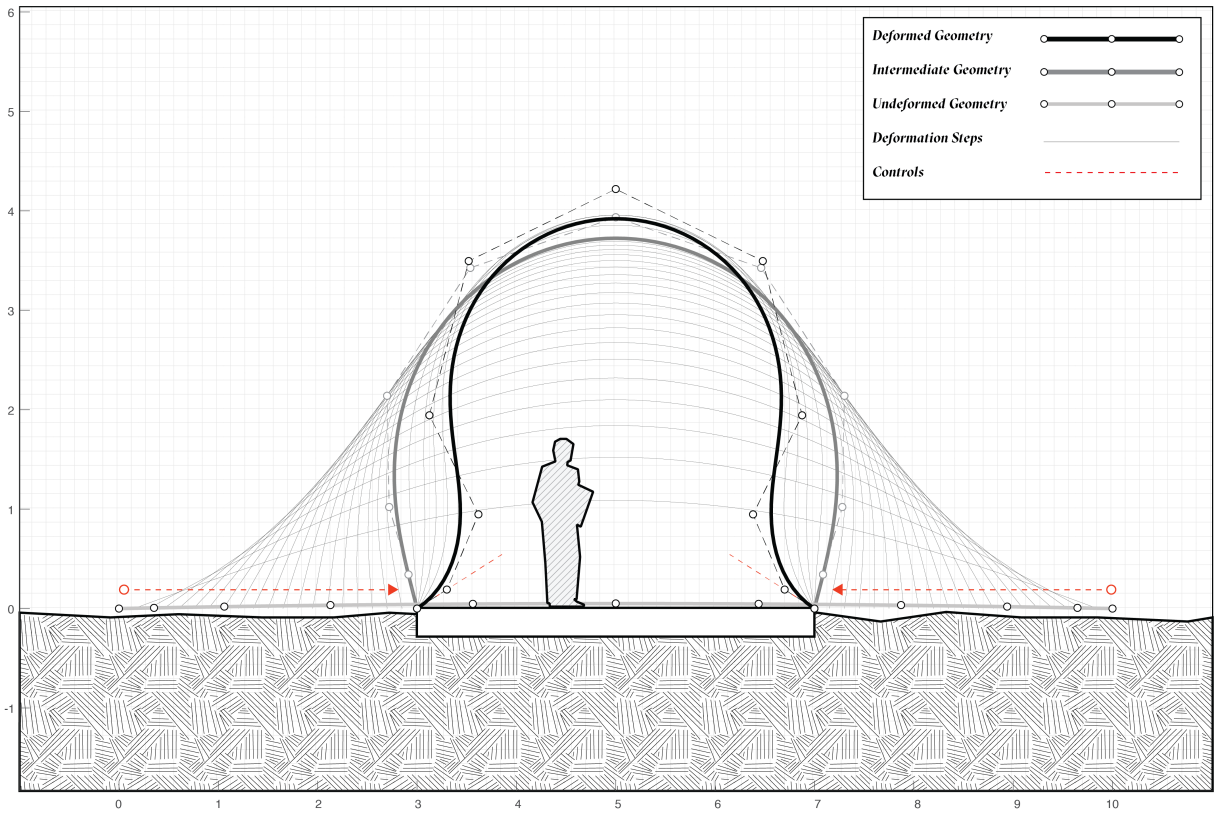


Figure 5.20: Example of a sequential process changing first positions and secondly angles at both ends

The following procedure shows a sequential process. First the end points of the spline are moved. In a second step the angle of the spline at the base is changed. This shows that the model can deal with sequential steps. Reference to an actual construction process.

5.3.5 Conclusions

Given the nature of the selected problem, displacements should be used to control the form. The application of dynamic relaxation showed some issues and with the time frame in mind, the choice was made to go for the more robust displacement controlled matrix solver. However, a lot of the problems associated with the presented solution using DR could most likely be solved by implementing more sophisticated damping, use of mass or time integration, although it was left outside of the scope of this thesis.

6
Design Tool and Analysis Procedure

The theory and procedures presented in the previous chapters were further exported into a design software. The following chapter will describe the software implementation along with a presentation of the functionality and expected workflows. With the solutions for the beams discussed and presented, the tool focuses on the implementation for shells in 3 dimensional space.

6.1 Design Software Framework

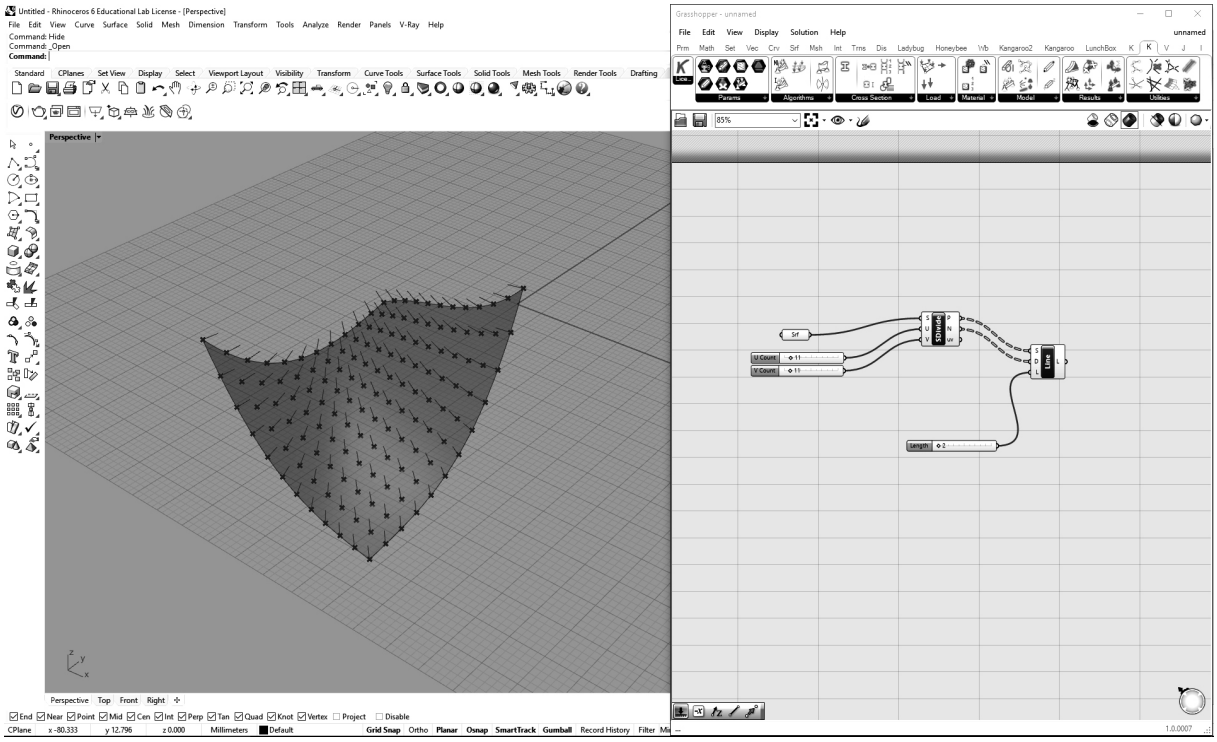


Figure 6.1: Example of Rhinoceros/Grasshopper interfaces

As previously mentioned, a key aspect of this thesis is to support the integration of design and analysis. The commercial analysis software and design software are distinctly separated, which provides a large bump in an integrated process. As one of the main motivations for IGA is the common framework, the processes should also run in an integrated platform. As the aim of this thesis is to embed analysis within a design process, the analysis procedures should therefore occur within a NURBS based design software. The most common, NURBS based, 3d modeling software within the AEC industry today is Rhinoceros by Robert McNeel and Associates. To make the simulations available it should be integrated into this, design native framework. Rhinoceros provides a robust framework for the manipulation of NURBS and the methods described in this thesis will therefore be used to further extend the possibilities of NURBS based design. Through the Rhinoceros environment, the analysis can find it's part within a large set of tools.

Rhino is a software open to additional third party plugins, which has led to a multitude of various plugins which has helped advance the popularity of the software. Another important step in the rise of popularity for Rhino was the introduction of the plugin Grasshopper. In 2009 David Rutten developed a plugin to Rhino called grasshopper. This is a visual scripting environment used to control rhino. It provides, more or less, the same methods and commands available in Rhino, but as a series of components that perform specific tasks within the rhino software. This can be used to model parametrically in by assembling flow charts describing geometric relationships. By programming form as a logic, it can then easily be adjusted and manipulated by changing a set of defined inputs or parameters. Grasshopper launches as an external window providing a canvas for the components. Figure 6.1 shows an example of a set of components used to draw a set of lines normal to an input surface.

One of the many reasons this plugin has been widely successful is its extendability. Given the component logic it is very easy for third party actors to extend the software in various ways. Due to its widespread popularity and it's simple extendability, this will serve as a perfect framework to implement the previously described methods. The graphical user interface is set and a plethora of additional components will be available to integrate into the workflow. Given the nature of analysis with a series of numerical inputs that will be tweaked back and forth, Grasshopper serves as a good framework for implementing the Isogeometric analysis.

6.2 Software Implementation

6.2.1 Implementation Logic

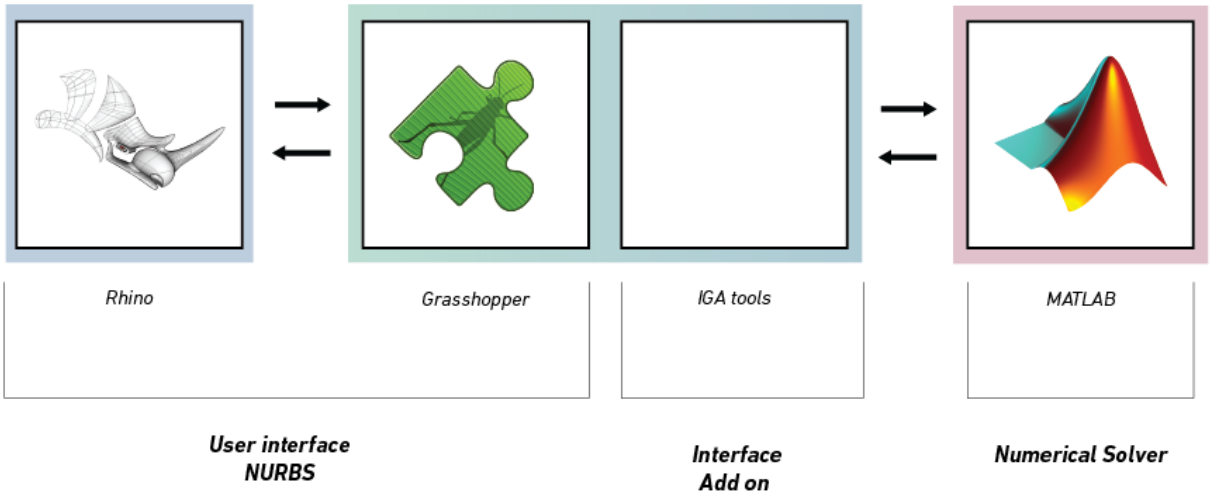


Figure 6.2: Elements of the implementation and their relationships

Background
Extending the grasshopper framework with additional components is made very easy as McNeel provide a template for the development of components. This is offered through visual studio and the main language for implementation is C#. The template lists the methods necessary to genereate input/output etc. The main issue is to make the numerical methods available to the grasshopper plugin. In general, this could be done in two ways. It could ether be done through developing a library in some object oriented code to be used by the plugin, or it could be done by calling the MATLAB code already developed. Writing a custom library in C# or C++ would be ideal, as it would free the software from the dependency of MATLAB as an external actor. However, given the timeframe of this thesis, the quicker method is to simply call the MATLAB files.

Rhino and grasshopper serve as the geometry framework and overall user interface. The MATLAB routines are used to solve the nonlinear IGA problems. This includes things such as functions the compute

the stiffness matrix for a patch or methods to solve the system or compute stresses. In order to activate these tools within the more user and geometry friendly framework, a set of additional components are developed for the grasshopper interface. These components serve to assemble the input from grasshopper into a certain structure that can be used to send data to matlab, to perform some calculations, and be read back into the workflow and can be used within the design process in a seamless way.

MATLAB Runtime
MATLAB is a software, that comes with a graphical user interface and a wide range of tools. However, the central aspect of it is to operate upon and handle matrices. This is hinted in the name, which stands for MATrix LABoratory. This means that it processes data structures mainly consisting of n-dimensional arrays or cells of doubles. As the main aim of this thesis is to operate within a design native context, the interaction with the calculations should only be done by passing geometric data and information to the grasshopper components and getting easily accessible data back. Therefore, MATLAB should act hidden in the background and only perform calculation remotely without the application of the userinterface or software.

For this purpose, Mathworks provide something that is called MATLAB runtime. That is a standalone set of shared libraries that enables the execution of compiled MATLAB applications or components. This enables .m libraries to be compiled and run remotely from other applications. A specific complier can be used to compile a set of .m files to be accessed from multiple different languages and environments. When used together, MATLAB, MATLAB Compiler, and the MATLAB Runtime enable you to create and distribute numerical applications or software components quickly.

The .m files developed for the Isogeometric analysis can then be compiled into a dll (Dynamic Link Library) that can be called directly from the grasshopper components written in C#. The interchangeability is done using a wrapper class called MWArray (MathWorksArray). This is a class that makes the .net arrays readable for the calculation engine.

In this thesis, the language C# is used to write the components. The .m files are then compiled into a wrapper dll that offers them to be called from C#. Through this a specific method can be called from C#, run the matlabs engine in the background, and then have the information passed back to the c# application. The data transfer is enabled through a interface class called MWArray which stores the data so that MATALB can read it.

6.2.2 Software Structure

To deal with the passing and structuring of data, a structure to be used and passed between the components was needed. Below follows the key classes used to structure the data from Rhino into an IGA model that could be composed into the matrices that MATLAB then solves. This provides the base for the additional interface embedded within the Grasshopper environment.

The key classes are:

IGA_Model
The model stores all the data and is the central hub of the analysis. The model assembles all the patches, couplings, supports and loads and keeps track of the global system. It also stores the information about the current state of the model (time t) regarding any previous analyses. The model is responsible for associating points with dofs and assembly of internal force vectors and stiffness matrices. Further, the model is responsible for computing itself, thus, calling the necessary MATLAB functions.

IGA_Patch
The patch is the main structural component, created from a geometric surface (generated in Rhino/-Grasshopper) and a set of material and thickness properties. The model can be made up of multiple patches, but each patch behaves independently of all others unless specific means are taken. The patch consists of several elements and assembles a stiffness matrix w.r.t itself. In this application the patch can not be trimmed and is parameterised from 0 to 1 in both u and v. The patch has a reference geometry and a current geometry, where the reference represents time 0 and current represents time t. The patch keeps track of it's local connectivity as it's made up of several elements. The user deals in patches, not

in elements.

IGA_Support
The supports are used to prescribe boundary conditions, either as zero or some numerical value representing a prescribed displacment. The support is an abstract class with two implementations. Either enforced on a point on the surface, done through the use of Lagrange Multipliers [ref section BC]. This is the appropriate way to constraint a physical point. However, it can also be enforced directly on a dof. It should be noted that his will not constrain a material point as the controlpoints most often are not found on the surface. This should be used to constraint rotations without introducing internal supports, for example as a fixed connection.

IGA_CouplingConstraint
Can be used to constraint a point on one patch to a point on another to link the behaviour of two patches. The patches only interact at given coupling constraint. The coupling constraint operates on material points through the use of Lagrange multipliers. Defined by a point of action and a set of prescribed dofs and values.

IGA_PointLoad
Applies a load to a point on the surface. Applied similarly to the material point support. Deifned by a point of action and a load vector. Point load is applied as a set of equivalent nodal loads through the basis functions.

IGA_Material
The material model can be defined as isotropic or orthotropic. The isotropic material is defined using solely a Youngs modulus and a poisson ratio. The orthotropic material is defined using one Youngs modulus and one poisson ratio for each direction. It also needs a material direction to use as the primary directio (u or v). The second direction will be defined as perpendicular to this one.

IGA_CroSec
The cross section is either given as a constant or mapped as described earlier. The mapping is done by specifying a reference surface. The height from the reference surface (z-value) is used as a thickness map relating to the uv parameterisation.

IGA_Result
Each model is associated with a results class. This keeps track of the current state of the model and makes sure any previous analysis is incorporated into a subsequent one. It also stores info about each step in the nonlinear process.

IGA_MATLAB
The final essential class is the MATLAB class. This class provides the exported .m functions and is used to call them remotely.

6.3 Functionality

With the practical aspects of the tool described the following chapter will go on to describe the userside of the tool. It will show the general workings and intended workflow of the process.

6.3.1 Overview

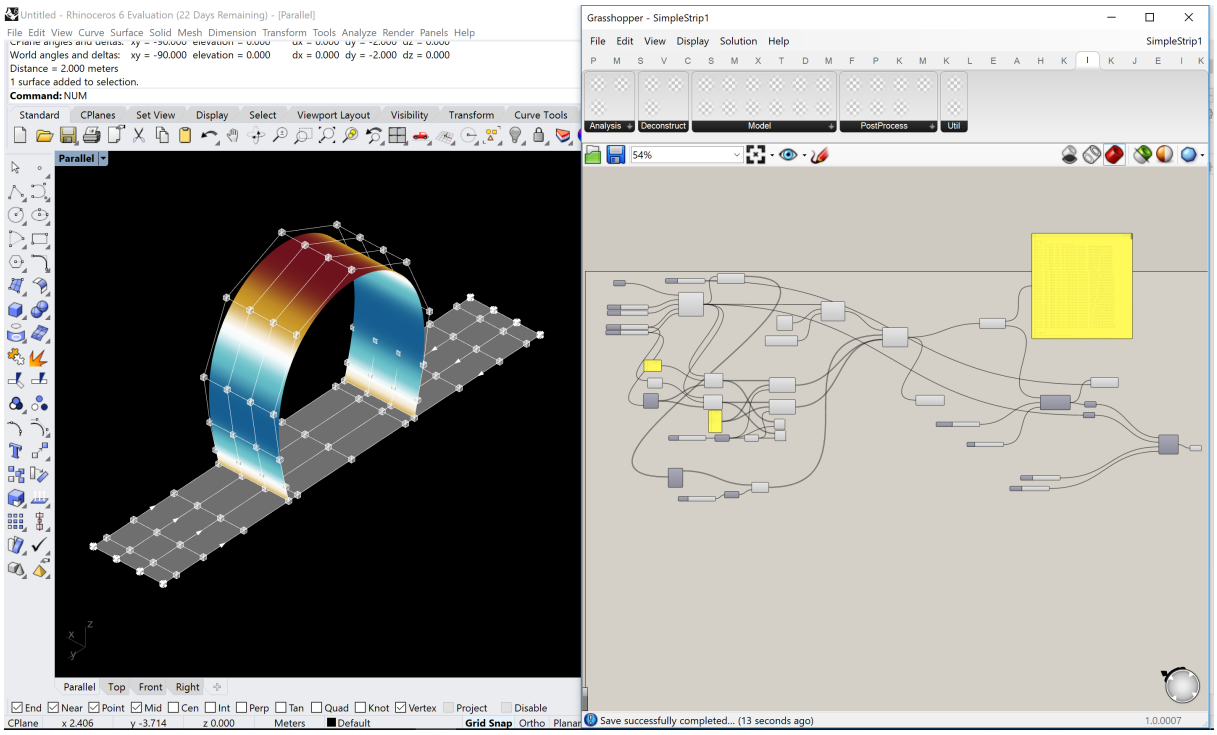


Figure 6.3: Example of the implementation used to solve a simple elastica shell

As previously mentioned, the workflow is modelbased. The models are assembled from custom objects that in turn are assembled from numeric or geometric input. On those models the specific tasks can then be performed. For each step or component, information is added to or extracted from the model. The approach to separate a lot of behaviour so that the work can be customized. Each component only provide a very specfic task. This benfits through the potential to tailoring what's calculated, thus making a faster workflow. The computations are done by each component w.r.t what the component does. For example the stresses are not calculated until the stress component does so.

The components that deal with the structure described earlier is categorised into 5 categories. The components are used to assemble data from Rhino into the structures described in the previous chapter.

The following categories have been established.

- *1. Model*
- *2. Analyse*
- *3. Post Process*
- *4. Deconstruct*
- *5. Utilities*

These categories provide a set of tools for each step of the process. The first category, *model*, is used to define elements, materials, loads and cross sections and assemble a model. The analyse tab, component that will solve the assembled models and provide them with results. The analysis components only return a model with added results. These results can then be visualized or extracted using the post processing tools to show out of balance forces, stresses or get the deformed geometry. In case further analysis should be made the deconstruct tab provides tool to extract data from the model. This can then be used to assemble further models with new loads or boundary conditions.

Figure 6.3 shows an example. The screenshot shows a simple strip subject to prescribed boundary conditions. The undeformed geometry is shown in grey and the deformed geometry is shown in a color plot. The plot shows the absolute distance that a point on the surface has deformed. The control polygon is plotted as boxes and lines.

The components on the following page are added to the grasshopper interface to enable the IGA calculations. The tools generate object following the structure described earlier. These are used to assemble and calculate the models. The workflow can be customized into any assembly or sequence of these components.

6.3.2 Component Library

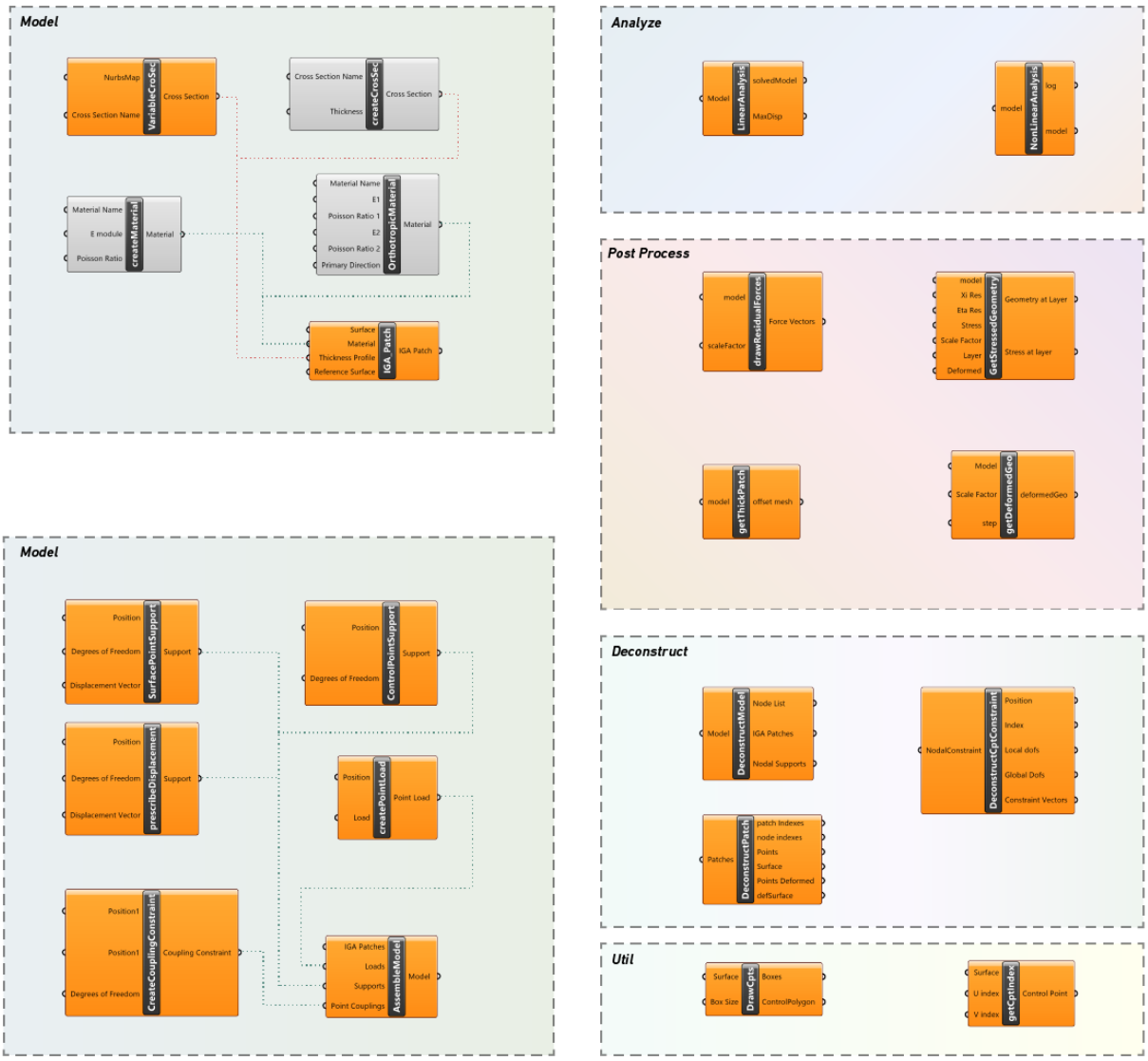


Figure 6.4: Developed component library

6.4 Workflow

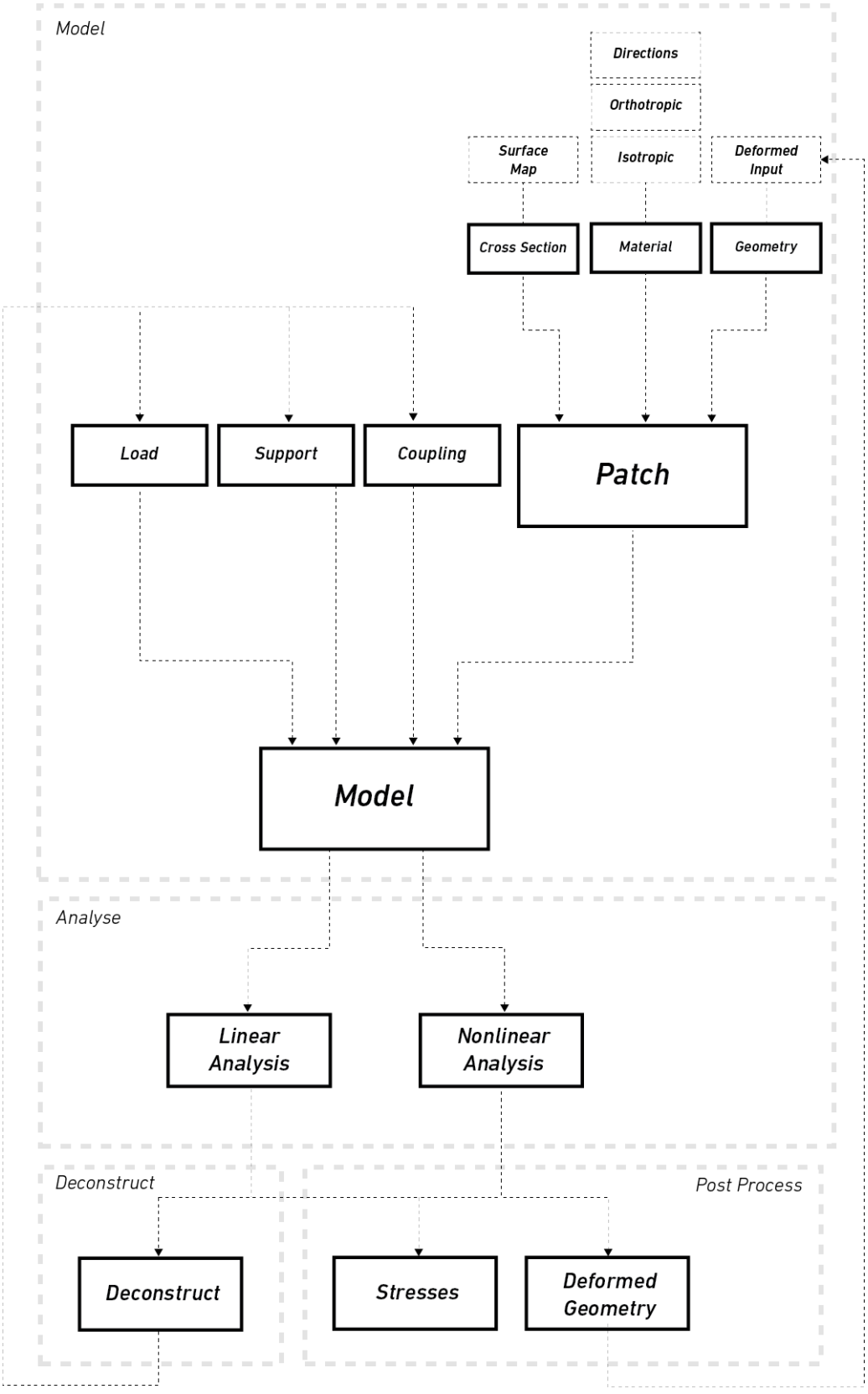


Figure 6.5: Intended flow for the components

Above a generalised flow of information is shown. It shows how patches, loads, supports and couplings assemble to form a model which is then further processed. The solid boxes denotes components whereas the dashed outlines denotes specific inputs. This outlines the workflow that will be further elaborated on w.r.t the *forwards prediction* and *backwards tracing* processes.

The difference between linear and nonlinear analysis is that the linear analysis only performes one linear step based on the current geometry. It does not perform any equilibrium iterations, but it does account for the state of stress.

6.4.1 Forward Prediction / Backward Tracing

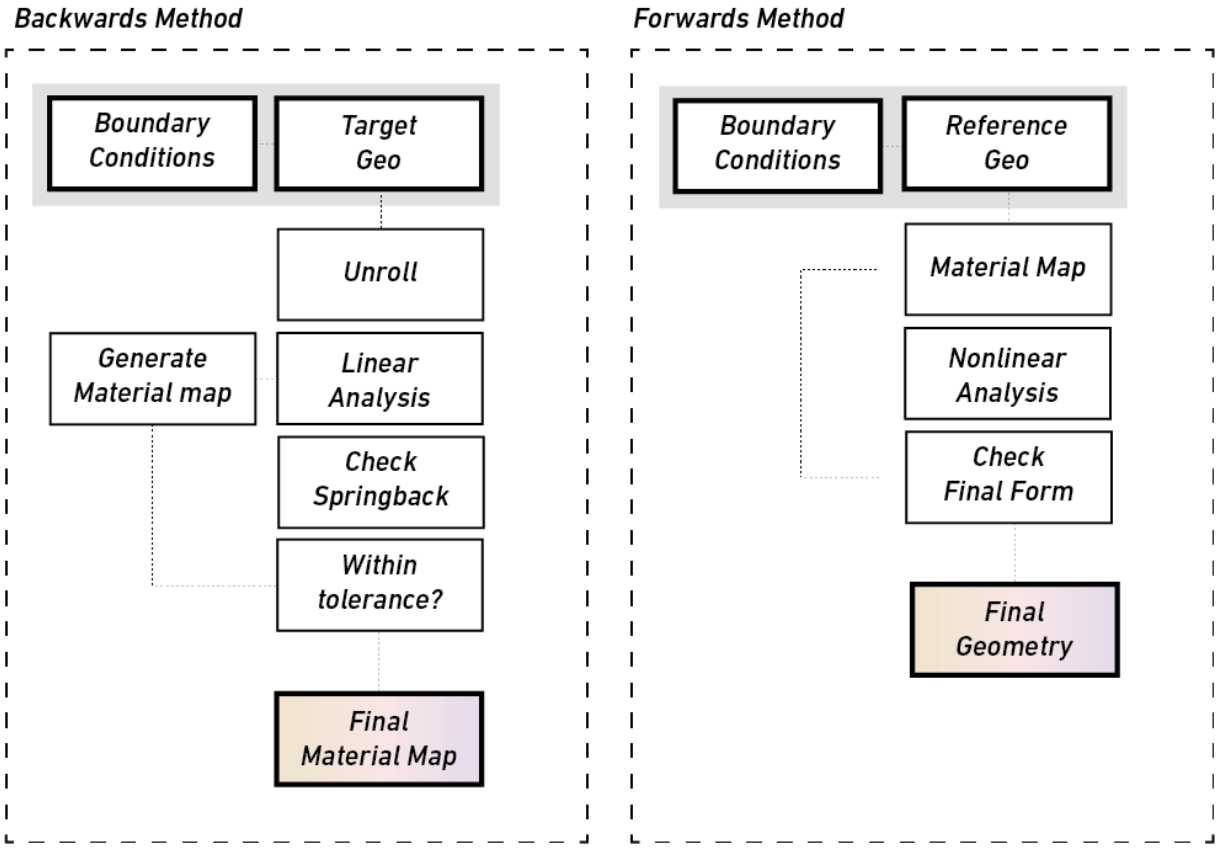


Figure 6.6: Workflow for the backwards tracing (left) and forwards prediction (right)

As stated earlier the aim is to provide one forwards predicting, and one backwards tracing approach. The process is concerned with four overarching input/outputs. These are the following:

- *Reference geometry (time 0)*
- *Thickness Distribution / Stiffness*
- *Boundary Conditions*
- *Deformed Geometry*

With the forwards prediction, the process follows that of construction. The reference configuration is then given along with a thickness distribution (Cross section). From a set of prescribed end conditions, a deformed geometry can be found. As the constraints that the geometry is supposed to fit most likely is given, the main alteration is done through manipulating the thickness distribution. This means checking final form along with making sure stress levels are within the acceptable range.

This is done by defining a patch with a set of prescribed displacements. From this a complete nonlinear analysis can be performed to trace the behaviour given the specified boundary conditions. The final outcome of the process is then the final form.

The *backwards tracing* starts from a different point, were a precedent surface is given. It should be stated that this would mean keeping in mind the geometric consideration presented in chapter [2], to make sure the intended design is kinetically viable without the need for absurd stresses. In this case the desired final outcome is then input together with a set of fixed boundary conditions. The aim is then to find the appropriate thickness distribution to associate the desired geometry to an equilibrium with respect to the initial geometry.

This can be done by unrolling the starting geometry. Unroll is a function provided by the rhino geometric framework which creates a flat approximation of a curved geometry. Given the desired geometry and the unrolled geometry, a linear analysis can be performed. When both deformed and reference geometries are present, the analysis will start from the reference at time t. This gives information about the spring back of the desired geometry. The aim of the process is then to adapt the thickness distribution. The thickness distribution can be generated through a surface map as defined earlier. This allows for full control over the complexity of the mapping. To find an equilibrium, the map can be modified. This can be done by manually adjusting, or by using other tools available within the framework, such as an evolutionary algorithm. By controlling the thickness, one can then manipulate the equilibrium to fit the target geometry.

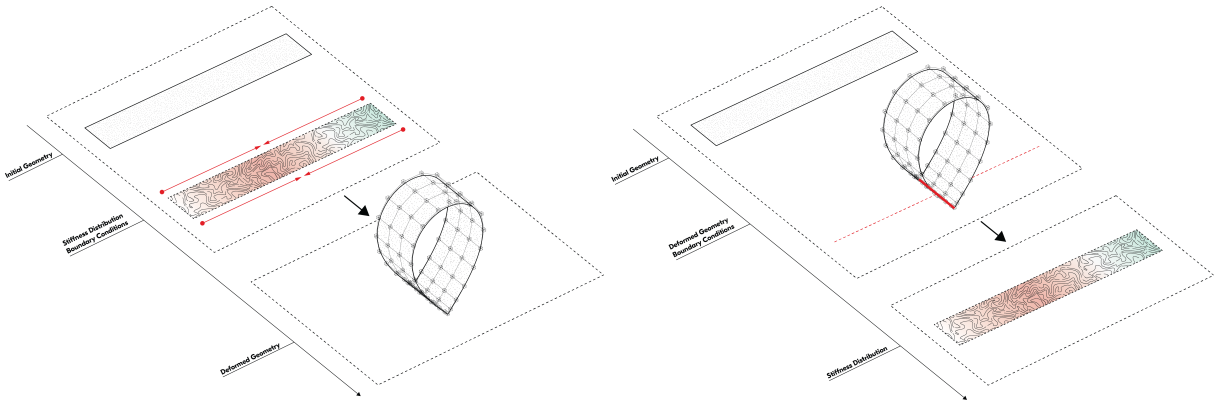


Figure 6.7: Input and results for the forward prediction (left) and backwards tracing (right)

6.4.2 Combined method

With the two approaches defined, there is nothing preventing the thickness map from the backwards tracing to inform the forwards prediction. This gives information about how close the final geometry will be. Here one can also negotiate the complexity of the thickness description with how close to the target that is necessary to be.

A complete workflow to recover a target geometry can then be described using the following step by step procedure shown in the following algorithm 4.

Algorithm 4 FORM FINDING PROCESS

```
1: procedure Backward Tracing(a)
2:     Model target geometry at time t
3:     Define a thickness parameterisation
4:     Apply planar map to target geometry to establish reference configuration at time 0
5:     Define fixed DOFs
6:     while deformation > tol do
7:         Solve next step from time t
8:         Update thickness distribution  ▷ In this thesis this iteration is done with a genetic algorithm
9:     return Thickness map
10: procedure Forward Prediction(b)
11:     Reference configuration at time 0 from previous procedure
12:     Define boundary conditions (Prescribed displacements)
13:     Thickness map from previous procedure
14:     Solve nonlinear displacement controlled procedure
15:     return Deformed geometry
```

7

Case Studies

With the tool and processes presented, it will in this chapter be tested on a few simple studies. These studies take off from the spatial spline presented in chapter 5 but now extends to study the effect on sheets and further applies the extended material properties through a custom thickness mapping. It makes use of the grasshopper components described in the previous chapter.

The study will start with the forwards tracing of a simple sheet with a set of different thickness distributions. From this a backwards process of recovering a thickness distribution will be illustrated and these two methods will then be combined to reconstruct a design geometry and find the final form considering the stiffness distribution.

7.1 Forwards Process

7.1.1 Setup

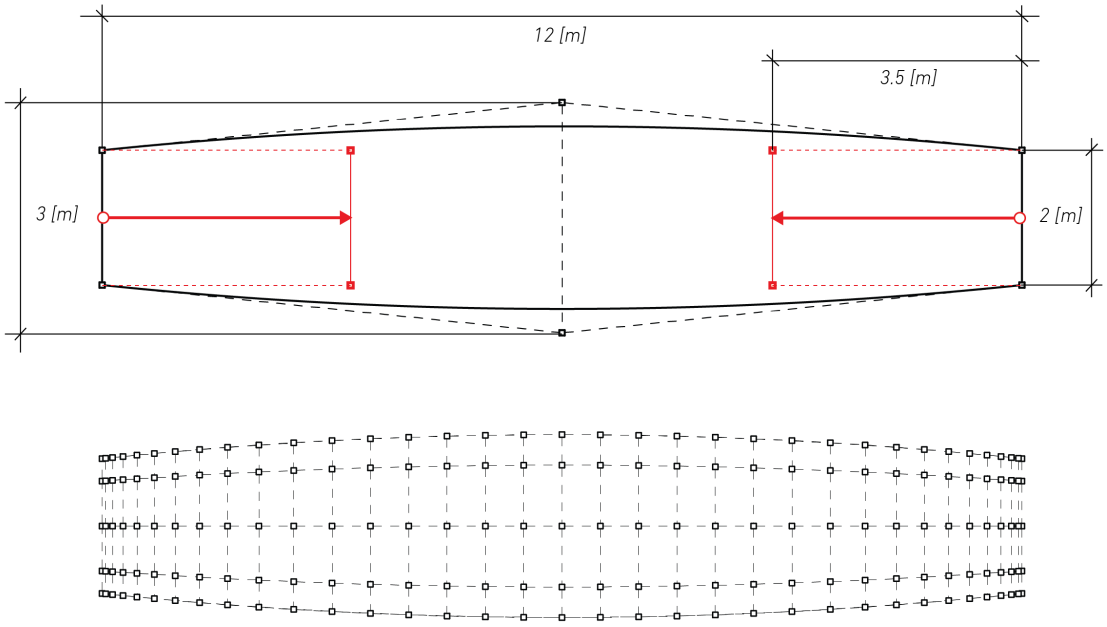


Figure 7.1: The flat reference configuration used for the forwards process

The first case is similar to the beam example illustrated in chapter 5. A planar sheet is modeled and subsequently deformed, by translating the boundaries, into a new and spatial configuration. These constraints are illustrated in red in figure 7.1 which shows the description of the reference geometry. To alter the design outcome of this process, different thickness distributions will be tried and their effect on both the nonlinear solution and the shape will be discussed.

The properties for the geometry and material can be seen in tables 7.1 and 7.2 and will be a linear material wit properties similar to typical plywood.

p	**q**	**numPts**	**numElems**	**Size [m]**
11	3	38x5	10	12x[2-2.5]

Table 7.1: Geometric properties

E $[N/mm^2]$	**nu**
10 000	0.2

Table 7.2: Material Properties

The nonlinear process needed to trace the equilibrium path will be displacement controlled and applied to the two short edges. The displacement of four points on each edge will be prescribed using lagrange constraints which will push the edges together. It will use the following process settings, shown in table 7.3.

Displacement [m]	**Num Steps**	**Tolerance [N]**	**Max Iter**	**Num Gauss Pts**
3.5	30	0.1	35	3x3

Table 7.3: Analysis setup for the simple elastica

7.1.2 Thickness Map

To manipulate the equilibrium, the stiffness will be varied over the sheet using a custom map which describe the thickness distribution. This is used by the tool as a reference surface that is modeled, which will be used to read the thickness values at each point. To study the influence of thickness variation on the process, four different maps will be used. These are only varying in the ξ direction and are therefore defined by a NURBS-curve that is extruded, to achieve the reference surface that describes the thickness profile. The following table describes the parameters used for the four maps, and as the maps are constant in the η direction, only the ξ direction is described.

Map	ξ positions	**Z values [m]**	**p**
Map 1	[0,0.2,0.5,0.8,1]	[0.02,0.02,0.02,0.02,0.02]	2
Map 2	[0,0.2,0.5,0.8,1]	[0.1,0.1,-0.02,0.1,0.1]	2
Map 3	[0,0.2,0.5,0.8,1]	[0.05,0.05,0.12,0.05,0.05]	2
Map 4	[0,0.1,0,3,0.4,0.6,0.7,0.9,1]	[0.1,0.1,0.066,0.033,0.033,0.66,0.66,0.167,0.66]	2

Table 7.4: Data for thickness maps

The resulting thickness maps are shown in figure 7.2 plotted on a unit domain. These are the result of a set of specified values that are used as the control points through which the thickness surface is interpolated.

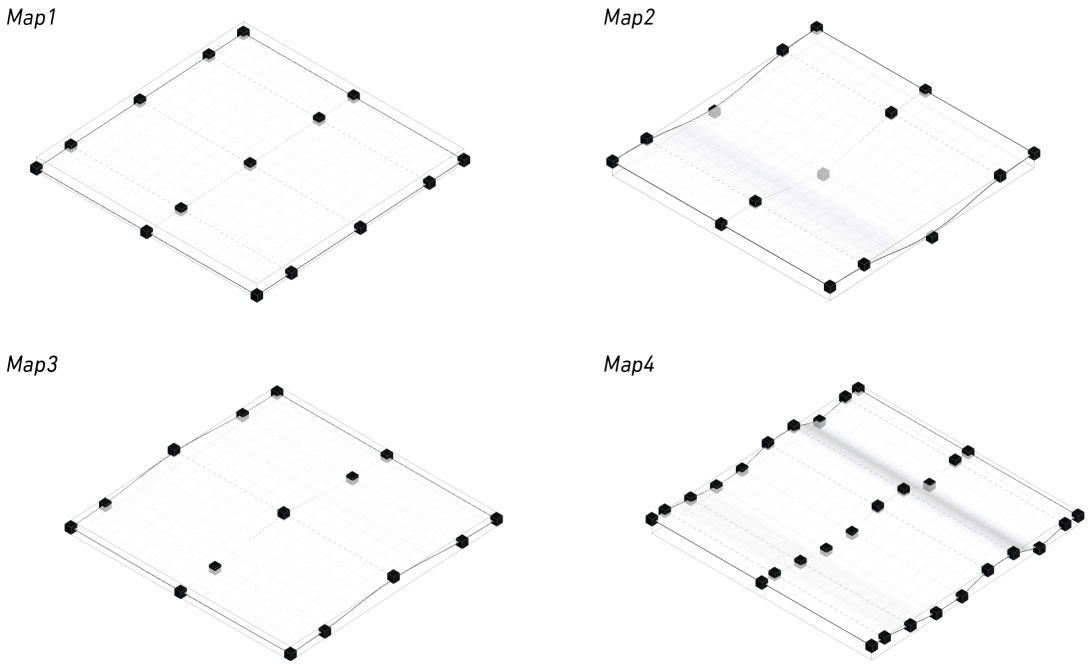


Figure 7.2: Thickness maps resulting from values in table 7.4

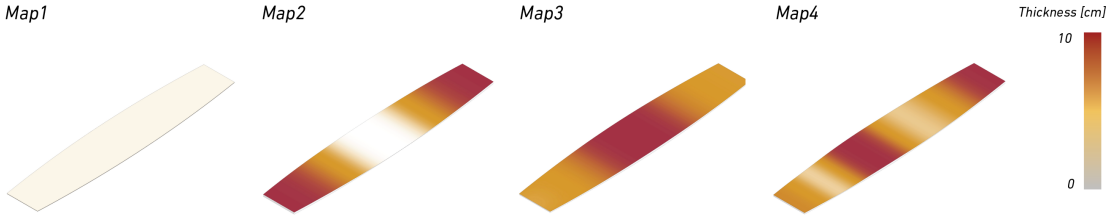


Figure 7.3: Thickness distributions resulting from maps in figure 7.2

The resulting distribution on the reference configuration can be seen in figure 7.3. The first map is simply a constant thickness. The second map has a reduced thickness in the middle of the sheet, whereas the third instead has a higher thickness in the middle. The first three maps are symmetric, with the fourth map introducing asymmetry to the system featuring two zones with reduced thickness.

7.1.3 Process

To find the deformed geometry the system is solved for the boundary displacements in the specified steps. An example of the deformation process for map 1 can be seen in figure 7.4, where the undeformed geometry (step 0) and the final geometry (step 30) are shown along with some intermediate steps (step 5,15).

The different thickness maps will result in different equilibrium geometries and different post-buckling behaviour. Figure 7.5 show the load/displacement relationship for the first degree of freedom, which can is show in figure 7.4 as a red arrow. With reduced stiffness zones, the curvature quickly concentrates and map2 and map4 show a quick transition over to a rather constant, bending dominated load-displacement relationship. The displacement controlled solution scheme showed no problem to trace any of the paths.

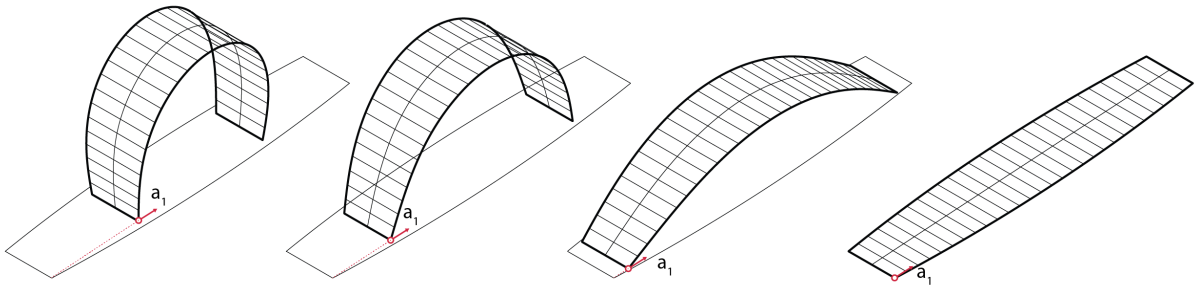


Figure 7.4: Steps 0,5,15,30 (Right to left) for the constant thickness model (Map 1)

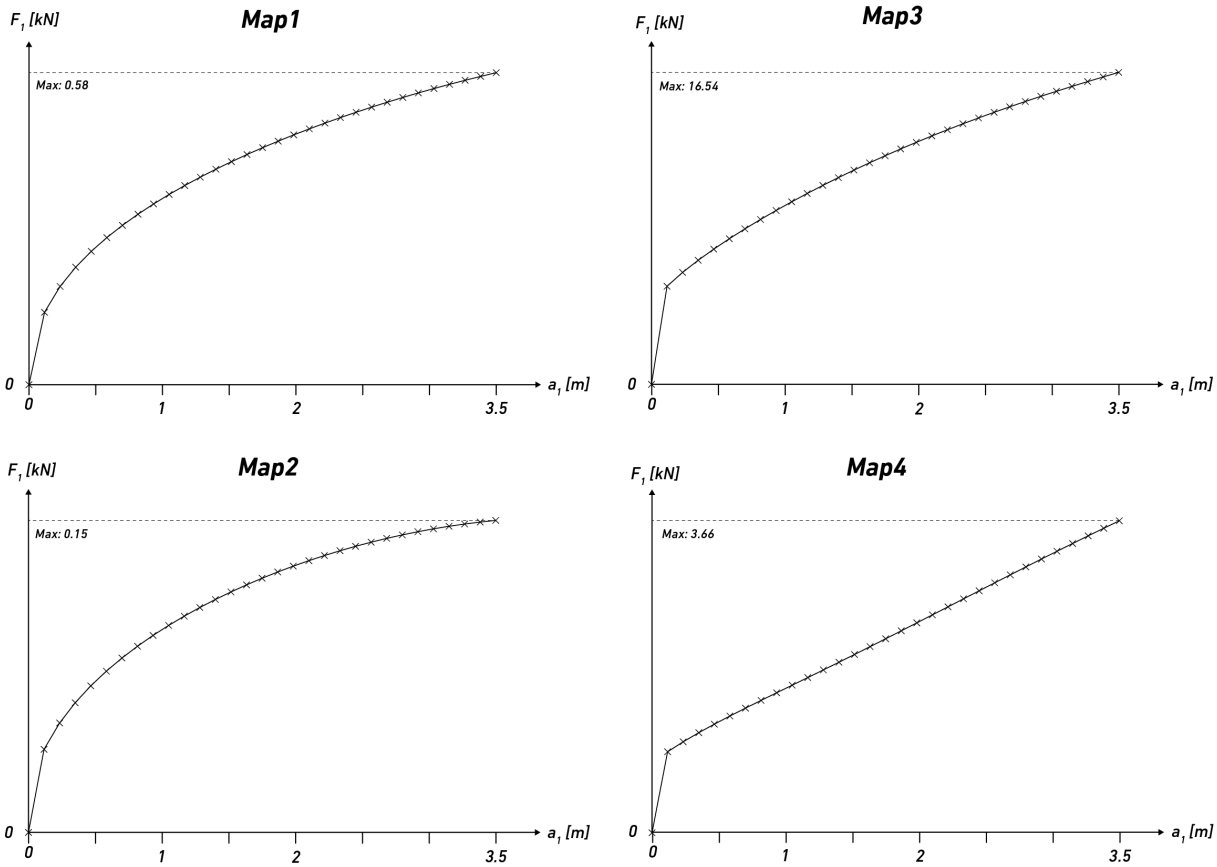


Figure 7.5: Load displacament curves for the maps in figure 7.2

7.1.4 Results

Figure 7.6 shows the resulting geometries for the different thickness maps. The thickness distribution is clearly visible in the resulting form as hints at hinges start as curvature concentrates in the thinner spots. The interplay between form and stiffness can be seen further in figure 7.7 where the thickness is shown on the final deformed geometry.

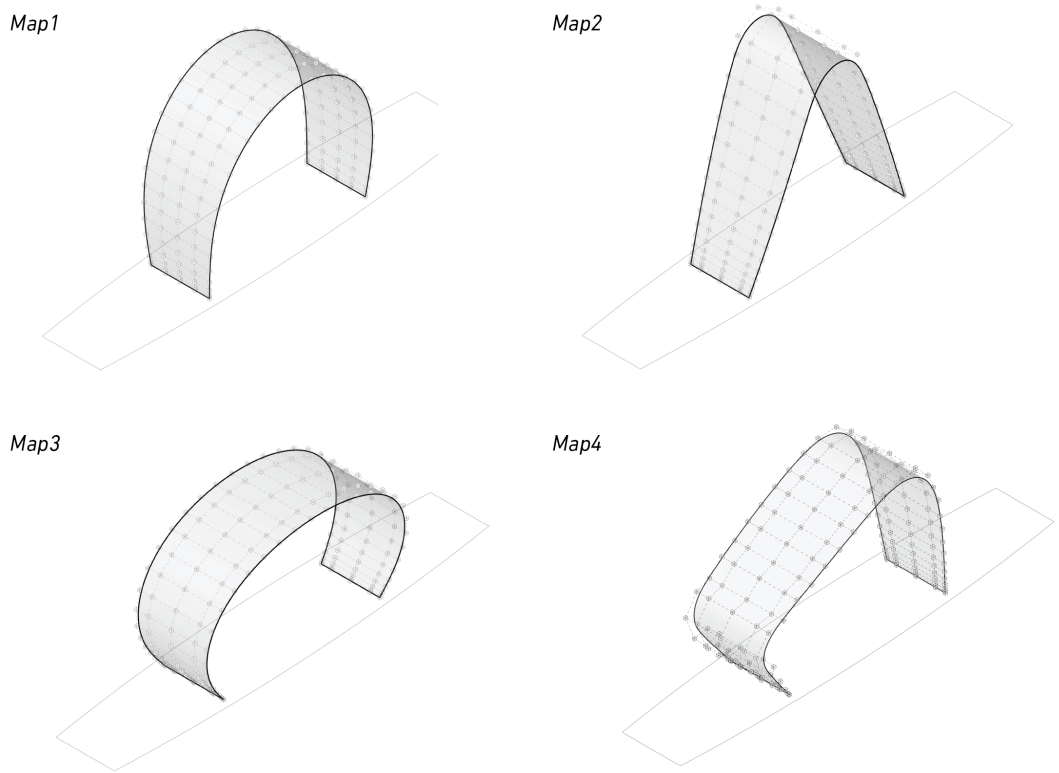


Figure 7.6: Geometries resulting from maps in figure 7.2

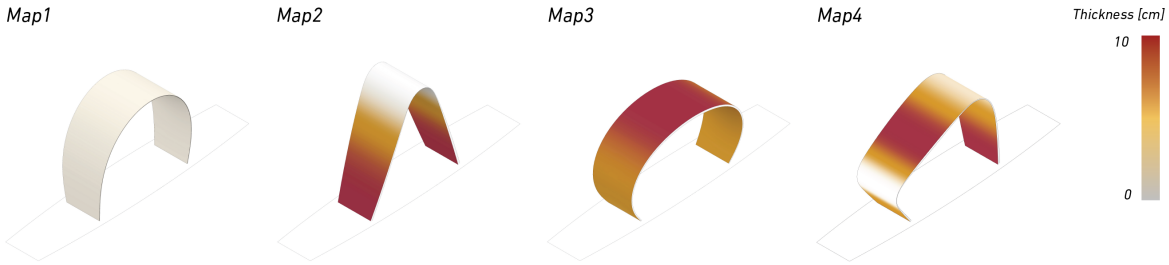


Figure 7.7: Deformed geometries with corresponding thickness distribution

7.2 Backwards Process

7.2.1 Setup

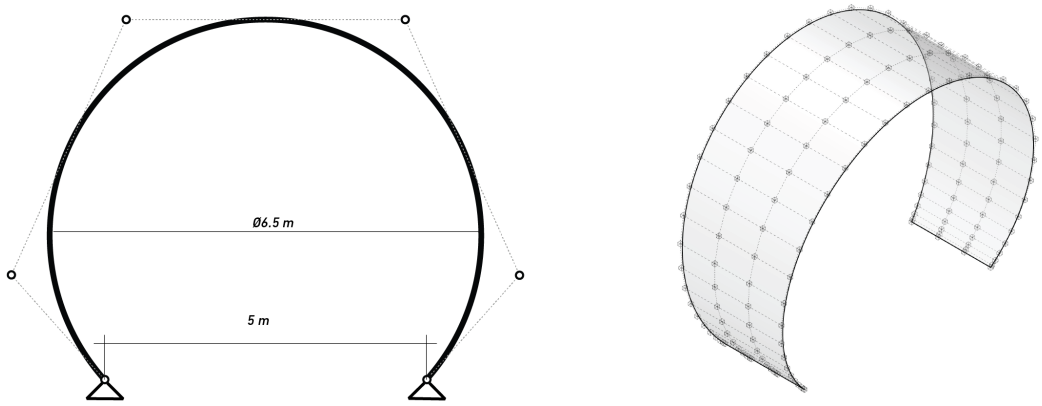


Figure 7.8: Target geometry at time t for the backwards process (left modeled profile, right Refined surface

The following backwards tracing will recover a thickness map that corresponds to the equilibrium of a predefined target geometry. Here, a similar geometry as the previous chapter will be traced backwards so that it's final equilibirum corresponds to a segment of a perfect circle. The target geometry has the following properties.

p	q	numPts	numElems	Size [m]
11	3	38x4	10	12x[2-3]

Table 7.5: Geometric properties for the geometry in figure 7.8 (right)

The equilibrium will be achieved by adjusting the thickness (stiffness) so that the internal forces are reduced. The forces will be asses by looking at the springback, which means, how much the target geometry deforms in a simple linear calculation. If the target is in equilibrium it should just stay in place as no force is acting on it.

To achieve find a thickness distribution that corresponds to the deformation of the target geometry, the target needs to be in equilibrium. This will be achived but performing a linear analysis, starting from the desired target at time t. An optimisation algorithm will be used to minimize the internal forces in the target by checking the spring back from the form as the analysis is run, as this deformation will come from residual internal force. The process will use the values of thickness for the control points of a preset thickness map as variables to minimize this springback. The result will therefore depend on the amount of freedom given to the thickness map.

The process is carried out in two steps; First a planar map is established as an undeformed reference configuration followed by a parameterisation and domain for the thickness distribution, which is then optimised to minimize the springback.

7.2.2 Planar Map

To establish a reference configuration, a planar map is needed. As the modeled surface is developable, the Rhino command *UnrollSrf* is used [30]. The result from the operation is seen in figure 7.9. An identical paramterisation is established by refining the unrolled surface to correspond to the target geometry.

There will be a small deviation in the area of the target and the flat representation. The deviation, however, is quite small which can be seen in figure. As the deformed shape should be primarly subject to bending deformation, the planar map should have a similar area to the target, which is achieved using *UnrollSrf* as can be seen in 7.6.

Layer	**Target Area [m2]**	**Planar Area [m2]**	**Refined Area [m2]**	**Deviation [%]**
Outer	41.01	41.01	41.01	0.0001

Table 7.6: Area comparison between planar map and target geometry

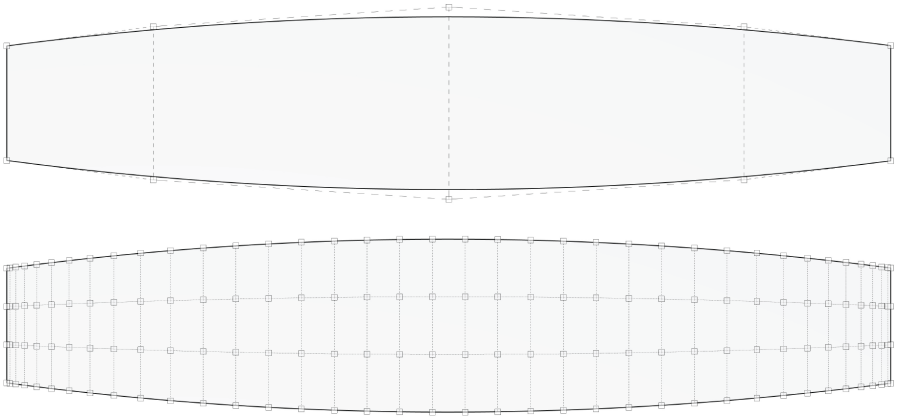


Figure 7.9: Raw and refined planar map recovered from 7.8 (right)

7.2.3 Thickness Map

To fit a thickness to the geometry a profile needs to be established. In this study, a NURBS map will be set up using a set number of points that can vary within a specific domain.

For this study, three levels of refinement will be applied. The first map will be based on a minimal setup for a smooth curve, using a 2nd degree interpolation and three control points. For the next maps this will be refined into 8 and 16 control points. The degree is kept to keep the effect of each point local to its position.

The setup for the three maps can be seen in table 7.7.

Map	**U Positions**	**Z domain [m]**	**p**
1	[0, 0.5, 1]	0.010 - 0.150	2
2	[0,0.08,0.25, 0.42, 0.58, 0.75, 0.92, 1]	0.010 - 0.150	2
3	[0,0.04,0.11, 0.18, 0.25, 0.32, 0.39, 0.46, 0.54, 0.61, 0.68, 0.75, 0.82, 0.89, 0.96, 1]	0.010 - 0.150	2

Table 7.7: Setup for Thickness maps

7.2.4 Process

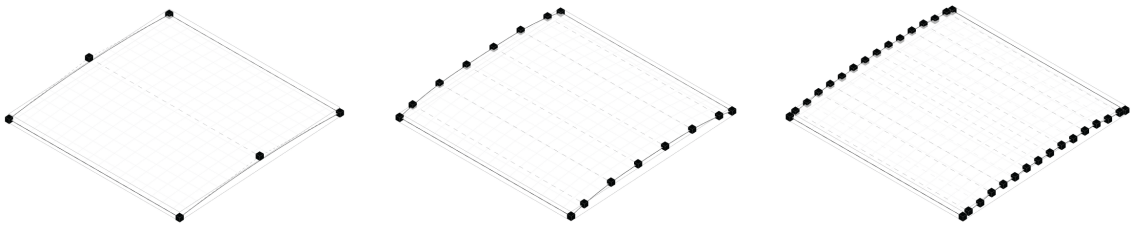


Figure 7.10: Maps from parameterisations described in table 7.7

The problem is solved using the genetic algorithm solver featured in the Grasshopper environment and the interface can be seen in figyre 7.11. This process will not be further explained here, but the reader is referred to sources on that topic elsewhere, for example [31].

The problems solved is simply to minimize the maximum deviation at any point of the geometry by adjusting the values of the control points of the thickness. The process converged very quickly, as can be seen in the top of the galapagos windows,which shows the distribution for each generation, and the resulting maps are shown in figure 7.10.

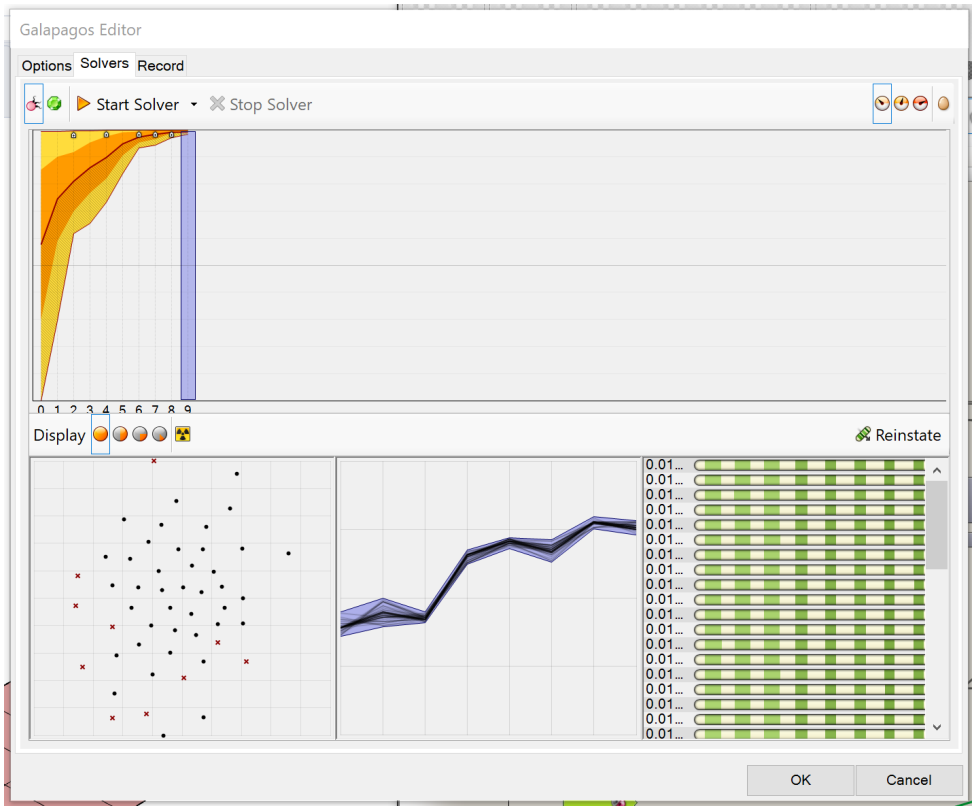


Figure 7.11: Snap from the galapagos interface (Map 3 process).

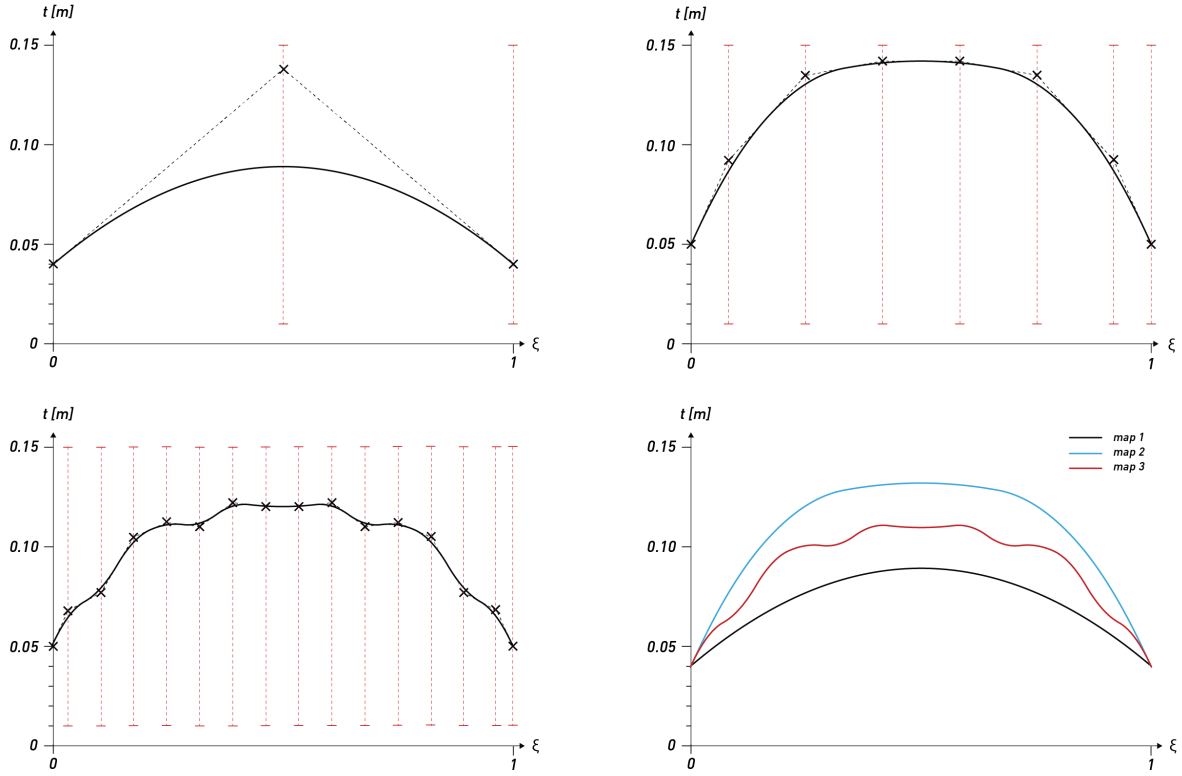


Figure 7.12: Resulting thickness after optimisation. Top left shows map 1, top right map 2, bottom left map 3 and bottom right all together

7.2.5 Results

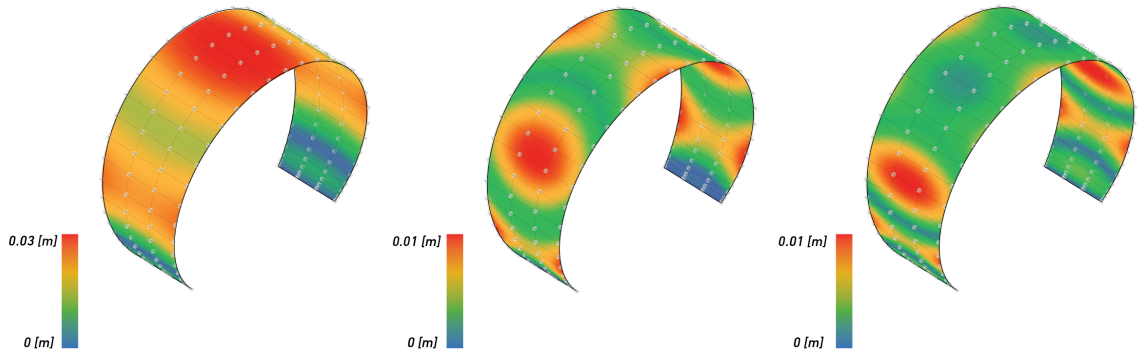


Figure 7.13: Springback for the different maps after optimisation (left: map1, center: map2, right: map3

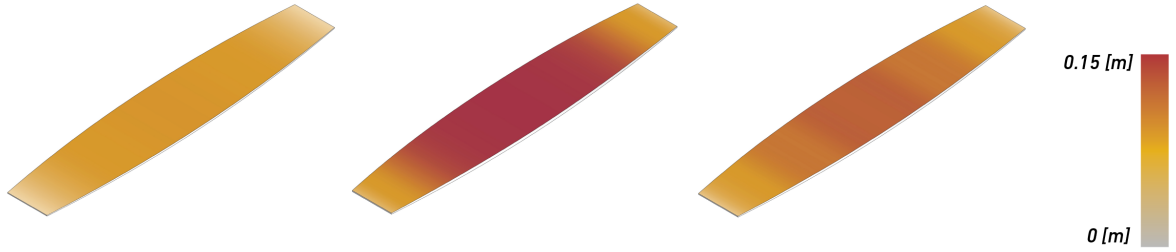


Figure 7.14: Thickness distribution for the different maps after optimisation (left: map1, center: map2, right: map3

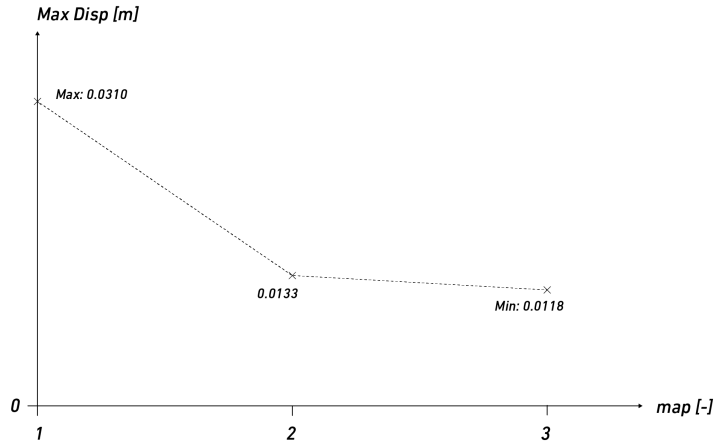


Figure 7.15: Comparison between max spring back for each map 7.2

Figure 7.13 shows the springbacks after optimisation, and figure 7.14 shows the resulting thickness distribution. The results show that an increased freedom for the thickness creates a better fit, however, for the current and rather simple example this seems to plan out quite quickly, as can be seen in figure 7.15

7.3 Design Case

The forwards and backwards tracing will in the following chapter be applied to a design case. This will follow very similar steps to the previous examples and will therefore be quickly stated and for detail, the previous examples can be consulted.

7.3.1 Setup

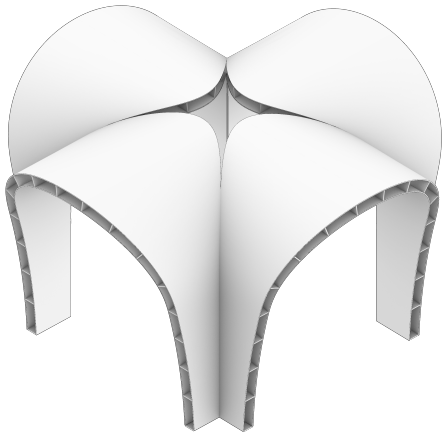


Figure 7.16: Geometries with thickness distribution from maps in figure 7.2

The design to be targeted is the following geometry modeled using NURBS in Rhino. It shows a doubly symmetric double layer shell which consists of eight pieces on total . As each piece is individually bent, the system can be reduced to two unique pieces, referred to as the *outer* and *inner* sheet respectively.

The two pieces can be seen in figure 7.17 and the geometric properties are described in table 7.8. The material will be the same as the previous examples (which was listed in table 7.2.

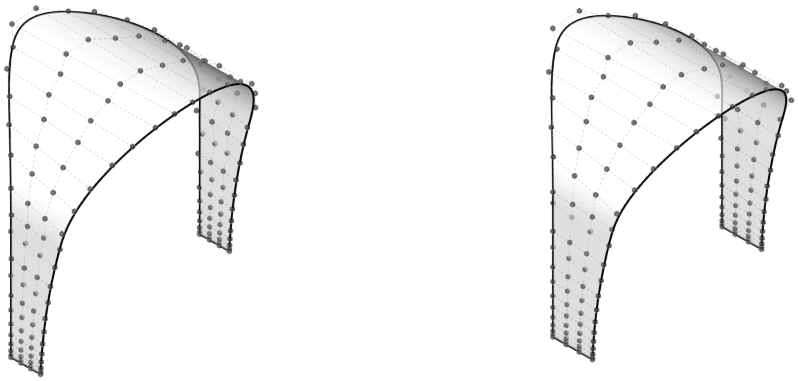


Figure 7.17: Geometries with thickness distribution from maps in figure, Outer (left) and inner (right) 7.2

p	**q**	**numPts**	**numElems**
11	3	38x4	26x1

Table 7.8: Geometric properties

7.3.2 Backwards

7.3.2.1 Process

The process starts of with the same steps as what was previously shown for the backwards tracing. This means starting with establishing an undeformed reference using *UnrollSrf* which is shown in figure 7.18. The slightly more complex geometry results in a less perfect area approximation, and a comparsion is shown in table 7.9 and the most critical zones are shown in figure 7.19. However, the deviation is still very small.

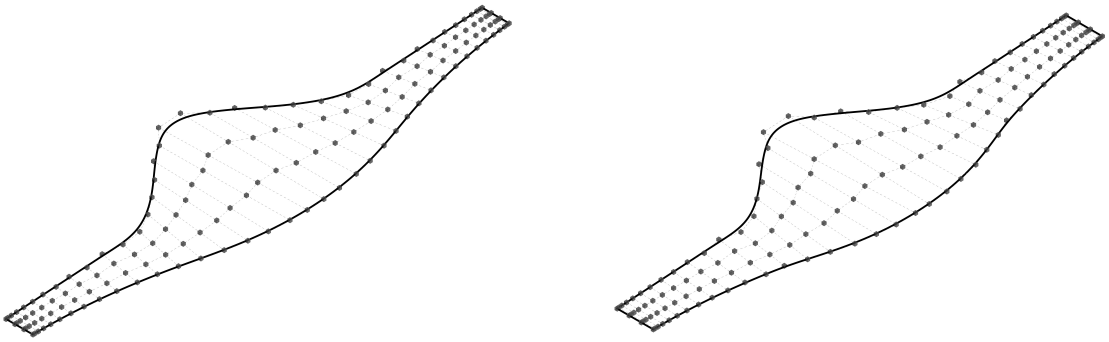


Figure 7.18: Planar maps for the outer (left) and the inner (right) layers of the design target

Layer	Target Geometry Area [m2]	Planar Map Area [m2]	Difference [%]
Outer	24.532	24.544	0.049
Inner	25.172	25.184	0.048

Table 7.9: Area comparison of target geometry and planar map

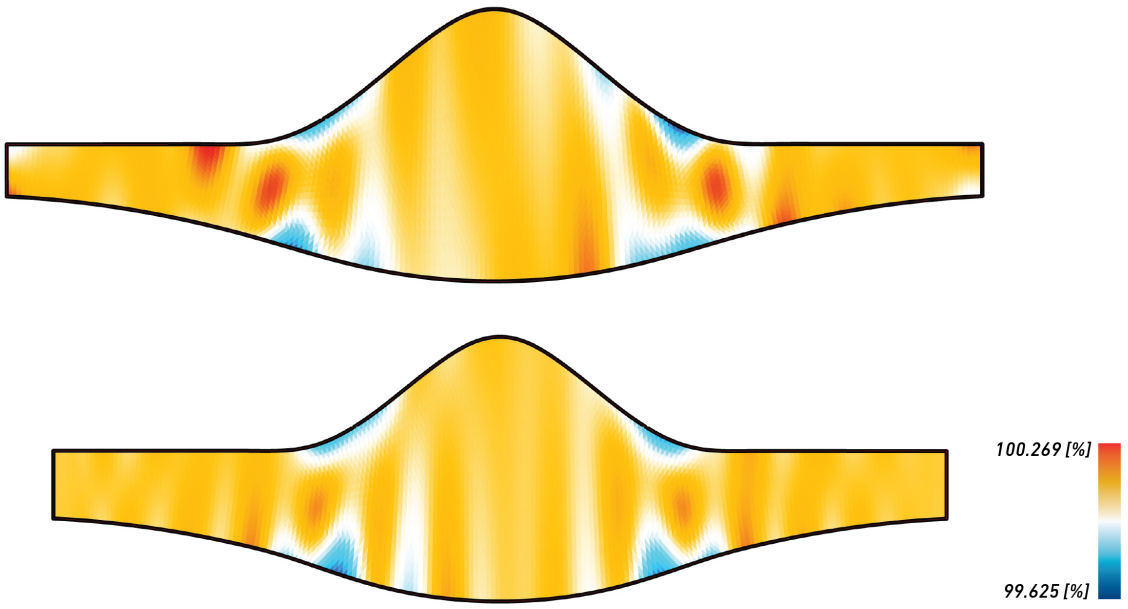


Figure 7.19: Area deviation across the two planar maps compared to their target. (Top: outer, Bottom, inner)

7.3.2.2 Result

The optimisation uses the same parameterisation as map 3 in the example for the backwards process, with the exception that it now is allowed to vary linearly in the η direction. The following results are achieved after running the optimisation process for the outer and inner layer.

Figure 7.20 shows the acquired thickness maps for the outer and inner layer, and the resulting springback is shown in figure 7.21, which can be seen to be a good fit with a maximum deviation of around 1cm.

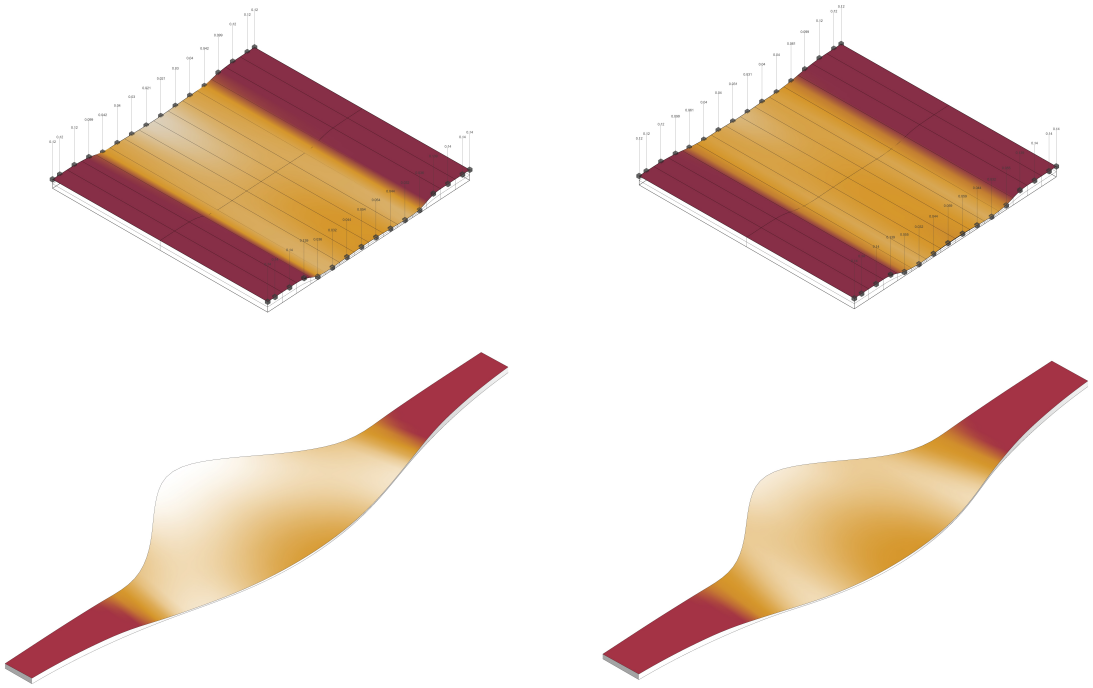


Figure 7.20: Geometries with thickness distribution from optimisation process: Outer (left), inner (right)

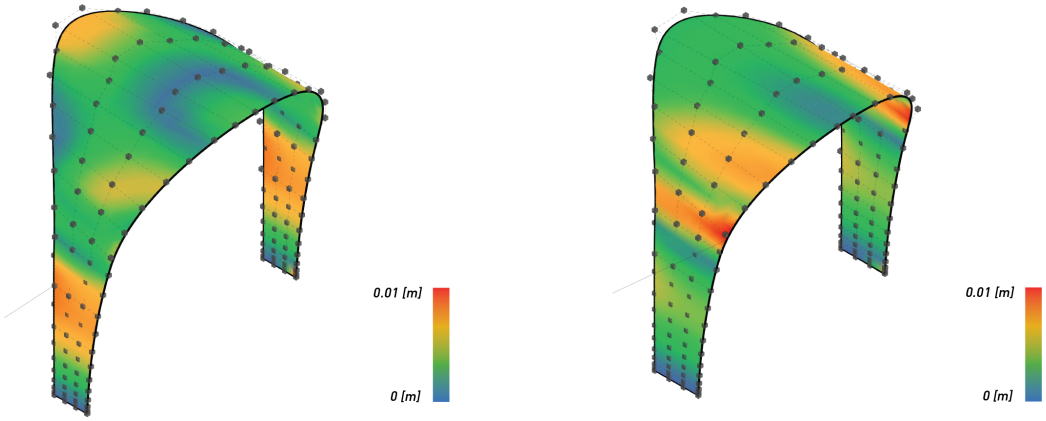


Figure 7.21: Resulting springback from the optimised thickness distributions, Outer (left) and inner (right)

7.3.3 Forwards

With a thickness distribution associated with the desired equilibrium geometry generated, we can now recover an approximation of the target design. This allows us to study that the construction process will return the desired equilibrium configuration, and not another possible equilibrium state. We can now also study the load displacement relationship for the construction and stress development.

7.3.3.1 Process

The process is again identical to the previous example. The end boundaries will be pushed together to correspond to the positions of the target geometry. The same number of steps and tolerances as earlier will be used here as well (table 7.3).

The form development can be seen in figure 7.22 with the stress in the bottom fibre plotted. The stress seem to be within, albeit on the limit, of the elastic range for a plywood or similar material. A comparison between the generated geometries and the reference can be seen in figure 7.17 where the deviation from the initial target is shown. The deviations are very similar to the springback from the backwards process in figure 7.21.

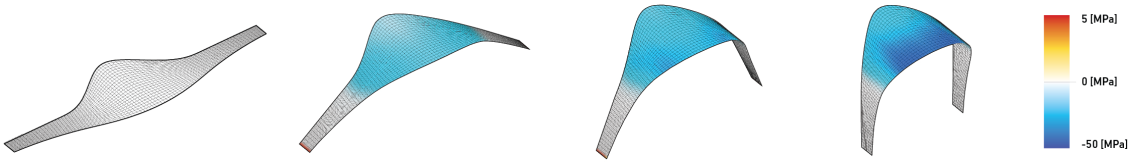


Figure 7.22: Stress development for the outer layer, stress shows bottom most fibre of material.

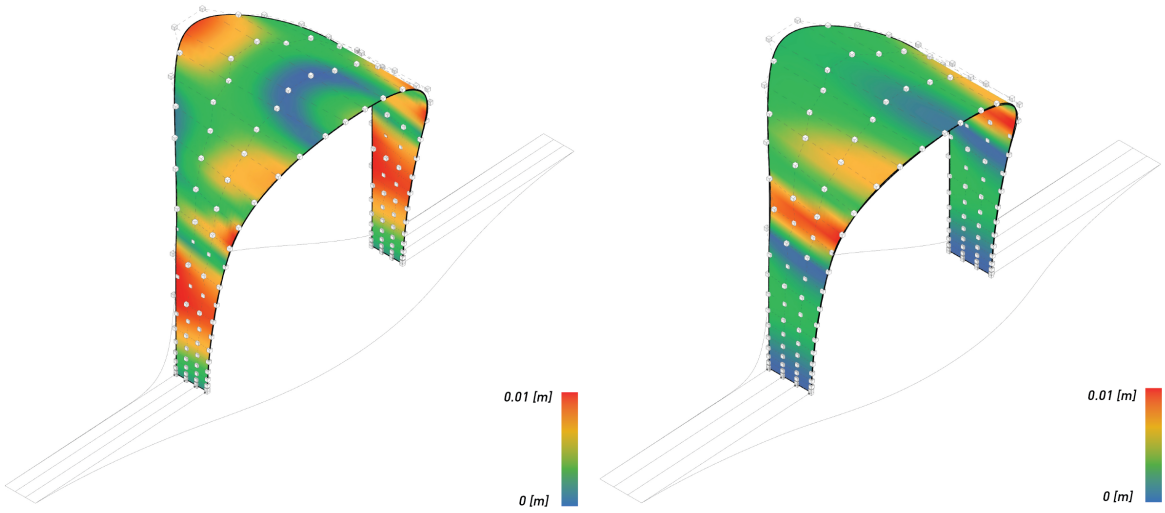


Figure 7.23: The devitation of the final step of the process compared to the initial target, Outer (left) and inner (right)

7.3.3.2 Result

The new, form found, pieces can then be assembled into a recovered design geometry. This extended geometry now hold information about stresses and thickness while still being a NURBS based design geometry. This can now be used for further analysis of the coupled system and to generate detailing etc.

The final geometry can be seen in figure 7.24 with the thickness shown in the left and the NURBS description on the right.

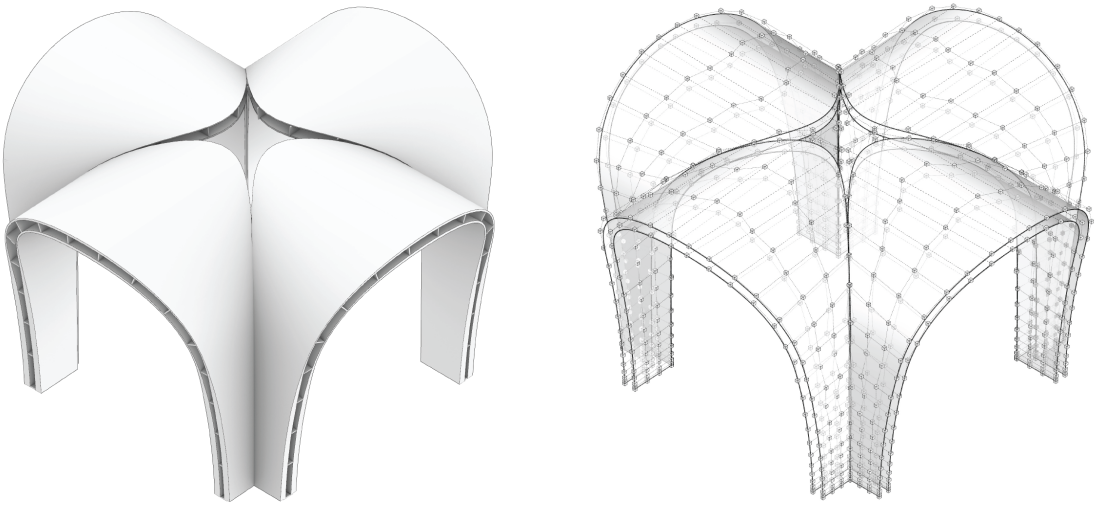


Figure 7.24: Recovered design geometry with thickness on the left and NURBS-representation on the right. 7.2

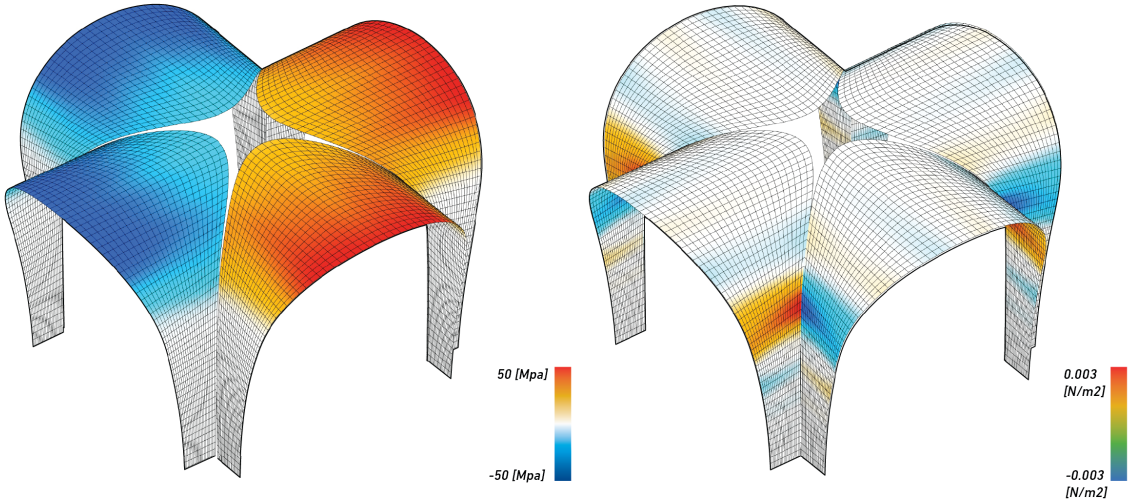


Figure 7.25: Normal stress in top and bottom layer (left) and torsional moment (right) 7.2

8

Conclusions

The following chapter will provide some concluding reflections on the work and recommendations for further work. It will cover IGA as a design method, the interface, along with recommendations for further work.

8.1 Reflection

8.1.1 General

In contrast to what is mainly associated with the term form finding, which is the generation of a form w.r.t a specific load case (e.g gravity) the process presented here instead targets the "load case" of the construction process, and how actively bent geometries can be controlled through the relations between undeformed geometry, boundary conditions and material and the final design as a mix of explicit geometric steering of the nonlinear process.

As the main ambition was to study the form giving procedure, thus only dealing with the shaping of individual sheets, no care was given to assessment or analysis of the bent and coupled shells. This is of course also an interesting topic, but was left out due to time.

The project shows that using IGA one can quite easily blend physical behaviour into the digiral design of shells and IGA provides a good framework for that. The parametric space also provide simple ways to apply various maps to the mesh which can be element independent, as can be seen in the used thickness maps, as all you need is to describe it in a normalised, square domain. This then makes the design process able to tailor custom fileds onto a design, which further can be manufactured, which allows us to actually control the physcial behaviour and prescribe how it should behave as a design approach.

By using the total lagrangian formulation the analysis formulation is very easily translatable into the physical design problem as both are concerned with on undeformed version and one arbitrarily deformed, meaning what you need for the formulation is what you have from the design problem. As the design problem were based on the relationship between undeformed and deformed, the formulation very naturally fits into the workflow.

8.1.2 IGA and Numerical Method

The main topic of the thesis was the application of IGA as a design tool and integrated analysis. On this note, a few conclusions can be made. Isogeometric analysis proves in one aspect a new of relating physical behaviour and analysis, by allowing the designer to work with NURBS should he see so fit. The seamlessness provide smooth integration within a NURBS based design environment. However, IGA come with some additional drawback compared to other methods (as mesh based FE-methods or other meshfree methods), and one of those is the patch based constraint when working with more free geometries. Working with trimmed patches and BREPs is necessary for a more multifunctional design tool, as now, everything need to have four edges. However, as the purpose in this case was to deal with segmented shells, this was for the task at hand not an issue.

The implementation of IGA used in this thesis have some issues that make using it a bit harder. One significant issue is membrane locking, which requires very high intepolation degrees to achive good results. A more sophisticated model using hierarchial elements for example.

During the process I was asked a question whether IGA was less intuitive than the conventional mesh based FE-methods and that is a good question. This comes from the fact that when dealing with mesh-based analysis, everything is on the surface (loads,constraints) which is not the case with IGA, as rotational constraints and Loads are applied to control points that are not necessarily part of the geometry. Rotational constraints for example become a bit unclear as the method does not have any rotations, only curvatures, however, this might also be something you just have to get used to. However, one of the perks of the use of NURBS is the way it enables the use of KL-shell theory as mentioned and further enables a higher order continuity which is desireable from both a design and analysis perspective.

8.1.3 Design Tool

The task was carried out and the final tool can absolutely be useful when dealing with the intended case. However, one can question the interactivity. The tools takes some time to solve the nonlinear equilibirum and in this process, you can't interact with it. The matrix solver used is robust and manages to solve the necessary cases, however, it needs a rigorous setup and one can not interact with the model as it's running. This reduces the sketching functionality of the tool making it hard to play around with and get a feel for the behaviour, but the precision and exactness needed makes it more suitable for finding a fabrication ready design geometry.

The outcome is a rather multifunctional analysis tool, which also can be used for assessment analysis of the coupled shells with the addition of some more load components, which isn't a too big effort. This would allow you to design the response of the structure accounting for the embedded stresses in a similar way to the backwards process, however, this was left out of the report.

Further, it would be ideal to decouple the tool from it's current MATLAB dependency, as that slows things down and is far from necessary. As this project started from scratch, MATLAB was good to use for developing the numerical methods and due to time this was not rewritten in some other language, which would have been ideal.

8.2 Recommendations for further work

- **Multipatch** modeling is an important aspect to make the method more usable. This topic has already been studied under the name IBRA (Isogeometric B-Rep Analysis) and breaks the constraint of the patch as it allows you to work with more free geometries.
- **Sketching Aspect** An idea to counter this is to instead use a dynamic relaxation solver, which was tried for the beam. The IGA formulation is very suitable to this, however, for this thesis it proved to much to get into, but it could potentially loosen up the numerical solution scheme into something more interactive.
- **Finding Thickness** The approach to use genetic algorithms is a very general approach to the optimisation problem of finding a good thickness distribution. This process could be further improved upon and potentially the material thickness could be computed in a more specific way, for example to a gradient descent to optimise the thickness values.
- **Eliminate Locking** The formulation used in this thesis show big issues with membrane locking, as can be seen in Appendix I. One needs to be aware of this when using the tool but even with very high degree of the NURBS interpolation, it's still present. Methods exist to eliminate this, and this should be implemented to have a good and rigorous numerical method.
- **Subsequent Analysis** With this thesis describing how the form can be found, the next step of course is to study the structural behaviour of these bent shells using IGA as a coupled and complete design.
- **Optimisation** When incorporating structural assesment analysis, another interesting problem emerges; namely how to optimise the material and form including both the elastic construction and the subsequent loads applied. How can you tie this whole process together, to find a thickness that renders an optimal form when subject to load, and an optimal thickness distribution.

Bibliography

[1] T.Hughes, J.Cottrell, Y.Bazilevs (2009) Isogeometric Analysis - Towards Integration of CAD and FEA, *Wiley*

[2] Lienhard, J. (2014) Bending Active Structures. *Dissertation, University of Stuttgart*

[3] Official Webpage for the workshops of Advances in Architectural Geometry 2018, *http://www.architecturalgeometry.org/aag18/workshops/*

[4] La Magna, R. (2017) Bending Active Plate Structures. *Dissertation, University of Stuttgart*

[5] Poulsen, E. (2016) Structural Design and analysis of elastically bent gridshells. *Master Thesis, Chalmers University of Technology*

[6] Almqvist, S. Safari, P. (2016) Isogeometric analysis of curved beams and thin shells. *Master Thesis, Chalmers University of Technology*

[7] Bischoff, M. Wall, W.A. Bletzinger, K-U (2004) Models and finite elements for thinwalled structures. *Encyclopedia of Computational Mechanics, volume 2, Solids, Structures and Coupled Problems.*

[8] Adriaenssens, S et. Al. (2014) *Shell Structures for Architecture, Routledge*

[9] Sasaki, M et. Al. (2007) *Morphogenesis of Flux Structure, AA Publications*

[10] G.Nordenson, T.Riley (2008) Seven Structural Engineers: The Felix Candela Lectures, *Museum of Modern Art, New York*

[11] R.Harriss, J Roynon (2008) The Savill Garden Gridshell Design and Construction *Structural Engineer 86(17):27-34*

[12] M.Carpo The Second Digital Turn: Design Beyond Intelligence (2016) *MIT Press*

[13] J.Knippers, J.Cremers, J.Lienhard, M.Gabler Construction Manual for Polymers and Membranes (2011) *Birkhäuser*

[14] O.Krieg, T.Schwinn, A.Menges Robotic Sewing (2016) *Proceedings: Advances in Architectural Geometry*

[15] Rhinoceros Official Website (www.rhino3d.com)

[16] Official Website *www.mathworks.com*

[17] J.Li Timber Shell Structures (2017) *Dissertation, University of Stuttgart*

[18] A.Alexandersson Isogeometric Analysis in Form Finding in Architecture (2015) *Master Thesis, Chalmers University of Technology*

[19] V.Alic, K.Persson NURBS based form finding of efficient shapes for shells (2016) *Proceedings of 29th Nordic Seminar on Computational Mechanics – NSCM29*

[20] M.Barnes Form finding and analysis of tension structures by dynamic relaxation (1999) *International Journal of Space Structures*

[21] J.Olsson Form Finding and Size Optimisation (2012) *Master Thesis, Chalmers University of Technology*

[22] L.Piegl, W.Tiller The NURBS Book (1996) *Springer-Verlag*

[23] Y. Basar and D. Weichert. Nonlinear continuum mechanics of solids. Springer Verlag, Berlin, Heidelberg, New York, 2000.

[24] A-M.Bauer et al. Nonlinear isogeometric spatial Bernoulli Beam (2016) *Computer Methods in Applied Mechanics and Engineering*

[25] K-J. Bathe Finite Element Procedures (2014)

[26] L.Sluys et al. Lecture Notes: Computational Methods in Nonlinear Solid Mechanics (2016) *TU Delft*

[27] J.M.Kiendl Isogeometric Analysis and Shape Optimal Design of Shell Structures (2011) *Dissertation, TU Munich*

[28] J.Brutting et al. Bending-Active Segmented Shells (2017) *Procedings: IASS Hamburg*

[29]

[30] Advanced Flattening; Official Rhino Website https://wiki.mcneel.com/labs/advancedflattening

[31] David Rutten, Description of Galapagos https://www.grasshopper3d.com/forum/topics/references-about-galapagos
[32] Kiwi3d Official Webpage https://www.kiwi3d.com/
[33] Photo: *Archive ILEK, Stuttgart* Source: *https://mai-nrw.de/vom-raumwunder-und-seinen-ingenieuren-die-multihalle-in-mannheim*
[34] Photo: *Addison Godel* Source: *https://www.flickr.com/photos/doctorcasino/12041660445/in/photostream/*

A

Appendix 1

The study in this appendix will show the relationship between membrane locking and interpolation degree. A simple elastica is deformed until it's short edges are pushed together. A series of models with increasing p degree will be evaluated and their stresses in the centerplane and maximum vertical displacement compared.

Figure A.1 and A.2 show how the lower degree models have a much stiffer behaviour and at around $p = 11$ it starts to plan out. This is due to the membrane locking effects that can be seen in figure A.3 where normal forces develop in an oscillating manner in the centerplane of the geometry that should mainly be subject bending. The stresses remaining in the last model are mainly due to the counter bending from poisson's ratio, however, some locking can still be seen close to the support.

Model	p	q	numPts	numElems	Geometry
1	2	2	12x3	10	10x2x0.05
2	3	2	13x3	10	10x2x0.05
3	5	2	15x3	10	10x2x0.05
4	7	2	17x3	10	10x2x0.05
5	9	2	19x3	10	10x2x0.05
6	11	2	21x3	10	10x2x0.05

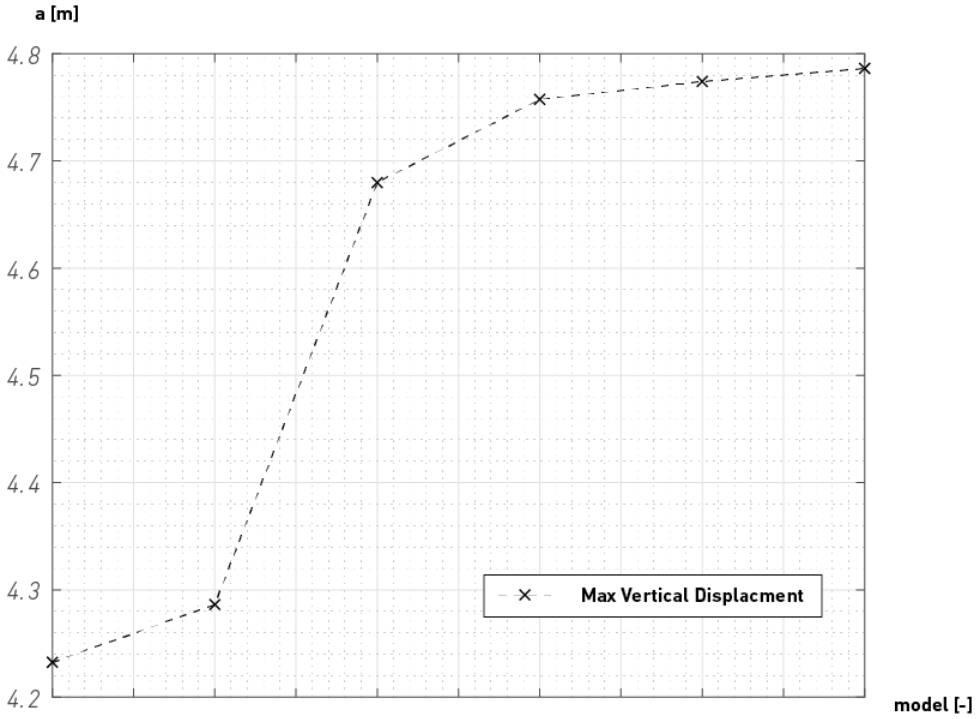


Figure A.1: Maximum vertical displacement for each model

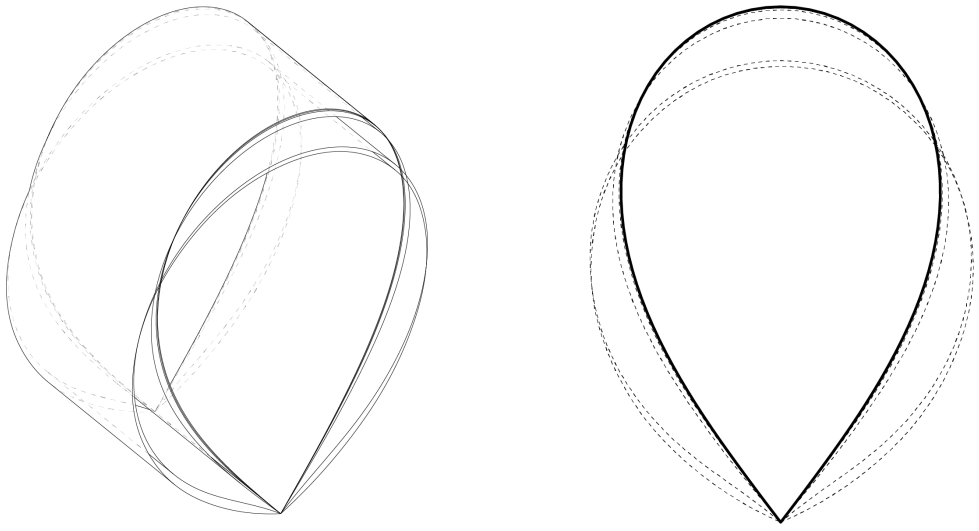


Figure A.2: Comparison of resulting geometries for the different models.

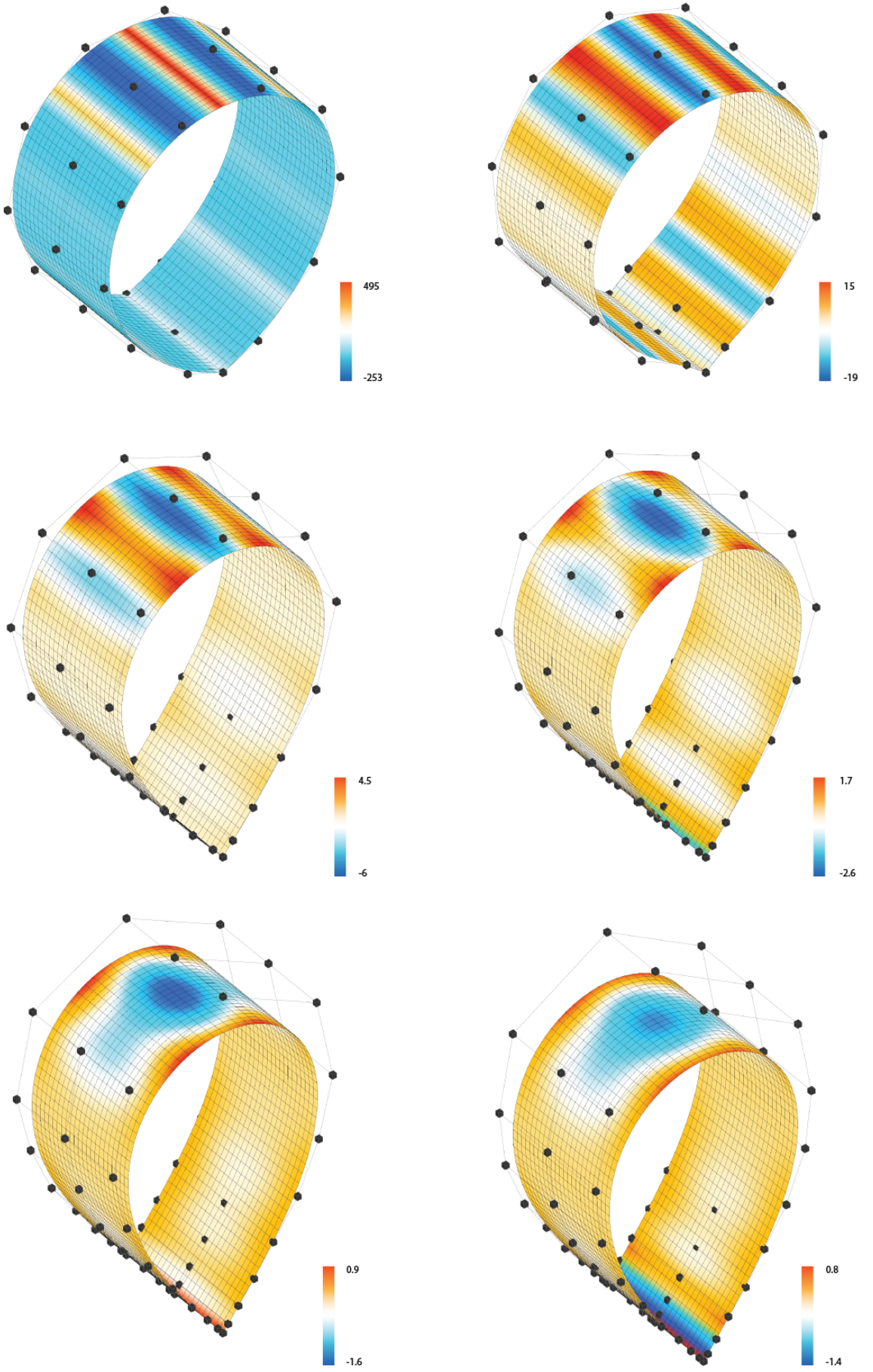


Figure A.3: Normal stress in the longitudinal direction in the center plane

B

Appendix 1

The following appendix will compare the results from the IGA analysis to some existing solvers.

For this benchmarking, two models will be examined. First, a simple clamped shell subject to a point load will be used to verify the implementation of the variable thickness. This will be compated to the FE solver Karamba, which is also available within the grasshopper framework, that uses triangular shell elements.

Secondly, a nonlinear, complete elastica will be compared to the Isogeometri Analysis tool Kiwi3d.

B.1 Clamped Shell

The first example will compare a clamped slab subject to a point load with a smooth thickness map applied. This result will be compared to a similar FE model using triangular shell elements with custom thickness per element to correspond to the smooth map of the IGA model. The setup for the analysis is seen in figure B.1. The shell model will use a constant thickness per element, as shown in figure B.3.

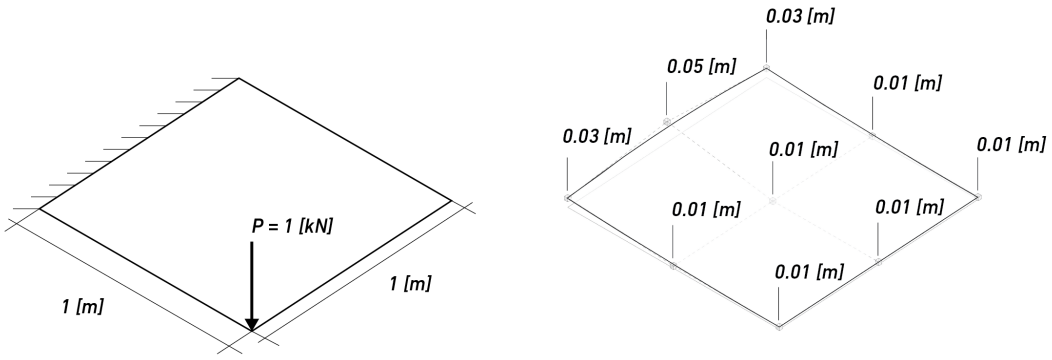


Figure B.1: System for the clamped model (left) thickness distribution (right)

Model	p	q	numPts	numElems	Geometry
1 (IGA)	3	3	15x15	12x12	1x1 [m]
2 (FEM)	1 (Triangle Basis)	-	420	200	1x1 [m]

Table B.1: The geometry data for the two models

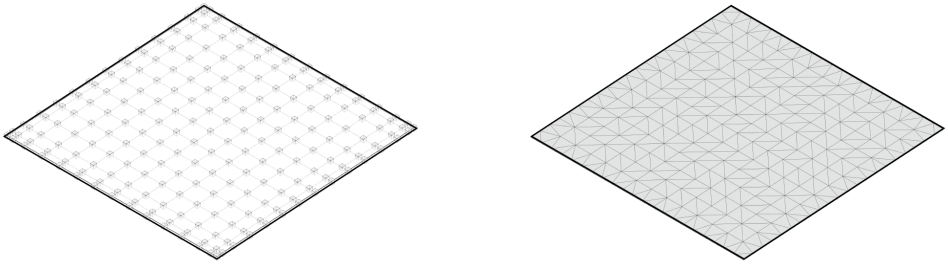


Figure B.2: Mesh for the IGA model (left) and FEA model (right)

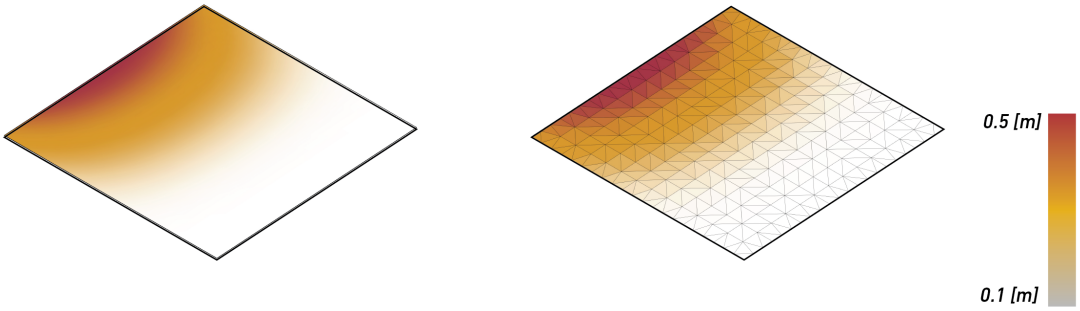


Figure B.3: Thickness map for IGA model (left) and element thickness for FEA (right)

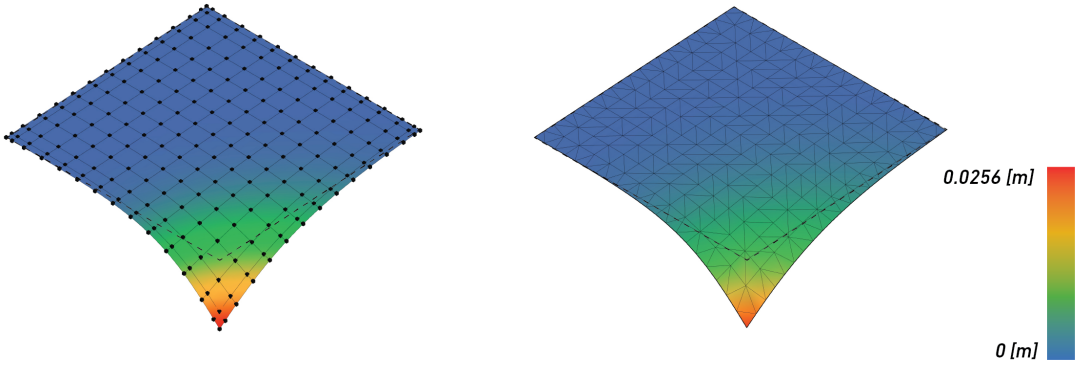


Figure B.4: Resulting deformation for IGA model (left) and FEA model (right)

Model	Result [m]	Difference [%]
1 (IGA)	0.0256	+0.39
2 (FEM)	0.0255	0

Table B.2: Results of the analysis and comparison

B.2 Elastica

This benchmark compares two nonlinear solutions for the elastica using a constant thickness comparing the developed tool to Kiwi3d. The input models are very similar except that, as Kiwi3d has a better implementation not subject to locking, it can work with lower interpolation degrees.

Model	**p**	**q**	**numPts**	**numElems**	**Geometry**
Thesis Tool	11	3	38x4	25x1	12x2 [m]
Kiwi3D	3	3	38x4	25x2	12x2 [m]

Table B.3: The geometry data for the two models.

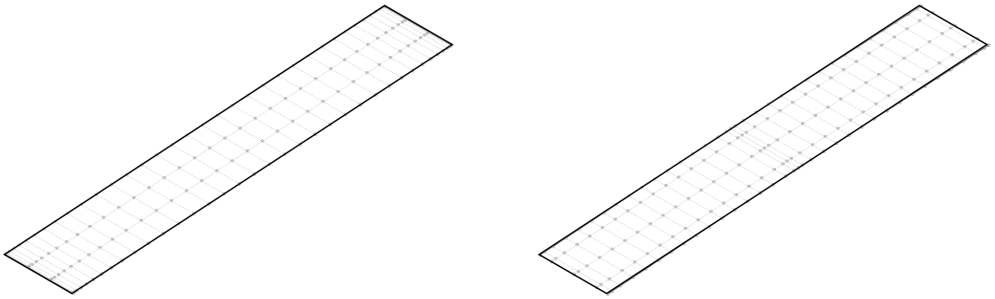


Figure B.5: The two geometries to be compared. Kiwi Model on the right

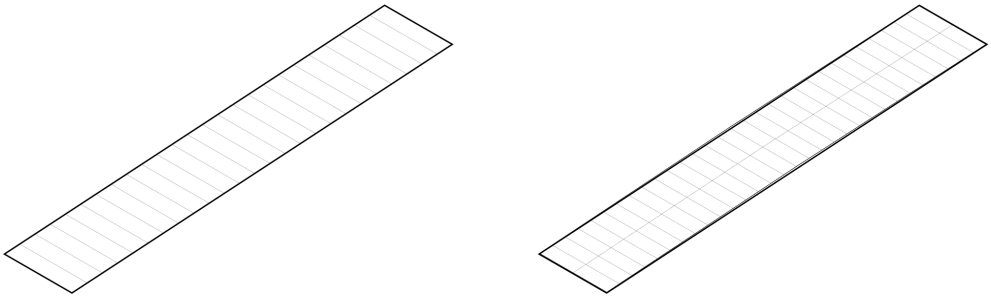


Figure B.6: Element division for the two models in figure B.6

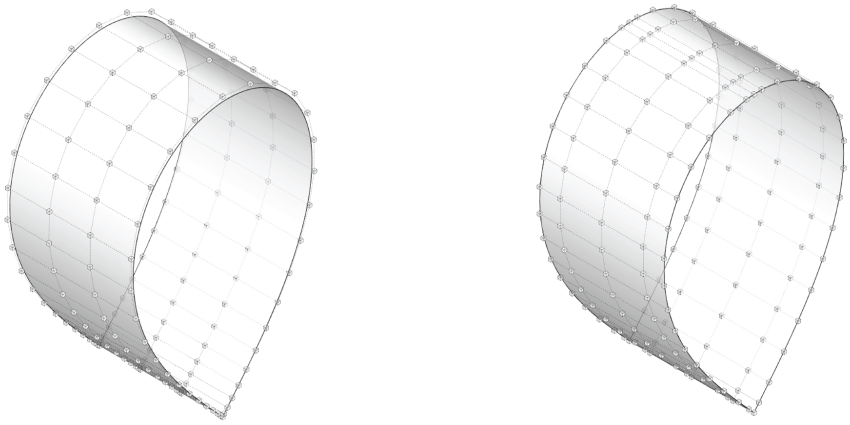


Figure B.7: The two elastica solutions the tool of this thesis (left) and Kiwi (right)

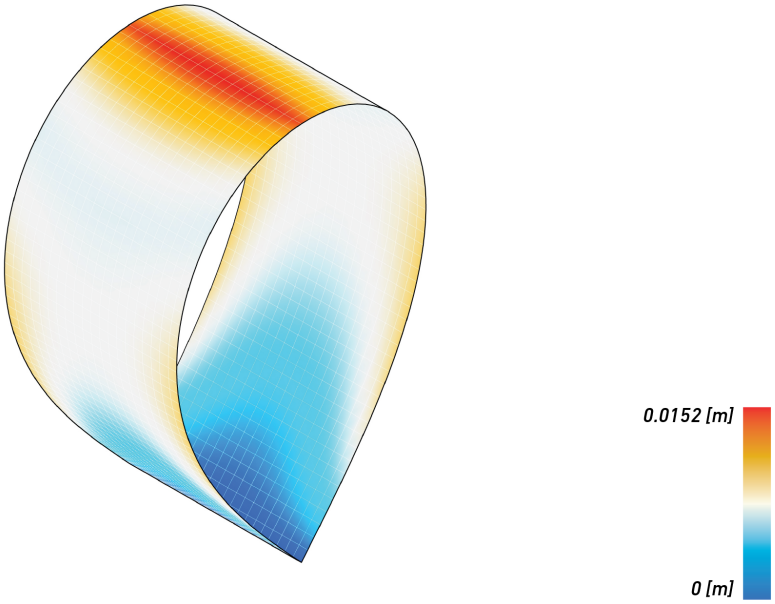


Figure B.8: Deviation between models in figure B.7