



CHALMERS



Isogeometric Analysis of Curved Beams and Thin Shells

Comparing the analysis technique to
classic finite element analysis

Master's thesis in Structural Engineering and Building Technology

PURIA SAFARI HESARI
SARA ALMSTEDT

MASTER'S THESIS IN STRUCTURAL ENGINEERING AND BUILDING TECHNOLOGY

Isogeometric Analysis of Curved Beams and Thin Shells

Comparing the analysis technique to classic finite element analysis

PURIA SAFARI HESARI
SARA ALMSTEDT

Department of Applied Mechanics
Division of Material and Computational Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2017

Isogeometric Analysis of Curved Beams and Thin Shells
Comparing the analysis technique to classic finite element analysis
PURIA SAFARI HESARI
SARA ALMSTEDT

© PURIA SAFARI HESARI , SARA ALMSTEDT, 2017

Master's thesis 2017:47
ISSN 1652-8557
Department of Applied Mechanics
Division of Material and Computational Mechanics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:

The sculpture "Crest" was installed at the Victoria and Albert Museum in London the summer of 2014. The structural and architectural design is the result of collaborative work between BuroHappold Engineering, Zaha Hadid Architects and LiteStructures. It is used as case study subject in this thesis project. Photograph courtesy of ©Ed Reeve.

Chalmers Reproservice
Göteborg, Sweden 2017

Isogeometric Analysis of Curved Beams and Thin Shells
Comparing the analysis technique to classic finite element analysis
Master's thesis in Structural Engineering and Building Technology
PURIA SAFARI HESARI
SARA ALMSTEDT
Department of Applied Mechanics
Division of Material and Computational Mechanics
Chalmers University of Technology

ABSTRACT

New analysis tools must be developed in order to realize designs of increasing geometric complexity. The iterative work flow that is practiced between engineers and designers today is not fully sufficient. Reports from several industries state that 80% or more of analysis time is spent on converting Computer Aided Design (CAD) data to analysis suitable models. Besides being highly time-consuming, this procedure is also prone to generate errors. When going from CAD models to analysis models a change of discretization is performed which introduces geometrical approximations. Thus, a common platform for designers and engineers to work on is needed. A technique that stands out in its ability to truly integrate analysis with CAD is Isogeometric Analysis (IGA). It is a young approach in Finite Element Analysis (FEA) that utilizes the same mathematical basis for description as CAD technology, i.e. Non-Uniform Rational B-Splines (NURBS). In this thesis, the benefits and drawbacks of using IGA for integrated structural and architectural design are investigated.

The question posed is whether IGA gives more accurate results and higher convergence rate than classic FEA on a per-degree-of-freedom basis. To this end, isogeometric analysis models based on Timoshenko and Euler-Bernoulli beam theory are implemented and the results are compared with results from classic FEA. Additionally, an isogeometric analysis model based on Kirchhoff-Love shell theory is implemented. For shells, the structural mechanics module of the finite element analysis software COMSOL Multiphysics[®], the software suite Abaqus[™] and the programs ETABS[®] and Robot[™] are used for comparing IGA with classic FEA.

The findings from the studies of beams suggest that IGA gives more accurate results. Additionally, the ability to utilize higher continuity over element boundaries in IGA seems to give higher convergence rate compared to classic FEA. The findings from the studies of thin shells also suggest that IGA provides higher accuracy and convergence rate than classic FEA. A mesh refinement method, called k-refinement, that is available in IGA but does not exist in classic FEA proves an efficient way of improving the performance of IGA further. The implication of the results is that IGA outperforms classic FEA in terms of both accuracy and convergence rate. Furthermore, advantages in terms of time savings due to more efficient work flows are also discussed. The basis for discussion is analysis of a free-form thin shell sculpture. Having to return to the CAD model each time changes and mesh refinement is necessary in classic FEA proved to be a tedious task. Thus, the ability to perform adjustments and refinement directly on the isogeometric analysis model is a substantial asset.

Keywords: Isogeometric analysis, Finite element analysis, Curved beam elements, Thin shell analysis, Timoshenko beam theory, Euler-Bernoulli beam theory, Kirchhoff-Love shell theory, Integrated design and analysis

SAMMANFATTNING

För att designa strukturer med ökande geometrisk komplexitet måste nya analysverktyg utvecklas. Det iterativa arbetsflödet som praktiseras mellan ingenjörer och designers idag är inte tillräckligt. Rapporter från oberoende industrier konstaterar att 80% eller mer av analystiden spenderas på att konvertera *Computer Aided Design* (CAD) geometri till modeller som är lämpliga för analys. Denna process är, utöver att vara mycket tidskrävande, även benägen att generera modelleringsfel. Vid övergången från designmodeller till analysmodeller introduceras geometriska approximationer till följd av det byte av diskretisering som krävs för kompatibilitet med analysmjukvaror. En gemensam plattform för designers och ingenjörer att arbeta från är därför nödvändig. En teknik som utmärker sig genom sin förmåga att verkligen integrera analys med CAD är Isogeometrisk Analys (IGA). Det är en ung metod inom Finita Elementmetoder (FEM) som använder sig av samma matematiska bas för att beskriva geometri och basfunktioner som används inom CAD-teknologi, så kallad *Non-Uniform Rational B-Splines* (NURBS). I detta examensarbete undersöks fördelar och nackdelar med att använda IGA i integrerat konstruktionstekniskt och arkitektoniskt designarbete.

Frågeställningen är huruvida IGA ger mer precisa resultat och högre konvergenshastighet än klassisk FEM för samma antal frihetsgrader. I det syftet implementeras isogeometrisk analysmodeller för balkar baserade på Timoshenko och Euler-Bernoulli teori. Resultaten jämförs med resultat från klassisk FEM. Därtill implementeras en isogeometrisk analysmodell för tunna skal baserad på Kirchhoff-Love teori. För studierna av skal används modulen *Structural Mechanics* i analysprogrammet COMSOL Multiphysics[®] och programmen Abaqus[™], ETABS[®] samt Robot[™] för att ta fram FEM-resultat att jämföra med.

Resultaten från studierna av balkar tyder på att IGA ger högre precision än vad som åstadkoms med klassisk FEM med lika många frihetsgrader. Därtill ger möjligheten att utnyttja högre kontinuitet över elementgränser inom IGA högre konvergenshastighet jämfört med klassisk FEM. Resultaten från studierna av tunna skal tyder också på att IGA ger bättre approximationer och högre konvergenshastighet än klassisk FEM. En metod för nätförfining som finns tillgänglig i IGA men inte i klassisk FEM, så kallad *k-refinement*, visar sig vara ett effektivt sätt att ytterligare förbättra resultaten med IGA. Slutsatsen är att IGA ger både bättre precision och högre konvergenshastighet än klassisk FEM för samma antal frihetsgrader. Därtill undersöks även tidsbesparingar som kan göras till följd av mer effektiva arbetsflöden över yrkesgränser. Att vara tvungen att återgå till CAD-modellen varje gång förändringar och förfiningar av modellen krävs i klassisk FEM visar sig vara en utdragen process. Möjligheten att direkt införa förändringar och förfiningar i analysmodellen i IGA visar sig därför vara en väsentlig fördel.

PREFACE

This master's thesis of 30 credits was carried out as the final part of the master's program "Structural Engineering and Building Technology" at Chalmers University of Technology. The work took place during the spring of 2017 at the Department of Applied Mechanics. The examiner of the thesis was Prof. Fredrik Larsson. The work was supervised by Prof. Larsson together with Senior Lecturer Dr. Mats Ander.

ACKNOWLEDGEMENTS

To our examiner and advisor Prof. Fredrik Larsson. Thank you for steering us in the right direction from day one and for pushing us to excel. Your vast knowledge, especially within the field of shell theory, has been a major asset. We did not always understand the purpose of the tasks you gave us, but it has all come together beautifully in the end thanks to your keen eye. All of this in combination with your endless sharing and unwavering patience with all of our questions has made this project possible.

To our advisor and strongest supporter Dr. Mats Ander. Thank you for popping by our cubicle and encouraging us when things did not look so bright. Thank you for your dedicated proof-reading of the report and for your sharp inputs. Your detailed knowledge of beam theory in combination with your commitment to the project has been greatly appreciated.

Another special thanks goes out to Engineering Licentiate Vedad Alic from Lund University. We would not have gotten this far without your help. Thank you for debugging code, sending examples and for implementing the theory for handling arbitrarily curved boundaries. Because of your genuine interest in the project it has been a delight to turn to you in times of need.

Civil Engineer Rasti Bartek from Cundall is someone else who has our warmest thanks. Having you on board and getting input from an industry perspective from you has been truly valuable. Thank you for providing us with a case study subject, for running finite element analyses on the same and for pushing this project forward.

CONTENTS

Abstract	i
Sammanfattning	ii
Preface	iii
Acknowledgements	iii
Contents	v
1 Introduction	1
1.1 Context	1
1.2 Purpose	4
1.3 Method	5
1.4 Limitations	5
2 Basics of Isogeometric Analysis	6
2.1 NURBS as geometric descriptions	6
2.1.1 Defining NURBS	6
2.1.2 Basis functions	9
2.1.3 Curves	12
2.1.4 Refining NURBS	13
2.1.5 Surfaces	15
2.2 The isogeometric finite element formulation	17
2.2.1 Analysis spaces	17
2.2.2 Finite elements with IGA	18
2.2.3 Shortcomings of NURBS based IGA	20
3 Practical aspects of implementation	21
3.1 Code structure	21
3.2 Connectivity matrices	22
3.3 Mesh refinement	23
3.4 Boundary conditions	23
3.5 Post-processing	23

4	Curved Beams	24
4.1	Notations	24
4.2	Timoshenko beam theory	25
4.3	Euler-Bernoulli beam theory	27
4.4	Shear and membrane locking	29
4.5	Section forces	30
4.6	Example	31
4.6.1	Problem setup	31
4.6.2	Analysis model	33
4.6.3	Results	34
5	Thin shells	39
5.1	Notations	39
5.2	Kirchhoff-Love shell theory	39
5.2.1	Preliminaries	39
5.2.2	The shell geometry	40
5.2.3	Kinematics	43
5.2.4	The Kirchhoff-Love hypothesis	43
5.2.5	Small strain kinematics	45
5.2.6	Equilibrium configuration	45
5.2.7	Elastic constitutive relations	46
5.2.8	Galerkin Discretization	47
5.3	Membrane locking	48
5.4	Section forces	48
5.5	Boundary conditions in the IGA shell model	48
5.6	Examples	49
5.6.1	Problem setup	49
5.6.2	Analysis model	51
5.6.3	Results	52
6	Case Study	64
6.1	Description	64
6.2	Limitations and simplifications	66
6.3	Analysis	67
6.4	Results	68

6.5 Workflow	78
7 Conclusions	79
7.1 Accuracy and efficiency	79
7.2 Potential for integrated design	79
7.3 Suggestions for further work	80
References	81
Appendices	i
A Second derivative of shape functions w.r.t. physical space coordinates	i
B Method for eliminating shear and membrane locking in classic FEA	iii
C Change of coordinate system in the beam models	v
D Change of coordinate system in the shell model	vi
E Enforcing fixed boundary and symmetry conditions in IGA	vii

1 Introduction

1.1 Context

There is a young approach in Finite Element Analysis (FEA). It is called Isogeometric Analysis (IGA) and is a viable alternative to standard polynomial-based FEA. The need for IGA originally sprung from the difference in how designers and engineers work. The models used for design and analysis are not inherently compatible because the analysis programs used by engineers are based on different mathematical discretizations than design software applications. The hinder in work flow over professional boundaries that is created by this leads both to geometric inaccuracy of analysis models and time consuming meshing processes. Interestingly, this gap exists in spite of the fact that analysis is underpinned by geometry and not the other way around. The explanation is that Computer Aided Design (CAD) as we know it today was developed later than classic FEA. However, by changing the way analysis is done through using the same underlying mathematical description as design applications use, the problem is dealt with before it even arises. Thus, the possibility to truly integrate the domains of design and analysis opens up [1], [2]. As stated by Hughes [3], who was the first to introduce the approach back in October 2005: "IGA expands the island of computational techniques to include a method that is both usable and ideal for design applications".

Today, isogeometric analysis can play a large role in early schematic engineering work, for example in form-finding. Further down the road it has the potential to completely replace commercial FEA for various applications. Many fields stand to benefit from the method, such as the aviation and automobile industry as well as the building industry. One area that is highly impacted by the new technology is analysis of shells. The new method has made it possible to resume the numerical study of thin shells based on classical Kirchhoff-Love shell theory. It was abandoned early on in finite element formulations due to the high continuity over element boundaries that the theory requires, something that is easily overcome in IGA [3]. In structural engineering, we are particularly interested in shell structures in the context of the built environment. These structures already play an interesting role in the integration of architectural and structural design. The engineer's understanding of structures is essential because the shapes of shell structures are directly derived from flow of forces. But the architect's understanding of context and culture is what makes the building an addition to society. Hence, a collaborative effort is essential in order to achieve successful designs.

The Hamburg Alster-Schwimmhalle, see Fig. 1.1, is an example of successful collaboration. The building's roof consists of a hyper shell and the architect required that the edge beams should be cantilevering beyond the facade. According to classical theory for membrane dominated shells this is not possible, something that can be seen when looking at the church of San José Obrero in Fig. 1.2. In order to meet the architectural intent the shell was designed by superimposing the classical saddle surface for the surface loads with the straight line generators' surface for edge loads. The result is a perfect structure that developed only because of the asserted architect and the skilled engineer [4].

L'Oceanografic is another shell example from the built environment. It is the re-elaboration of a thin shell by Felix Candela. The shape is intricate and it is impressive in its light expression. Fig. 1.3 shows the shell during the construction phase and Fig. 1.4 the finished building. Again, extensive engineering work goes into realizing a shell like this with its remarkable slenderness of 6 cm thickness over a 40 m span [5].

With the increased freedom of design that CAD software enables greater geometrical complexity is pursued. It is evident that in order to enable realization of these increasingly complex designs analysis must keep up. The situation today where meshes are generated from CAD data is not sustainable. Not only does it result in a different geometric description that is only approximate of the original design, it is also highly time-consuming. In several industries it is estimated that 80% or more of the total time modeling and only 20% on the analysis itself. Thus, the need for a technique that takes the exact geometry and effortlessly meshes its curves, surfaces or solids is urgently needed [3].



Figure 1.1: Cantilevering shell of Hamburg Alster-Schwimmhalle. Photograph from [6].



Figure 1.2: Church of San José Obrero by Félix Candela. Note the need for vertical support of the edge beams. Photograph from [7].



Figure 1.3: Roof structure of L'Oceanografic in Valencia during construction. Photograph from [5].



Figure 1.4: L'Oceanografic after completion. Photograph from [5].

For a new technique to succeed commercially the underlying discretisation must be either directly compatible with or easily converted from CAD technology. In isogeometric analysis this is facilitated through use of the same mathematical basis for description as is used in dominating design software, i.e. Non-Uniform Rational B-Splines (NURBS) [8]. Independent of how successful the adoption of IGA eventually becomes for everyday engineering tasks it is a highly current topic with great potential and ideal for efforts of bringing architectural and structural design together.

In this report, the basics of NURBS and how they are used in the implementation of isogeometric analysis are explained. Additionally, differences in the implementation of IGA compared to FEA are addressed. It is assumed that the reader has prior basic knowledge of finite element analysis based on the Bubnov-Galerkin method, in this report referred to as classic FEA [9], [10]. It is also necessary to have some understanding of differential geometry of surfaces to fully understand equations and derivations, a quick review of Ch. 2.4. of Kiendl's thesis [11] is sufficient. After implementation a couple of basic examples are given and then the new-found knowledge is applied to analysis of an existing project. The report should be read as a summary of the thesis project as well as a "how-to" guide for basic implementation of IGA for anyone without previous knowledge of the subject. The main claim investigated is that IGA is more accurate on a per-degree-of-freedom basis compared to classic FEA. To this end IGA is applied to problems within the field of structural engineering. Structural elements that are difficult to analyze with classic FEA are studied and the results are compared with the traditional approach.

1.2 Purpose

The purpose of the project can be split into two parts. Part one consists of a thorough guide to implementing IGA. This is something that the authors found lacking while working with the project. Most of the available literature on the topic deals with complex problems which makes it harder to get a comprehensive basic understanding of IGA. To achieve this the simple examples in 1D that were investigated while studying IGA and implementing the necessary algorithms for the final case study are presented in-depth. Part two consists of using the implemented IGA shell model to analyze an existing project. Throughout the project the advantages and disadvantages of using IGA over a standard FE-analysis are studied.

For the final case study IGA with Kirchhoff-Love shell theory is implemented. Analysis according to this theory can not be performed using classic FEA but has been shown to be both robust and perform well under large deformations with IGA. Thus, by choosing IGA with Kirchhoff-Love theory the possibility to analyze problems that could not have been dealt with in the same way otherwise opens up [12].

Based on current research the hypothesis is that IGA will prove more time efficient and yield more accurate results than classic FEA on a per-degree-of-freedom basis. The main reasons include elimination of the step of converting geometry from CAD to classic FEA meshes and avoiding the geometrical error that is otherwise introduced by classic FEA basis.

The sub-tasks devised to build theoretical and implementation knowledge in part one are:

1. *Task:* Analyze a simple bar with varying distributed axial load using IGA and classic FEA.
Purpose: Master basic NURBS theory and differences in notation compared to classic FEA, as well as basic implementation of IGA (results and implementation are omitted from report).
2. *Task:* Derive equations for an arbitrarily curved beam in 1D using both Timoshenko and Euler-Bernoulli beam theory and analyze using IGA and classic FEA.
Purpose: A first step towards shells through curved elements, handling of locking phenomena in IGA and a hint to whether the hypothesis will be confirmed (see Ch. 4).
3. *Task:* IGA with Kirchhoff-Love shell theory of benchmark shell problems.
Purpose: Study Kirchhoff-Love shell theory and master implementation in 2D as well as verifying the developed model (see Ch. 5).

The case study is performed for a thin shell. The geometry consists of a single patch NURBS surface and it is analyzed with Kirchhoff-Love shell theory. The chosen project is called "Crest" and is a sculpture designed by Zaha Hadid architects in collaboration with BuroHappold Engineering and LiteStructures. Deflections and stresses under the impact of its own self-weight are studied.

1.3 Method

Review of articles and other literature on the topic is done throughout the thesis work both to build theoretical knowledge and to provide guidance for analysis model decisions. For the subtasks the software Matlab[®] is used to perform complex operations and to practice algorithmic thinking in relation to IGA. The main analysis program is built from scratch with some subroutines by supporting supervisor Alic and shape function routines and inspiration from the open source Matlab[®] code package "igafem" by Nguyen et al. [8]. The software Rhinoceros[®] by Robert McNeel & Associates and the plug-in Grasshopper are used to extract NURBS information through a self-implemented Grasshopper[®] component.

1.4 Limitations

For isogeometric analysis, the use of any other basis than NURBS is omitted from the project. Research has shown that T-spline basis exhibits better properties than NURBS basis, for example by enabling "watertight" compatibility between multiple patches [13]. However, using T-splines adds an extra level of complexity to the project and the basis is yet to be fully implemented in design software. Additionally, the advantages of using T-splines are not relevant for this project as only single patch geometries are analyzed.

Other numerical methods than Gauss point integration are also omitted from the analysis models. Gauss point integration is neither optimal nor the most computationally efficient method for IGA, but it is commonly used for FEA and provides a good foundation for comparing the methods [8].

Additionally, the analyses are limited to linear elastic problems inflicted by point, line and/or body loads.

2 Basics of Isogeometric Analysis

2.1 NURBS as geometric descriptions

There are three different types of mathematical descriptions of curves and surfaces: the explicit, the implicit and the parametric description. NURBS curves and surfaces belong to the parametric descriptions category. It is the most relevant group of mathematical descriptions when it comes to free-form geometries. NURBS as geometric description is a powerful tool that has become standard in CAD modelling due to its versatility in the ability to represent smooth free form shapes as well as linear shapes, sharp edges, geometric objects such as spheres and more. Since NURBS constitute the basis of isogeometric analysis, the first step towards understanding the analysis method is getting used to NURBS. Any NURBS geometry is defined by four parts:

1. a knot vector,
2. a list of control points,
3. corresponding weights,
4. and a polynomial degree.

All of these will be briefly reviewed in the next section. A more detailed explanation is given by Piegl and Tiller [14].

2.1.1 Defining NURBS

Knot vector

A knot vector is a set of ascending values corresponding to coordinates in parameter space. The vector is denoted by:

$$\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\} \quad \text{where} \quad \xi_i \leq \xi_{i+1} \quad \text{and} \quad \xi_i \in \mathbb{R}. \quad (2.1)$$

The subscript i denotes the i^{th} knot, n is the number of basis functions and p denotes the polynomial degree. The knot vector defines the parametric space and its knot intervals divides the same into so called knot spans. The knot spans of non-zero length represent elements in the physical space. Note that the term is not equivalent with elements in classic FEA. If the knots are equally spaced the knot vector is said to be uniform, otherwise it is non-uniform. If the same value appears m times, i.e. $\xi_i = \xi_{i+1} = \dots = \xi_{i+m-1}$, the knot is said to have multiplicity m . By increasing the multiplicity of a knot the continuity at its location can be reduced. Most commonly knot vectors with multiplicity $m = p + 1$ of the first and last knots are used. These are called open knot vectors. They are standard in CAD literature because they allow the start and end point of a curve to be specified by the designer [8],[2],[11].

An example: The NURBS curve shown in Fig. 2.1 (1a) has the knot vector $\Xi = \{0, 0, 0, 0, 1, 1, 1, 1\}$. It is uniform because the unique knots $\{0, 1\}$ are equally spaced. The end knots have multiplicity $m = 4$. If it has degree $p = 3$ it is also an open knot vector since the multiplicity of the end knots is $m = p + 1 = 4$. The curve has one element since there is only one non-zero knot span. The same curve can be represented with more elements as shown in Fig. 2.1 (1b). Then the knot vector becomes $\Xi = \{0, 0, 0, 0, 0.5, 1, 1, 1, 1\}$. The unique knots $\{0, 0.5, 1\}$ define two non-zero spans and thereby two elements. This configuration represents the exact same geometry. Note that knots are different from nodes in classic FEA and should not be used interchangeably.

Control points and weights

Control points together with their corresponding weights control the NURBS geometry. A linear interpolation between the control points is called the control polygon. When defining the control points and weights of a NURBS geometry the information is usually stored in matrices. For this project the coordinates and weights are structured in a matrix with $\mathbf{c} = [\mathbf{x} \ \mathbf{y} \ \mathbf{z} \ \mathbf{w}]$ where \mathbf{x} , \mathbf{y} , \mathbf{z} and \mathbf{w} are column vectors containing the space coordinates and the weights. The row index of \mathbf{c} correspond to the control point index. Control points can be grabbed and pulled and the geometry is thereby changed, see (2a) and (2b) in Fig. 2.1. The new configuration of the curve in (2a) can also be achieved by decreasing the value of the weight instead of changing the location of the control point. If the weight had been increased the curve would have been pulled towards the control point and if it had been lowered the curve would have moved in the opposite direction. The impact of both actions is compact and contained on certain intervals. The specifics will be further discussed in the section about basis functions. However, the ability to locally influence NURBS geometries by changing the position of control points is visualized in (2b). Note that only the left element is affected by the change in the control polygon. When a control point is pulled things move in an monotone fashion without oscillations. This monotone behaviour is critical in design and very important for analysis, furthermore it is not present in classic FEA [8],[2],[11].

Polynomial degree

There are different definitions of the polynomial degree. The algorithms in this project are based on the definition $p = k - n - 1$ where k is the number of knots and n the number of control points. This seems to be the most common definition in IGA. The terms order and degree of NURBS are often used interchangeably in literature but in its strict definition the order equals $p + 1$ [15]. Changing the order without changing the knot vector, control points or weights changes the geometry as shown in Fig. 2.1 (3a) [8],[2],[11].

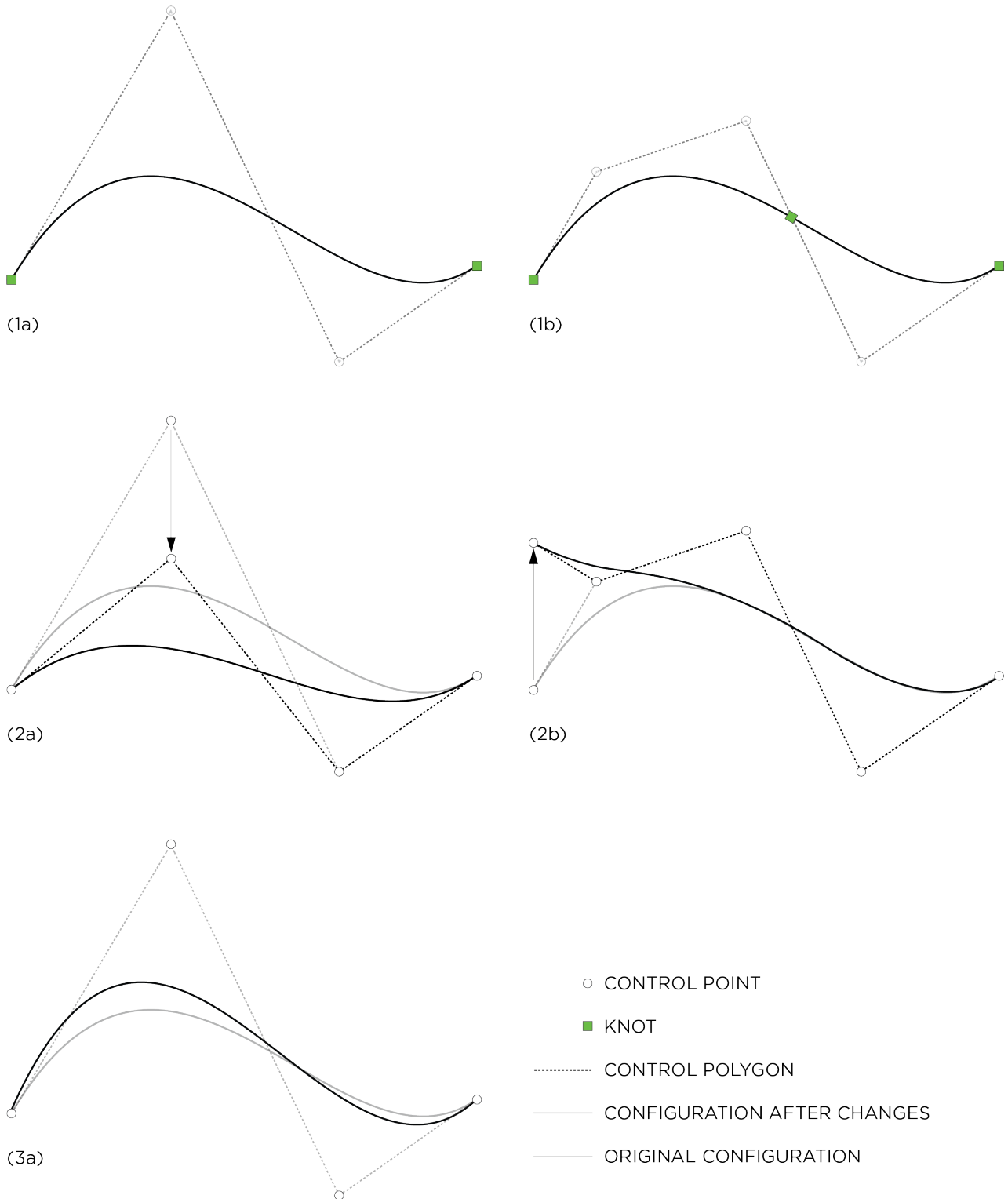


Figure 2.1: (1a): A NURBS curve with one non-zero knot span. (1b): The same NURBS curve with two non-zero knot spans or "elements". Note that the geometry remains the same when the number of elements is increased. (2a): The same curve as in (1a) but with a control point pulled downwards, dark lines show the new configuration. (2b): The same curve as in (1b) but with a control point pulled upwards. Note the local impact on the shape of the curve, the element to the right is not affected. (3a): The same curve as in (1a) but with the order of the curve lowered, the new configuration is shown with dark lines.

2.1.2 Basis functions

The key to eliminating geometrical errors, as is the main the superior attribute of IGA, is to use NURBS for meshing and as basis functions for the analysis. Non-Uniform Rational B-Splines are, as the name suggests, a generalization of B-splines. A good base for NURBS-discussion is therefore to first address B-splines. In contrast to NURBS, B-splines do not have weights attached to their control points. However, they are defined by the knot vector, polynomial degree and control points just as NURBS. Fig. 2.2 displays the basis functions for a B-spline and a NURBS curve respectively. The only difference between the curves is that the weight of the fourth control point is changed for the NURBS curve.

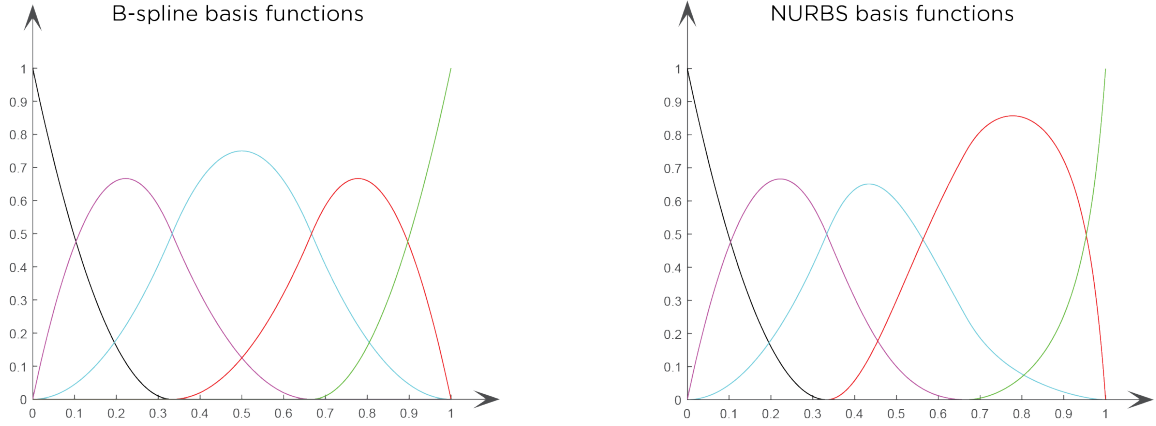


Figure 2.2: Basis functions for a B-spline curve on the left and NURBS basis for the same curve but with increased weight of the fourth control point on the right.

B-spline basis functions

Given a knot vector, the zeroth order B-spline basis functions are given by:

$$N_{i,0} = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} . \quad (2.2)$$

For polynomial orders $p \geq 1$ the basis functions are defined recursively by the Cox-de Boor recursion formula:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) . \quad (2.3)$$

The first derivative of B-spline basis functions are computed by:

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) . \quad (2.3)$$

Basis function properties

There are several important properties of the B-spline basis to take note of. Often highlighted attributes include [8], [2], [1]:

1. they constitute partition of unity, $\sum_i N_{i,p}(\xi) = 1$ for any p ,
2. each basis function is non-negative over the entire domain, $N_{i,p}(\xi) \geq 0$,

3. they are linearly independent,
4. the support of a B-spline basis function $N_{i,p}$ is compact and contained in the interval $[\xi_i, \xi_{i+p+1}]$,
5. they exhibit C^{p-m} continuity across knot ξ where m is the multiplicity of the knot.

Continuity

The fifth property, continuity across knots, is visualized in Fig. 2.3 (1a). The knot vector is given on the x-axis and the continuity over element boundaries are given by C^{p-m} . From the knot vector it can be seen that the degree is $p = 4$ and that the end knots have multiplicity $m = 5$, thus the end knots have $C^{4-5} = C^{-1}$ continuity. In the same way it can be seen that there is $C^{4-1} = C^3$ continuity between element 1 and 2.¹

Continuity over element boundaries is, as illustrated, controlled by the knot vector. The ability to easily control continuity is one of the key aspects that for example enable implementation of the Kirchhoff-Love shell theory. What continuity means in terms of the geometry is visualized in Fig. 2.3 (2a)-(2d). For example, the C^0 continuity between the fourth and fifth element means that the end of the fourth element has the same position as the start of the fifth element. This is how one introduces kinks in NURBS curves. Note that all basis functions except for one disappear at this location, the one that remains takes the value 1. Where there is C^1 continuity, i.e. between the third and the fourth element, the tangent of the curve is the same on both sides of the element boundary in addition to positional continuity.

NURBS basis functions

NURBS basis functions are constructed from the B-spline basis and NURBS weights through:

$$R_i^p = \frac{N_{i,p}(\xi)w_i}{\sum_{i=1}^n N_{i,p}(\xi)w_i} . \quad (2.4)$$

The weights play an important role in defining the basis. Note that the NURBS basis coincides with the B-spline basis if all the weights are equal to each other (following the partition of unity property of B-splines). In other words, B-splines are a special case of NURBS. As one might expect from this, NURBS basis bear much in common with the B-spline basis. The continuity and support follows directly from the knot vector as before. Additionally, the basis still constitutes partition of unity and remains non-negative over the entire domain.

Derivatives of B-splines basis functions are efficiently represented in terms of lower order B-spline basis, as follows naturally from its recursive definition. Given that the NURBS basis is constructed from the B-spline basis it also follows logically that NURBS basis derivatives are constructed from B-spline derivatives. Hughes et al. [2] give a good overview of B-spline and NURBS basis derivatives and Piegl and Tiller [14] present an in-depth analysis. Here the equation for the first derivative of the NURBS basis \mathbf{R} with regards to the parametric coordinate ξ is given:

$$\frac{d}{d\xi} R_i^p(\xi) = w_i \frac{W(\xi)N'_{i,p}(\xi) - W'(\xi)N_{i,p}(\xi)}{(W(\xi))^2} , \quad (2.5)$$

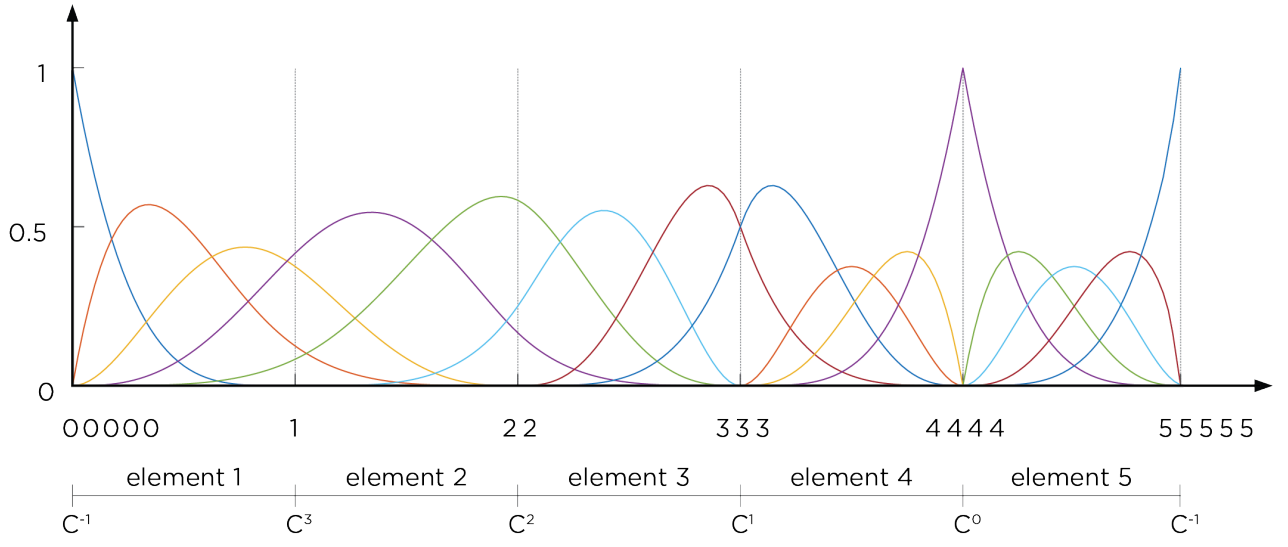
where $N'_{i,p}(\xi)$ is defined as:

$$N'_{i,p}(\xi) \equiv \frac{d}{d\xi} N_{i,p}(\xi) . \quad (2.6)$$

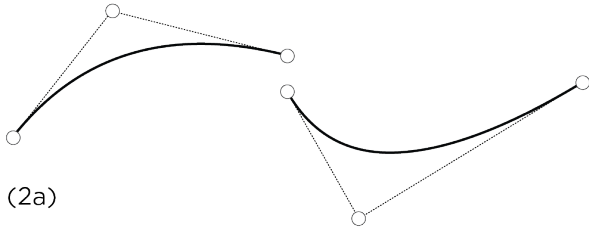
Furthermore, $W'(\xi)$ is given by the sum:

$$W'(\xi) = \sum_{i=1}^n N'_{i,p}(\xi)w_i . \quad (2.7)$$

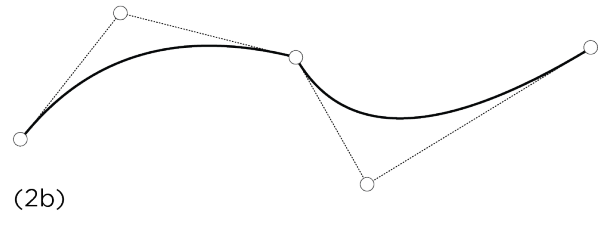
¹Note that continuity at end knots is a matter of definition



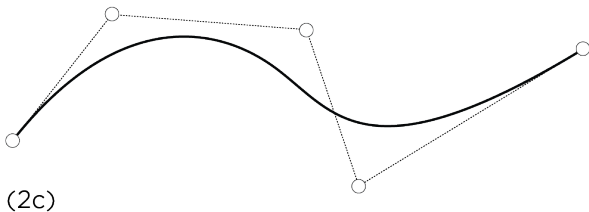
(1a)



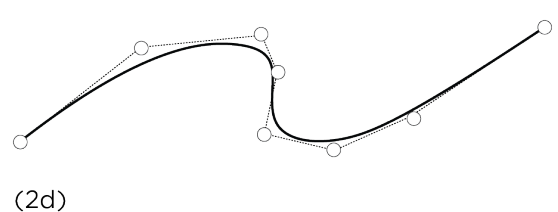
(2a)



(2b)



(2c)



(2d)

Figure 2.3: (1a): Continuity over element boundaries in IGA. The basis functions are plotted with the knot vector listed on the horizontal axis. (2a): Discontinuity, C^{-1} . (2b): Positional continuity, C^0 . (2c): Tangential continuity, C^1 . (2d): Continuity in curvature, C^2 .

2.1.3 Curves

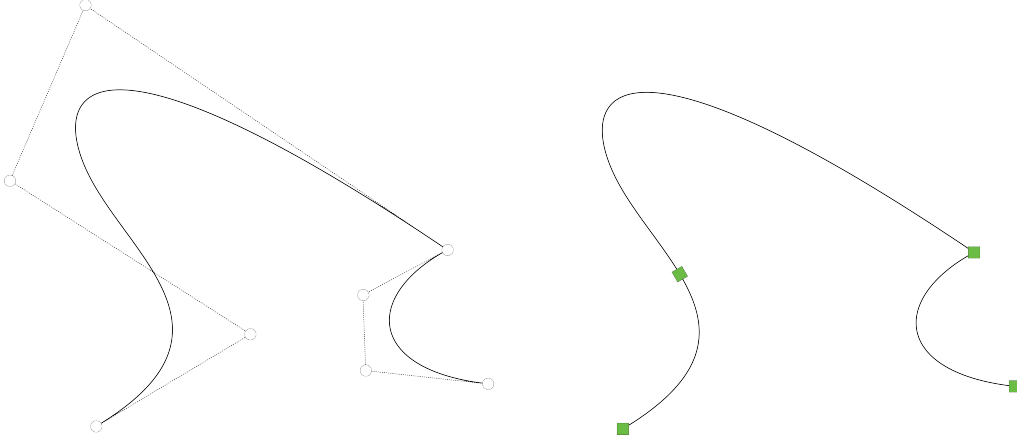


Figure 2.4: A NURBS curve with control points and control polygon displayed on the left. The knots mapped to the physical space (onto the curve) so that the non-zero knot spans divide the curve into elements on the right.

B-spline curves

B-spline curves are defined as a linear combination of control points and B-spline basis functions. The coefficients of the basis functions are the earlier mentioned control points. Given n basis functions and corresponding control points $\mathbf{B}_i \in \mathbb{R}^d$ with $i = 1, 2, \dots, n$ a piece-wise polynomial B-spline curve is given by [2]:

$$\mathbf{C}(\xi) = \sum_{i=1}^n N_{i,p} \mathbf{B}_i . \quad (2.8)$$

NURBS curves

Given a B-spline curve and its control points, the control points of the NURBS curve are given by:

$$(\mathbf{B}_i)_j = (\mathbf{B}_i^w)_j / w_i, \quad j = 1, \dots, d . \quad (2.9)$$

The control point $(\mathbf{B}_i)_j$ is the j^{th} component of the vector (\mathbf{B}_i) and w_i is the i^{th} weight. With Eqs. (2.4) and (2.9) a NURBS curve can be expressed as [2]:

$$\mathbf{C}(\xi) = \sum_{i=1}^n R_{i,p} \mathbf{B}_i , \quad (2.10)$$

this is the NURBS equivalence to Eq. (2.8). Fig. 2.4 shows a NURBS curve with the open non-uniform knot vector $\Xi = \{0, 0, 0, 0, 1, 2, 2, 2, 3, 3, 3, 3\}$. The control points are denoted by white circles in the left picture and linear interpolation of the control points give the control polygon. The right picture shows the knots mapped to the physical space as green squares, the line segments represent the elements.

2.1.4 Refining NURBS

In order to use NURBS as basis for analysis in isogeometric analysis the possibility to refine meshes must exist. The refinement space for B-splines, and by extension NURBS, is rich. Changes can be done to the basis by controlling element sizes, continuity of basis over element boundaries and degree of curves while leaving the geometry both geometrically and parametrically intact. The three ways of refinement in IGA are: h-refinement, p-refinement and k-refinement.

h-refinement

In IGA, h-refinement is performed by insertion of knots. A new knot can be chosen or an existing one repeated. However, the latter leads to a decrease in continuity of the basis. Note that if an internal knot appears more than p times the curve becomes discontinuous. Given a knot vector as presented in Eq. (2.1) a new knot is defined by $\bar{\xi} \in [\xi_k, \xi_{k+1}]$.

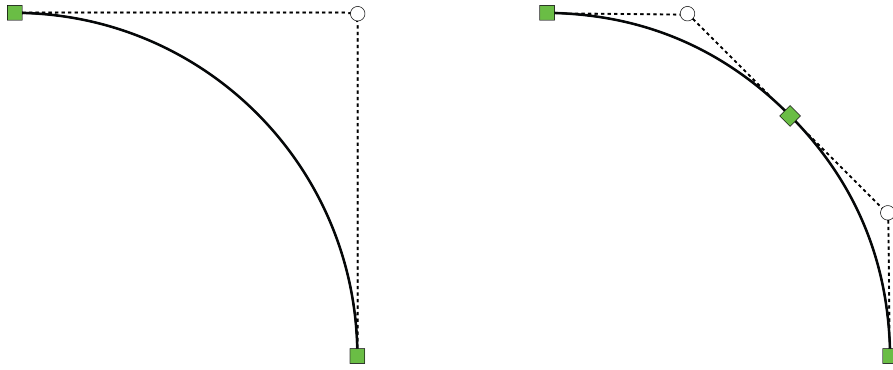


Figure 2.5: NURBS curve before h-refinement (left) and after (right).

This gives $n + 1$ new basis functions which are calculated using Eqs. (2.2) and (2.3). The new $n + 1$ control points are calculated from the original control points through:

$$\bar{B}_i = \alpha_i B_i + (1 - \alpha_i) B_{i-1} , \quad (2.11)$$

where

$$\alpha_i = \begin{cases} 1 & \text{if } 1 \leq i \leq k - p . \\ \frac{\bar{\xi} - \xi_i}{\xi_{i+p} - \xi_i} & \text{if } k - p + 1 \leq i \leq k . \\ 0 & \text{if } k + 1 \leq i \leq n + p + 2 . \end{cases} \quad (2.12)$$

An example is shown in Fig. 2.5 where a new knot $\bar{\xi} = 0.5$ is inserted in the knot vector $\Xi = \{0, 0, 0, 1, 1, 1\}$. The new curve is identical to the original one but more elements and control points are added. This leads to more basis functions of the same order, an enrichment of the base. The strategy is analogous to h-refinement in classic FEA [1].

p-refinement

When the polynomial order of basis functions is increased without changing the curve geometrically or parametrically it is called p-refinement. The process involves subdividing the curve by knot insertion, elevating the order of each segment, and removing unnecessary knots before combining the segments again. Piegl and Tiller give an in-depth explanation [14]. The strategy is analogous to p-refinement in classic FEA [1].

k-refinement

The k -refinement technique differs from the other two in that there is no analogue refinement method in classic FEA. The order of the curve is elevated from p to q and unique knot values are inserted. This leads to a basis with $q - 1$ continuity at the new knots, i.e. the highest continuity that can be obtained with NURBS over element boundaries. The operations must be performed in the order described because the process of order elevation and knot insertion is not commutative, see Fig. 2.6. Hughes, Cottrell and Bazilevs [1] state that k -refinement is important and potentially superior for high-precision analysis compared to p -refinement.

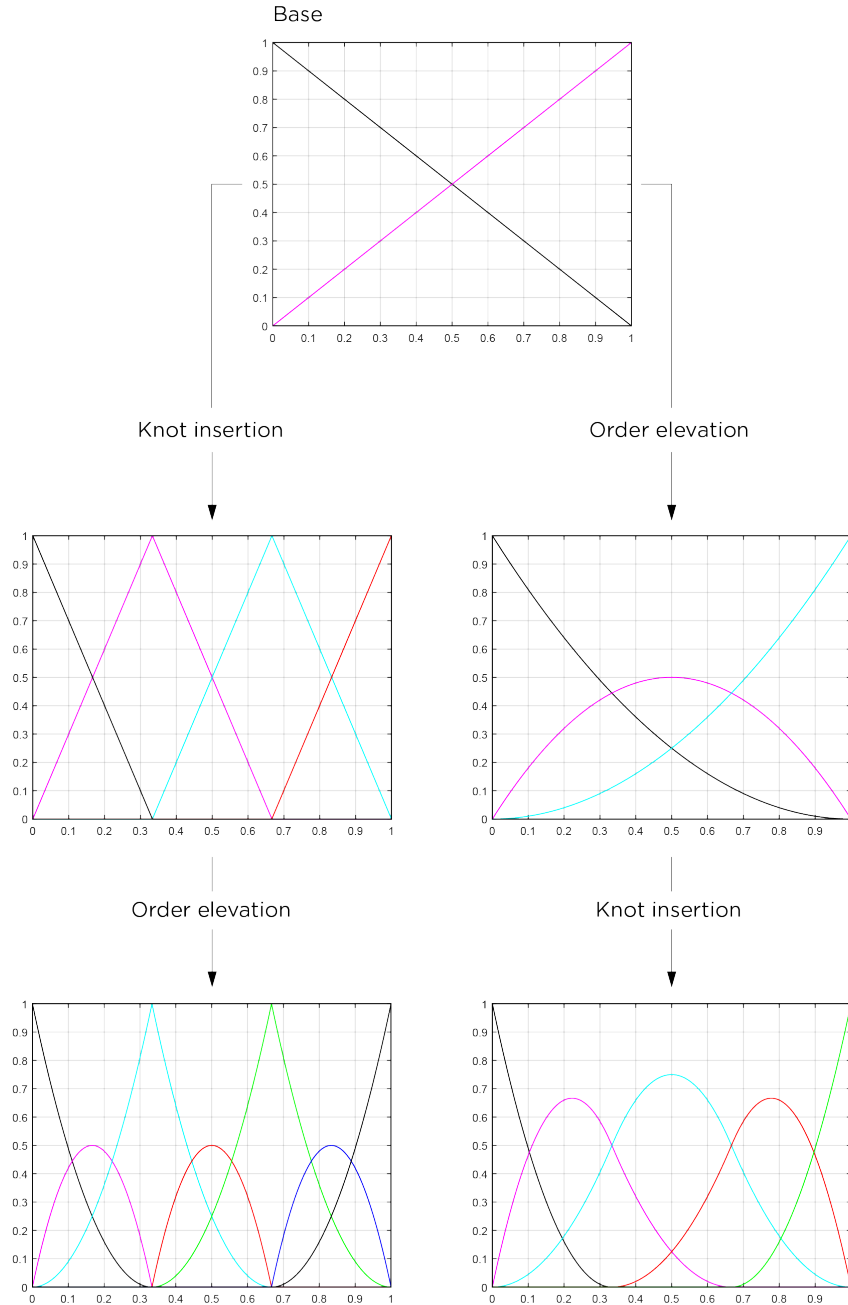


Figure 2.6: The right column displays order elevation and then knot insertion, the refinement method is called k -refinement and continuity over element boundaries is maintained. It is not commutative and the result from first doing knot insertion and then order elevation is not the same, as seen on the left.

2.1.5 Surfaces

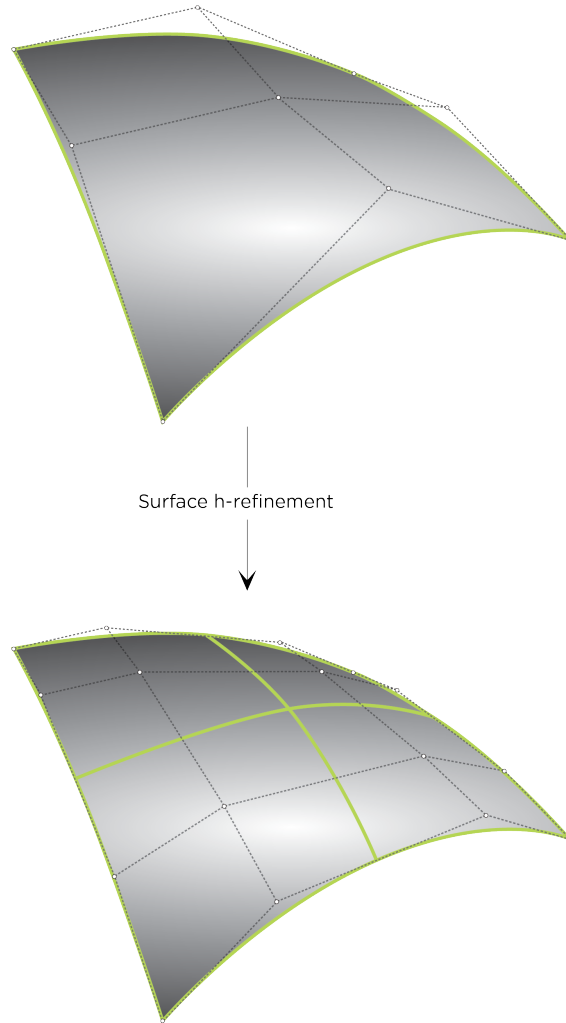


Figure 2.7: A NURBS surface and its control net. Original configuration on the top and configuration after h -refinement on the bottom.

There are four types of NURBS surfaces. The general cylinder, for example, is a NURBS surface consisting of a NURBS curve swept a distance d along a vector. The ruled surface is constructed through linear interpolation between two NURBS curves with points of equal parameter values. The surface of revolution is constructed from a NURBS curve which is revolved through an arbitrary angle about an arbitrary axis. Additionally, there are NURBS surfaces that are simply made up from two or more NURBS curves. Again, a more elaborate description is given by Piegl and Tiller [14].

B-spline surfaces

B-spline surfaces are defined as a tensor product of two curves. Given that a surface has two dimensions, two knot vectors and their respective polynomial orders are required. What was previously the control polygon is now a so called control point net [1].

Given the knot vectors $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ and $\mathcal{H} = \{\eta_1, \eta_2, \dots, \eta_{m+q+1}\}$ with polynomial orders p and q ,

as well as the control net $\{\mathbf{B}_{i,j}\}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$ with $n \times m$ control points, a B-spline surface is given by:

$$S(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi) M_{j,q}(\eta) \mathbf{B}_{i,j} . \quad (2.13)$$

The basis function notation $N_{i,p}(\xi)$ is the same as was used Ch. 2.1.2. The corresponding basis functions for the knot vector \mathcal{K} are denoted $M_{j,q}(\eta)$. The tensor product nature gives the B-spline surface various properties. For instance, the combined basis is point-wise non-negative and constitutes partition of unity. Furthermore, the local support of basis functions follow directly from the one-dimensional curves that form them. For a surface an element is defined as the non-zero knot spans $[\xi_i, \xi_{i+1}] \times [\eta_j, \eta_{j+1}]$ [1].

NURBS surfaces

NURBS surfaces are defined in the same way as B-spline surfaces, but with control points as given in Eq. (2.9) and with rational basis functions [2]:

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi) M_{j,q}(\eta) w_{i,j}}{\sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi) M_{j,q}(\eta) w_{i,j}} . \quad (2.14)$$

Giving the NURBS equivalent to Eq. (2.13):

$$S(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m R_{i,j}^{p,q}(\xi, \eta) \mathbf{B}_{i,j} . \quad (2.15)$$

Just as for curves, NURBS surfaces share properties with B-spline surfaces. Fig. 2.7 depicts a NURBS surface with corresponding control net \mathbf{B} . Each of the NURBS curves that the surface is constructed from can be refined with the refinement methods described. The bottom picture shows the same NURBS surface but after h -refinement. Note how the control net approaches the surface.

2.2 The isogeometric finite element formulation

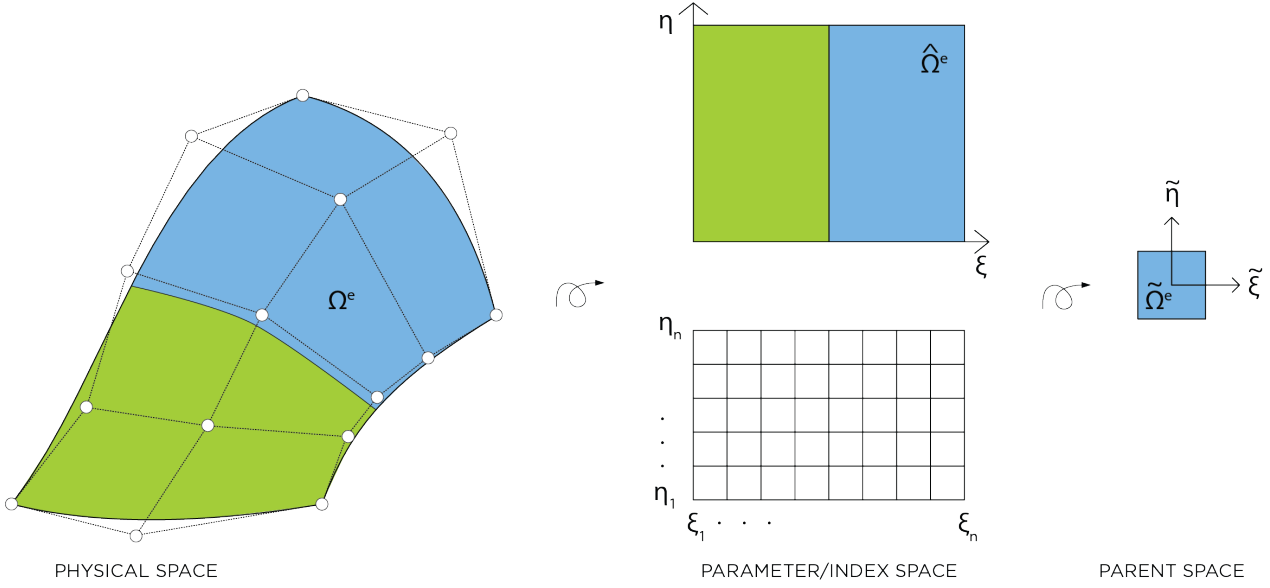


Figure 2.8: An overview of analysis spaces and mapping between them in IGA.

So far NURBS in their natural setting have been treated, but only touching upon what they are used for. NURBS are the foundation for bringing geometry and analysis together in IGA. As explained by Hughes, Cottrell and Bazilevs [1], the NURBS basis is part of the isogeometric finite element formulation in the following ways:

- to generate a mesh that, for a single NURBS patch, is defined by one or a product of multiple knot vectors,
- define elements by subdivisions of the domain created by knot spans,
- generate basis functions with local support on a small number of elements,
- define the geometry through control points,
- through the isoparametric concept, meaning that the same basis is used for geometry and analysis,
- and to create refined meshes.

2.2.1 Analysis spaces

In IGA, there are three relevant analysis spaces: the physical space, the parametric/index space and the parent space. In Fig. 2.8 the spaces are visualized for a 2D NURBS surface. On the left the surface and the element subdivision is shown as well as the control polygon. The whole patch is transformed at once through geometrical mapping to parameter space, this is in contrast to classic FEA where transformations are done elementwise. Transformation to parent space is done elementwise through affine mapping.

Integration by Gaussian quadrature is performed on one element at a time in parent space. The element is then pulled back to the parametric domain where the majority of the analysis takes place, e.g. calculating the basis functions, their gradients and the Jacobian determinant of the pullback. The index space is used to identify nonzero basis functions. The whole patch is then pulled back to the physical space. After solving the

system of equations, post-processing is required to obtain results on the geometry itself. Seldom do the control point coefficients coincide with what is sought, especially since control points differ from classic FEA nodes in that they do not lie on the geometry itself [2].

2.2.2 Finite elements with IGA

The isogeometric analysis in this project consist of NURBS based Galerkin finite elements. However, many different numerical solution methods have been explored for IGA and some are better suited than others. Galerkin finite elements are not necessarily the best method, but it is common in classic FEA and by choosing it parallels can more easily be drawn between IGA and classic FEA.

Comparing IGA with classic FEA

While IGA and classic FEA share a lot, many terms and their meaning differ between the two methods. By being similar but yet different, terms used in IGA can be confusing for the reader educated in FEA. Some of the things that seem similar but lead to faulty assumptions if used interchangeably are presented by Hughes, Cottrell and Bazilevs [2]:

IGA	Classic FEA
Exact geometry	Approximate geometry
Control points	Nodal points
Control variables	Nodal variables
Basis not interpolatory between control points and variables	Basis interpolatory between nodal points and variables
NURBS basis	Polynomial basis
High, easily controlled continuity	C^0 continuity
<i>hpk</i> -refinement space	<i>hp</i> -refinement space
Pointwise positive basis	Basis not necessarily positive

The isoparametric concept

One of the things shared by IGA and classic FEA is the isoparametric concept. The basic idea is to use a basis that exactly models the geometry for the solution space of the numerical method. It is the foundation for IGA and it is common in FEA as well, only with a basis that approximates the geometry instead of one that mirrors it. In other words, in classic FEA the solution fields are imposed on the geometry whereas in IGA the geometry is imposed on the fields [2]:

$$IGA : \text{Geometry} \rightarrow \text{Fields}$$

$$FEA : \text{Geometry} \leftarrow \text{Fields}$$

Numerical solution methods

Most of FEA is based on the Bubnov-Galerkin method. The steps are well-known to the reader who has studied finite element methods, starting with defining the weak form and then using Galerkin's method to construct finite-dimensional approximations by multiplication with a weighting function. This leads to a coupled system of linear algebraic equations the solution space consists of all linear combinations of a given set of basis functions.

Beside Galerkin finite elements, collocation, least-square and meshless methods are some examples of numerical methods that have been adopted by IGA. It has been shown that IGA lends advantages over classical FEA to some of these methods. However, since none of these methods are incorporated in this thesis project the theory is not covered either. One issue that needs to be addressed in relation to this is efficient quadrature. Since Gauss point integration is used to solve the linearized system it should be discussed how IGA behaves with different quadrature rules [2].

Efficient quadrature

Gaussian quadrature rules are optimal for polynomials in one dimension. They are commonly used for other elements as well even though more efficient spacial rules have been derived for several elements. The study of efficient quadrature rules for NURBS based IGA was initiated by Hughes, Reali and Sangalli [16].

Because IGA includes variable order and variable continuity within a patch numerical procedures are required to determine efficient quadrature rules. The key is to take precise account of the continuity at element boundaries and construct rules that span more than one element. The conclusion by Hughes et al. is the so called "*half-thumb-rule*" which state that the number of quadrature points should roughly equal half the number of degrees of freedom. This rule differs from other rules that circulate, such as the "*equal to degree*" or "*degree plus one*", in that it is independent of the polynomial degree. However, as stated under limitations efficient quadrature has been omitted from this project. It is mentioned here because insufficient quadrature led to oscillations of the results when implementing the theories covered in the coming chapters. Hence, the reader who wishes to follow this thesis for their own implementations should be aware of this. A simple solution is to increase the number of Gauss points when needed.

Proof of convergence

It is important to be able to prove that NURBS based IGA results in convergent methods. Sufficient conditions for a basic convergence proof for a wide class of problems are satisfied by a basis that is:

- C^1 on element interiors,
- C^0 on element boundaries,
- and complete.

The ease with which continuity over element boundaries can be controlled was discussed previously. Furthermore, it can be shown that the isoparametric concept and the partition of unity property are enough to ensure the third condition, a complete basis, both of which are satisfied in IGA by the NURBS basis as discussed earlier. The proof of several other theorems regarding convergence of NURBS based IGA are discussed by Hughes, Cottrell and Bazilevs [2].

2.2.3 Shortcomings of NURBS based IGA

Isogeometric analysis is not the only effort towards unifying the processes of design and analysis. Other efforts to integrate CAD and analysis include for example the finite cell, structured XFEM and immersed boundary methods. One technique that has had some success for shell applications is the use of subdivision surfaces [8]. However, NURBS based analysis remains highly promising in the area of integrated design and analysis.

Among the downsides of using NURBS basis is that there is not watertight compatibility between multiple patches and there are difficulties in dealing with trimmed surfaces. The implication is that patches do not fit together tightly geometrically. One attempt at solving this is the use of immersed T-splines for basis. The inclusion of T-splines in CAD applications is quickly advancing. For shells, the local refinement capability of T-splines together with the discretization of Kirchhoff-Love shells shows promise as a highly efficient and accurate formulation. T-splines can reparameterize trimmed surfaces to form analysis suitable untrimmed T-spline representations [13]. Whether designers prefer immersed T-splines over NURBS is not of high importance though because both problems will be solved in IGA.

There is also a lot of work to be done in the area of solids. Hierarchical B-splines, immersed T-splines and some boundary element methods are examples of what is being explored [3].

Efficient implementation of IGA is another area that needs to be further developed. It is one of the topics for this years international conference on isogeometric analysis (2017) [17]. Other topics that will be addressed during the conference include industrial applications of IGA, local refinement and adaptivity for IGA and isogeometric spaces over unstructured meshes.

3 Practical aspects of implementation

3.1 Code structure

Given the similarities between classic FEA and IGA already discussed, it should not come as a surprise that the code structure for both analyses methods are very similar. As can be seen in Fig. 3.1, the main code structure for IGA is created by adding a couple of steps to the classic finite element analysis structure. In the figure these steps are visualized by grey boxes.

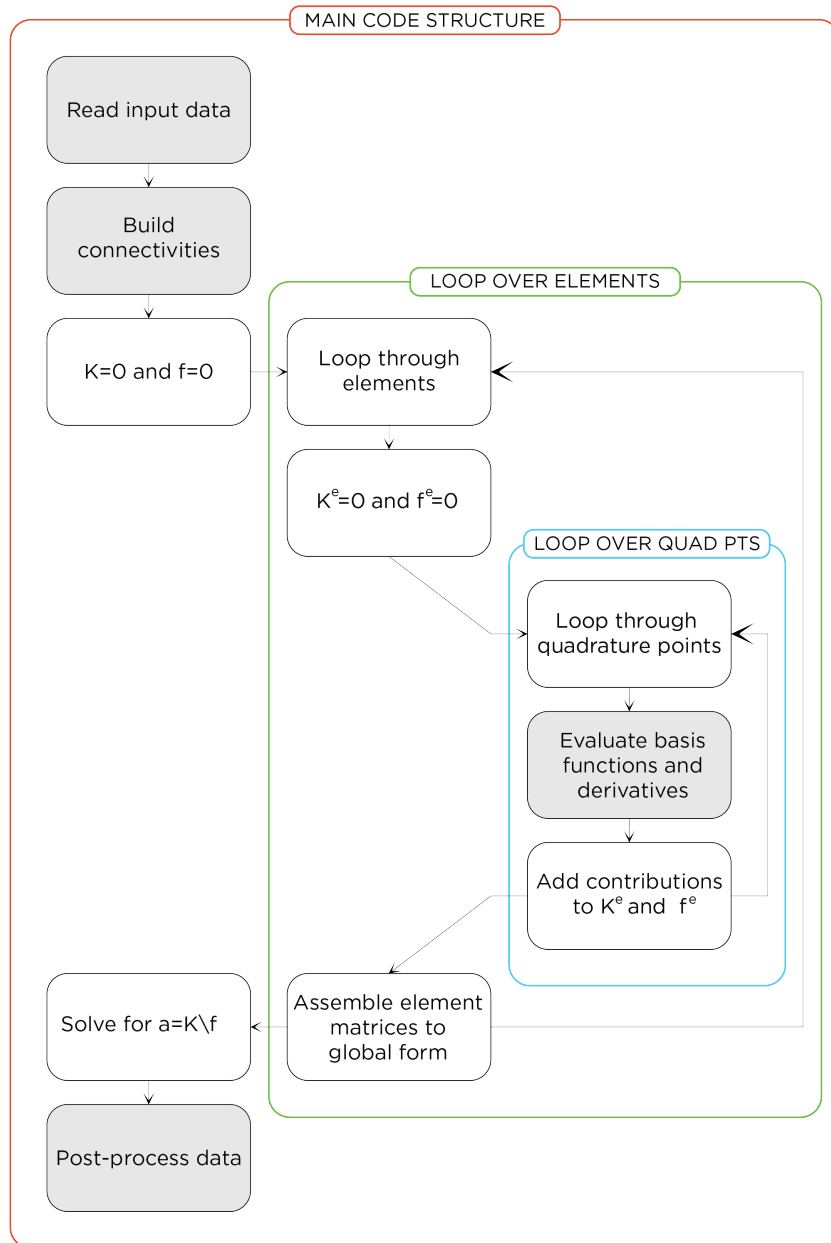


Figure 3.1: Code structure for IGA.

The added steps consist of a different way of reading input data, i.e. data in the NURBS format, slight tweaks to the connectivity arrays used for analysis, a new routine for evaluation of basis functions and a final step where output data is post-processed to obtain results on the geometry evaluated.

For the first step, reading input data, it is important to decide on a structure for NURBS data that is consistent throughout the code. Typically, it is convenient to store control points and weights in one matrix while the degree is stored as a separate variable and knot vectors as separate column vectors. However, the format is free as long as it is consistent. To quickly be able to test different geometries it was decided for this project to export NURBS data from Rhinoceros[®] through a self-written Grasshopper[®] component. This is an effective way for easily importing arbitrary geometries to Matlab[®] as the control point matrix and the knot vector quickly increase in size with complex geometries.

After refinement to desired mesh properties, connectivity matrices are constructed. The ones used in this thesis project are explained in Ch. 3.2. By looping over elements and quadrature points element stiffness matrices and load vectors are built. Pseudo code for shape function algorithms are stated by Piegl and Tiller [14]. This is the basis for most IGA shape function routines found as free downloads online. The routines by Nguyen et al. [8] in their open-source Matlab[®] package "igafem" work well for implementation. Note that there is an inconsistency in their routine as one approaches upper element boundaries which might lead to numerical errors. For a more dependable routine the Matlab[®] package by Alic [18] is suggested. It contains shape function routines for B-splines. By reviewing Eq. 2.4. in this report and the examples given in the download the NURBS basis can easily be obtained from these routines.

Finally, the system is assembled to global form and solved for sought solution field. The solution field is given for the control points. The results must be post-processed to obtain the sought values since the control points are not attached to the geometry itself in IGA. Theoretically, it works the same way for FE only then the nodal values are located on the approximated geometry. Thus, the step is often omitted from classic FEA. An example of post-processing is given in Ch. 3.5.

3.2 Connectivity matrices

There are several different connectivity matrices used in the implementation of IGA. The important thing though is not what they are called or how the data is structured but to understand the connectivity of NURBS. The ones mentioned here are partly used by Hughes [2] and partly used by Nguyen et al. [8].

NURBS coordinates array (INC)

INC is a matrix that relates a global function number to uni-variate knot indices. The column index is a global function number and the row index a parametric direction, e.g. ξ or η . Out comes the corresponding NURBS coordinate of the index space. For example, global function A and parametric direction ξ gives index space coordinate i in direction ξ by $INC(\xi, A) = i$.

Element functions array (IEN)

IEN relates a global basis function number to its local ordering. It is often referred to as the "element nodes array" in classic FEA. In IGA it takes the element number e and a local basis function number a and the output is the global basis function number A , e.g. $IEN(e, a) = A$.

The above mentioned are enough for the beam models implemented in this project. However, the information is restructured according to the needs in the shell model. Three different connectivity matrices are used, namely *element*, *index* and *sctr*. In the *element* matrix the row index is the element number and the output is the index of the control points affecting that element. By inputting the element number in *index* the index of ξ is obtained from column 1 and η from column, this is used to get element ranges in respective directions. For degree of freedom per element *sctr* is used, again with row index as element number.

3.3 Mesh refinement

There are two notions of a mesh in isogeometric analysis. There is the control mesh and then there is the mesh embedded in the geometry. The control mesh can be seen as a scaffold that enables manipulation of the geometry and the design. The embedded mesh consists of images of knot spans mapped onto the geometry. The number of knot spans in the knot vector will mirror the number of elements. Since a new knot can be inserted in an arbitrary location of the knot vector local refinement is possible. Local refinement is not covered in this thesis but it is easy to test by simply adding a knot in one of the knot spans of the knot vector and thereby locally turning one element into two. It is the dual mesh description that is key to implementation. It is true for classic FEA as well but due to the interpolatory basis the meshes coincide. The result is that one works on the control polygon in FEA and not the exact geometry [3].

The refinement methods described in Ch. 2.1.4. are used to refine the analysis mesh. These methods affect both the control mesh and the embedded mesh but not the geometry. Knot insertion corresponds to h -refinement in classic FEA. Additionally, p -refinement is similar in both FEA and IGA. However, k -refinement does not have an equivalent method in classic FEA. Throughout this project h -refinement is used for analyses in 1D and k -refinement for analyses in 2D.

3.4 Boundary conditions

Since the dual mesh and the degrees of freedom are located on the scaffold instead of on the geometry, applying boundary conditions might seem diffuse. However, the procedure is very similar to that of finite element methods with control points instead of nodes being restricted. The main thing to take note of is whether the formulations are in local or global coordinates and what these coordinate systems are. Change of coordinate systems is covered in detail in relation to the examples given in Ch. 4 and 5.

Something else that complicates how boundary conditions are dealt with is when the formulations are rotation-free. This is also covered in-depth in Ch. 4 for curves and in Ch. 5 for surfaces.

3.5 Post-processing

Obtaining the solution fields in terms of node coefficients is enough in classic FEA because the control mesh and the embedded mesh coincide. However, additional steps post the solution are required in IGA because the control points do not intersect with the geometry. The steps begin with choosing a set of coordinates on the geometry in physical space for which solution fields are sought. Then the corresponding values ξ in parameter space are obtained. With ξ known the knot span and connectivity at that location can be found. Thus, control point coefficients for active control points and basis functions at ξ can be computed. Finally, coefficients are multiplied with shape functions and the sought value is given by the sum. Change of coordinate system might have to be applied depending on the problem. Note that this is not novel in any sense, it can be done in classic FEA as well.

Example

Say one is interested in the displacement in tangential direction at the middle of a NURBS curve representing a loaded curved beam, see subsequent discussion in Ch. 4.6.1 Fig. 4.4. The knot span and connectivity of the corresponding parameter space value is $\xi = 0.5$ if it is a single element NURBS curve. If the curve has degree $p = 2$ it will have $p + 1 = 3$ control points affecting the geometry at this point. The tangential displacement coefficients for these control points can be extracted as a vector a_t^e where t stands for tangential displacements and e for element in the sense that the vector contains the coefficient that affects that element. Thereafter the shape functions at $\xi = 0.5$ are computed. The displacement on the physical curve at the midpoint is then given by $a_t^{mid} = \mathbf{N}(\xi = 0.5)\mathbf{a}_t^e$.

4 Curved Beams

Isogeometric analysis is especially advantageous when the subject for analysis is arbitrarily shaped with strong curvature. In classic FEA geometries are approximated with straight elements, leading to greater geometrical errors originating from the meshing process the larger the curvature is. As has been mentioned previously, this geometric approximation is avoided by using IGA. With the ultimate objective to analyze a shell in mind, implementation of IGA is initiated with routines for curved members in 1D. More specifically, curved beams are chosen as subject for analysis. While practicing implementation of IGA, there is also an interest in investigating whether the size of the geometrical error can be measured. Furthermore, the claim that IGA provides improved accuracy on a per-degree-of-freedom basis compared to classic FEA is studied. Results provide a good indication to whether the hypothesis presented in the introduction is correct before moving on to analysis in 2D and shells.

The beam analysis models implemented in this chapter are applicable to linear elastic problems. Analysis models based on both the Euler-Bernoulli and the Timoshenko beam theory are implemented. Firstly, the beam theories are introduced. The theories are based on curved beam elements instead of straight elements, which typically are used in classic FEA. By using curved elements the solutions are brought as close to the exact geometry as possible and potential geometrical errors can be measured more accurately. Secondly, shear and membrane locking is discussed before moving on to an example. The analytical solution to the beam problem chosen is easy to find in literature, it is for example described by Cazzani et al. [19]. Thus, it is easy to verify the model.

4.1 Notations

The models are valid for arbitrarily curved beams in one dimension, see schematic sketch in Fig. 4.1. The following restrictions and notations apply:

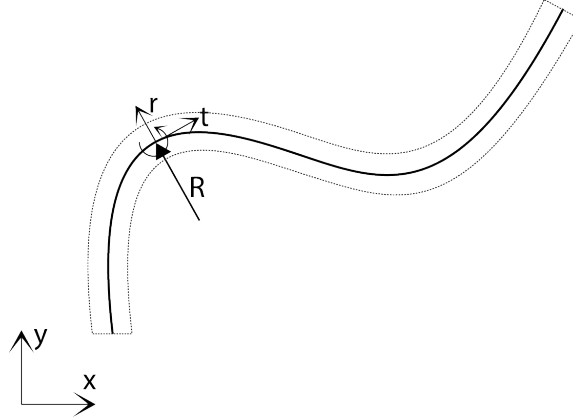


Figure 4.1: Local and global coordinate system definition for arbitrarily curved beams in one dimension.

- the centroid line is a planar curve,
- the curve is parametrized by arc length $s \in [0, l]$,
- one of the principle inertia axes and the shear centre of the cross-section lie in the same plane,
- the global reference system is denoted by $(0 ; x, y)$,
- the local reference system is denoted by $(0 ; t, r)$ where t stands for tangential and r for radial,
- and the radius of the curve is denoted by R .

4.2 Timoshenko beam theory

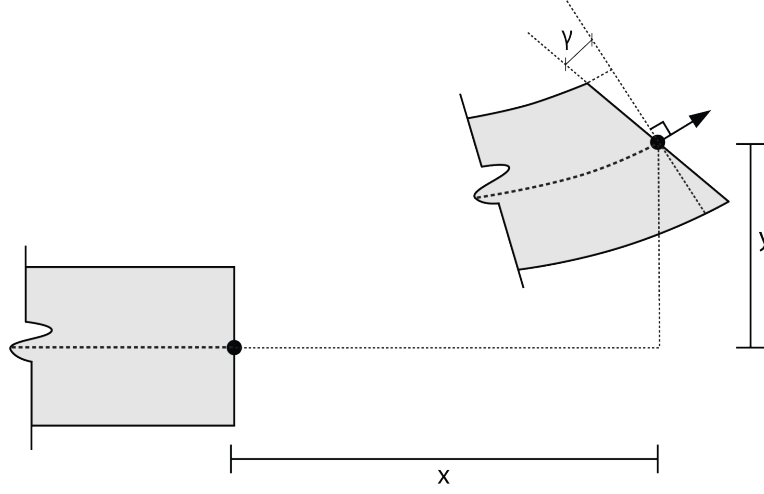


Figure 4.2: Illustration of assumptions of the Timoshenko beam theory. The undeformed state is displayed on the left and the deformed state on the right. Deformations are exaggerated.

The Timoshenko beam theory is based on the following assumptions:

- all deformations and strains are small,
- Hooke's law applies for relating stresses and strains,
- and planar cross sections remain planar to the deformed axis.

The third property is shown in Fig. 4.2. The figure also shows that the Timoshenko beam theory, in contrast to the Euler-Bernoulli beam theory, allows rotation between the cross section and the deformed axis. To include shear deformations leads to a less stiff beam model compared to a model based on Euler-Bernoulli theory. The equilibrium, kinematic compatibility and constitutive relations for a curved beam element based on Timoshenko beam theory are stated in this section. The equations used to derive the finite element formulation are described by Lim and Kitipornchai [20].

Firstly, the equilibrium equations are stated. Here the normal force is denoted N , shear force T and bending moment M . The variable s refers to a coordinate on the beam centroid line. Additionally, distributed forces in tangential and radial direction are denoted q_t and q_r , whereas distributed moment is denoted m .

$$\frac{dN}{ds} + \frac{T}{R} + q_t = 0 \quad (4.1)$$

$$\frac{dT}{ds} - \frac{N}{R} + q_r = 0 \quad (4.2)$$

$$\frac{dM}{ds} - T + m = 0 \quad (4.3)$$

Secondly, the kinematic compatibility equations are expressed. Displacements in tangential and radial direction are denoted u and w respectively, whereas rotations are denoted φ . The generalized axial ε , shear γ and curvature bending χ strains are then given by:

$$\varepsilon = \frac{du}{ds} + \frac{w}{R}, \quad (4.4)$$

$$\gamma = \frac{dw}{ds} - \frac{u}{R} + \varphi, \quad (4.5)$$

$$\chi = \frac{d\varphi}{ds}. \quad (4.6)$$

Finally, the constitutive relations are expressed. Young's modulus is denoted E , the cross-section area A , area moment of inertia I , the shear modulus G and the shear reduction factor κ .

$$N = EA\varepsilon \quad (4.7)$$

$$T = GA\kappa\gamma \quad (4.8)$$

$$M = EI\chi \quad (4.9)$$

The principle of minimum potential energy is used to derive the FE-formulation. Based on Eqs. (4.1)-(4.9) the expression is defined as:

$$u, w, \varphi = \arg \min_{\hat{u}, \hat{w}, \hat{\varphi}} \left\{ \frac{1}{2} \int_0^L (EA\hat{\varepsilon}^2 + EI\hat{\chi}^2 + GA\kappa\hat{\gamma}^2) ds - \int_0^L (q_t\hat{u} + q_r\hat{w} + m\hat{\varphi}) ds \mid \hat{u}(0) = \hat{w}(0) = \hat{\varphi}(0) = 0 \right\} \quad (4.10)$$

It gives the strain displacement matrix \mathbf{B} , stiffness matrix \mathbf{K} and load vector \mathbf{f} . The variable N_i denotes the shape functions whereas R and s are the radius and coordinate on the centroid line respectively. Additionally, the Jacobian J that maps between parameter and physical space is introduced. In accordance with previous denotations, the variable ξ is a coordinate in the parameter space. The variable \mathbf{q} denotes distributed load.

$$\mathbf{B} = \begin{bmatrix} \frac{dN_1}{ds} & \frac{N_1}{R} & 0 & \dots & \frac{dN_n}{ds} & \frac{N_n}{R} & 0 \\ -\frac{N_1}{R} & \frac{dN_1}{ds} & N_1 & \dots & -\frac{N_n}{R} & \frac{dN_n}{ds} & N_n \\ 0 & 0 & \frac{dN_1}{ds} & \dots & 0 & 0 & \frac{dN_n}{ds} \end{bmatrix} \quad (4.11)$$

$$\mathbf{K} = \int_{\xi} \mathbf{B}^T \begin{bmatrix} EA & 0 & 0 \\ 0 & GA\kappa & 0 \\ 0 & 0 & EI \end{bmatrix} \mathbf{B} J d\xi \quad (4.12)$$

$$\mathbf{f} = \int_{\xi} \mathbf{N}^T \mathbf{q} J d\xi \quad (4.13)$$

4.3 Euler-Bernoulli beam theory

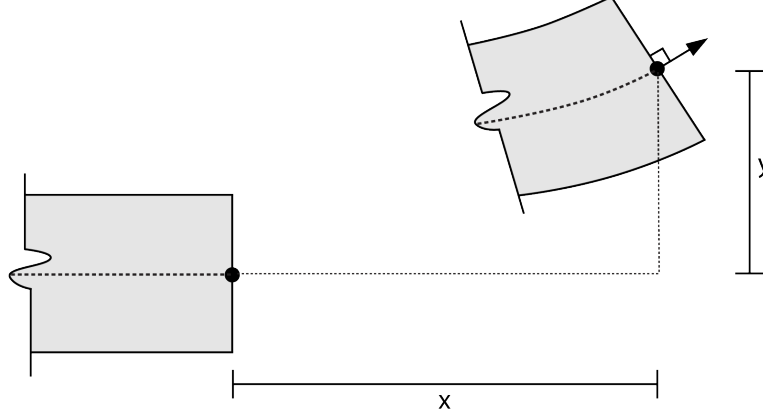


Figure 4.3: Illustration of assumptions of the Euler-Bernoulli beam theory. The undeformed state is displayed on the left and the deformed state on the right.

The classic beam theory, Euler-Bernoulli theory, is based on the same assumptions as Timoshenko beam theory. However, shear strain is neglected and it is assumed that the cross-sections remain normal to the deformed axis during deformation. This affects the kinematic compatibility equations, there is no longer a shear strain part. Additionally, the generalized shear stress T is no longer a part of the constitutive relations. It can be computed from the equilibrium equations if sought.

Thus, the same equilibrium equations that were defined for the Timoshenko beam applies:

$$\frac{dN}{ds} + \frac{T}{R} + q_t = 0 , \quad (4.14)$$

$$\frac{dT}{ds} - \frac{N}{R} + q_r = 0 , \quad (4.15)$$

$$\frac{dM}{ds} - T + m = 0 . \quad (4.16)$$

The kinematic compatibility equations that follow are expressed without the rotation degree of freedom φ . There is no generalized shear strain γ , only axial ε and curvature bending χ strains::

$$\varepsilon = \frac{du}{ds} + \frac{w}{R} \quad (4.17)$$

$$\chi = -\frac{d^2w}{ds^2} + \frac{1}{R} \frac{du}{ds} \quad (4.18)$$

Note that the curvature bending strain χ includes the second derivative of the radial displacement w with regards to the centroid line coordinate s . This leads to a finite element formulation that require the second derivative of the shape functions with regards to s , as will be shown further down.

The constitutive relations excluding the generalized shear stress are:

$$N = EA\varepsilon \quad (4.19)$$

$$M = EI\chi \quad (4.20)$$

Again, the principle of minimum potential energy is used to derive the FE-formulation. Based on Eqs. (4.14)-(4.20) the expression is defined by:

$$u, w = \arg \min_{\hat{u}, \hat{w}} \left\{ \frac{1}{2} \int_0^L (EA\hat{\varepsilon}^2 + EI\hat{\chi}^2) ds - \int_0^L (q_t \hat{u} + q_r \hat{w} + m \frac{\partial \hat{w}}{\partial s} ds \mid \hat{u}(0) = \hat{w}(0) = \frac{\partial \hat{w}}{\partial s}(0) = 0 \right\} \quad (4.21)$$

Just as for the Timoshenko beam, the strain displacement matrix \mathbf{B} , stiffness matrix \mathbf{K} and load vector \mathbf{f} are obtained. The same variable denotations apply. Note the new component $d^2 N_i / ds^2$ that is required for implementation of the strain displacement matrix. It has been shown previously that the second derivative with regards to parameter coordinate ξ easily is obtained in isogeometric analysis from the recursive formulation of shape function routines, but the second derivative with regards to physical space coordinate s is more complicated. For a full derivation and further explanation turn to Appendix A.

$$\mathbf{B} = \begin{bmatrix} \frac{dN_1}{ds} & \frac{N_1}{R} & \cdots & \frac{dN_n}{ds} & \frac{N_n}{R} \\ \frac{1}{R} \frac{dN_1}{ds} & -\frac{d^2 N_1}{ds^2} & \cdots & \frac{1}{R} \frac{dN_n}{ds} & -\frac{d^2 N_n}{ds^2} \end{bmatrix} \quad (4.22)$$

$$\mathbf{K} = \int_{\xi} \mathbf{B}^T \begin{bmatrix} EA & 0 \\ 0 & EI \end{bmatrix} \mathbf{B} J d\xi \quad (4.23)$$

$$\mathbf{f} = \int_{\xi} \mathbf{N}^T \mathbf{q} J d\xi \quad (4.24)$$

4.4 Shear and membrane locking

Geometrical errors are not the only source of inaccuracy in finite element solutions. Another important aspect to investigate is locking phenomena. As described by Bouclier et al. [21] locking occurs in thin beams due to inability of beam finite elements to reflect "shearless bending" and "inextensible bending". The result is an inaccurate answer that reflects much too small deformations and strains. It is the increasing slenderness that give rise to increasing shear-to-bending and membrane-to-bending stiffness that, in turn, introduces errors. Spurious shear and membrane contributions absorb a greater part of the strain energy which results in the tendency of the solution to lock to the transverse shear and/or membrane. Both membrane and shear locking are present in the Timoshenko beam model, while only membrane locking is an issue in the Euler-Bernoulli beam model.

Locking phenomena in isogeometric analysis can often be reduced by using a higher degree to describe the NURBS geometry, so called k -refinement. This approach is used in the shell model that is implemented in the next chapter. However, since a close comparison between IGA and classic FEA beam models is the basis of this chapter the implementation is restricted to h -refinement. Since low degrees in isogeometric analysis leads to locking problems in the same way that classic FEA experience locking, methods for relieving low order solutions from locking are implemented. Two different methods to alleviate locking are used, one for the classic finite element analysis and another for the isogeometric analysis. They both consist of projecting shear and bending strains to a lower-order basis, but in the method used with IGA continuity over element boundaries is maintained.

Locking in classic finite element analysis

To deal with locking in classic FEA a stable discretization of a mixed problem is used. The concept of static condensation is used to solve the system. Practically, this means that the number of degrees of freedom are reduced on element level. Note that if this method is to be applied to isogeometric analysis the reduction must be done on global level since one control point affect multiple elements. Each element has three nodes before the mid node is eliminated. The shape functions living on this node are assigned the properties degree $p = 1$ and continuity C^0 . Remaining shape functions are given properties degree $p = 2$ and continuity C^0 . See Appendix B for full description.

Locking in isogeometric analysis

The method chosen to alleviate the IGA beam model from locking is called the $\bar{\mathbf{B}}$ strain projection-type method. As described by Bouclier, Elguedj and Combescure [21] advantage is taken of the distinctive properties of the NURBS basis [21]. More specifically, the approach utilizes the ability to easily control continuity over element boundaries. The basic idea is to produce maximum continuity on a patch and then projecting the shear and bending strain onto a basis of lower dimension. The method is based on a strain projection type method originally used in classic FEA to remove volumetric locking and transverse shear locking in plates. It is equivalent with mixed methods where the associated mechanical variables are searched in the lower dimension space. The discrete form of the mixed problem in the $\bar{\mathbf{B}}$ strain projection-type method is given by:

$$\begin{bmatrix} \mathbf{K}_f & \bar{\mathbf{B}}_m^T & \bar{\mathbf{B}}_s^T \\ \bar{\mathbf{B}}_m & \mathbf{M}_m & \mathbf{0} \\ \bar{\mathbf{B}}_s & \mathbf{0} & \mathbf{M}_s \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \alpha_N \\ \alpha_T \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad \xLeftrightarrow{\text{condensation}} \quad \mathbf{K}_{\bar{\mathbf{B}}} \mathbf{a} = \mathbf{f} \quad (4.25)$$

where \mathbf{f} is the load vector previously defined and the displacement vector \mathbf{a} is obtained by solving the system $\mathbf{a} = \mathbf{K}_{\bar{\mathbf{B}}}^{-1} \mathbf{f}$.¹ For the Euler-Bernoulli beam model the third block column and the third block row in the stiffness matrix in Eq. (4.25) are omitted. The matrices \mathbf{M}_m and \mathbf{M}_s are computed with the basis of lower order, here denoted by $\bar{\mathbf{N}}$. This basis is produced from NURBS data with order $p - 1$ and continuity C^{p-1-m} .

¹Note that $\mathbf{K}_{\bar{\mathbf{B}}}^{-1}$ formally denotes the inverse. However, in practice a linear system of equations is solved and not the inverse.

$$\mathbf{M}_m = \mathbf{M}_s = \int_0^L \bar{\mathbf{N}}^T \bar{\mathbf{N}} ds \quad (4.26)$$

Just as before s denotes a coordinate on the centroid beam line in physical space and R the radius. The matrix $\bar{\mathbf{B}}_m$ is obtained from:

$$[\bar{\mathbf{B}}_m] = [[\bar{\mathbf{B}}_m^1] [\bar{\mathbf{B}}_m^2] [\mathbf{0}]] , \quad (4.27)$$

$$\bar{\mathbf{B}}_{mij}^1 = \int_0^L \bar{N}_i^T \frac{dN_j}{ds} ds , \quad \bar{\mathbf{B}}_{mij}^2 = \int_0^L -\bar{N}_i \frac{N_j}{R} ds , \quad (4.28)$$

and the matrix $\bar{\mathbf{B}}_s$ is produced by:

$$[\bar{\mathbf{B}}_s] = [[\bar{\mathbf{B}}_s^1] [\bar{\mathbf{B}}_s^2] [\bar{\mathbf{B}}_s^3]] . \quad (4.29)$$

$$\bar{\mathbf{B}}_{sij}^1 = \int_0^L -\bar{N}_i^T \frac{N_j}{R} ds \quad \bar{\mathbf{B}}_{sij}^2 = \int_0^L \bar{N}_i^T \frac{dN_j}{ds} ds \quad \bar{\mathbf{B}}_{sij}^3 = \int_0^L -\bar{N}_i^T N_j ds \quad (4.30)$$

The matrix $\mathbf{K}_{\bar{B}}$ is given by the sum $\mathbf{K}_{\bar{B}} = \bar{\mathbf{K}}_m + \bar{\mathbf{K}}_s + \bar{\mathbf{K}}_f$ where the bending related parts of Eqs. (4.11) and (4.22) for the Timoshenko and Euler-Bernoulli theory respectively gives $\bar{\mathbf{K}}_f$. The other two stiffness matrices $\bar{\mathbf{K}}_m$ and $\bar{\mathbf{K}}_s$ are given by:

$$\bar{\mathbf{K}}_m = (EA) \bar{\mathbf{B}}_m^T \mathbf{M}_m^{-1} \bar{\mathbf{B}}_m \quad (4.17)$$

$$\bar{\mathbf{K}}_s = (G\kappa A) \bar{\mathbf{B}}_s^T \mathbf{M}_s^{-1} \bar{\mathbf{B}}_s \quad (4.18)$$

4.5 Section forces

To complete the beam model the section forces are computed. This section covers how it is done in isogeometric analysis. An easy and quick way to extract them in classic FEA is to use the Matlab[®] package CALFEM[®]. The method for extracting the forces is a result of the chosen locking handling method. The mechanical variable $\alpha_{\mathbf{N}}$ is used to obtain the normal force for the Timoshenko and Euler-Bernoulli beam models. The variable $\alpha_{\mathbf{T}}$ is used to obtain the shear force for the Timoshenko beam model. These are given by:

$$\alpha_{\mathbf{N}} = (EA) \mathbf{M}_m^{-1} \bar{\mathbf{B}}_m^T \mathbf{a} , \quad (4.19)$$

$$\alpha_{\mathbf{T}} = (G\kappa A) \mathbf{M}_s^{-1} \bar{\mathbf{B}}_s^T \mathbf{a} . \quad (4.20)$$

The mechanical variables are used in the same way that control point coefficients are used to obtain displacements of points in physical space. The shape functions of a given evaluation point in parameter space $\mathbf{N}(\xi)$ are multiplied with active control point coefficients $\alpha_{\mathbf{N}}$ or $\alpha_{\mathbf{T}}$ to get the normal and/or shear force at a physical coordinate $s(\xi)$. The moment is computed from the kinematic variable \mathbf{a} and constitutive relation $M = EI\chi$.

4.6 Example

4.6.1 Problem setup

The problem chosen for analysis is a circular beam. More specifically, it consists of a quarter-circle that is fixed at one end and loaded with a point load transversely to the beam at the other end. The problem and the initial unrefined NURBS geometry is shown in Fig. 4.4.

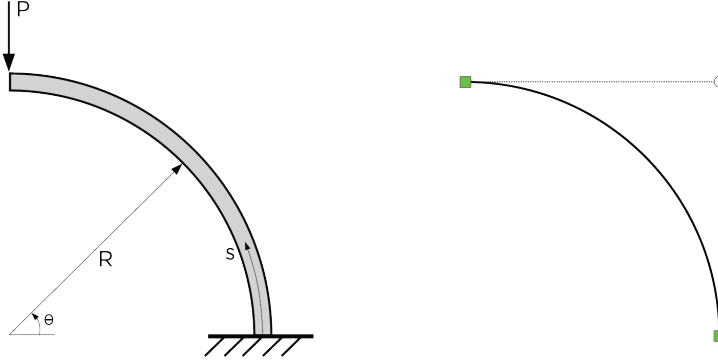


Figure 4.4: Quarter-circle beam fixed at one end and loaded with a point load at the other end on the left. Unrefined NURBS geometry on the right.

The most basic form of this NURBS geometry is given by the knot vector $\Xi = \{0, 0, 0, 1, 1, 1\}$, the degree $p = 2$ and three control points with corresponding weights. These are the x -coordinates $\mathbf{x} = [1 \ 1 \ 0]$, the y -coordinates $\mathbf{y} = [0 \ 1 \ 1]$ and the weights $\mathbf{w} = [1 \ 1/\sqrt{2} \ 1]$. The problem is solved for displacements and section forces. Both beam models are solved using IGA with curved beam elements and classic FEA with straight beam elements. The Timoshenko beam model is also solved with classic FEA using curved beam elements. This enables a comparison of two highly similar set-ups where the first is solved with IGA and the second with classic FEA, without any other differences except for the locking handling method. It is important because a comparison between classic FEA and IGA is the objective of this chapter. Setting up the Euler-Bernoulli beam model is mainly a step towards handling implementation of rotation-free formulations in 2D, which is the next step towards next chapter's Kirchhoff-Love shell formulations. Therefore, it is sufficient for our purposes to implement curved beam elements in classic FEA only for the Timoshenko beam model.

The analytical solution to the problem is given by Cazzani et al. [20]. It is given in terms of an angle θ instead of the centroid line coordinate s as shown in Fig. 4.4. The force P [N] is the point load applied at the end and R is the radius. The variables E, A, I, G, κ are the material parameters that were previously defined in the beam theory sections. The tangential displacement u is given by:

$$u = -P(c_1(\sin(\theta) + \theta \cos(\theta)) + c_2 \sin(\theta)) , \quad (4.21)$$

$$c_1 = \frac{1}{2} \left(\frac{R}{EA} + \frac{R}{GA\kappa} + \frac{R^3}{EI} \right) , \quad (4.22)$$

$$c_2 = \left(\frac{R}{GA\kappa} + \frac{R^3}{EI} \right) . \quad (4.23)$$

Additionally, the analytical solution for the generalized stresses N , T and M are obtained from equilibrium:

$$N = -P \cos(\theta) , \quad (4.24)$$

$$T = P \sin(\theta) , \quad (4.25)$$

$$M = -PR \cos(\theta) . \quad (4.26)$$

All of these entities are used to verify the beam models. Different heights h of the beam are tested but in general the material parameters used are:

Variable	Value	Unit
Young's modulus, E	80e9	[Pa]
Height of beam, h	0.001	[m]
Width of beam, b	0.2	[m]
Poisson's ratio, ν	0.2	[-]
Radius, R	1	[m]
Point load, P	-1	[N]

The cross-section is rectangular and the area A is given by $A = bh$ while the area moment of inertia I is given by $I = bh^3/12$. The shear modulus G follows analytically for isotropic materials:

$$G = \frac{E}{2(1 + \nu)} . \quad (4.27)$$

The shear reduction factor κ is given by [22]:

$$\kappa = \frac{10(1 + \nu)}{12 + 11\nu} . \quad (4.28)$$

Now the only thing left to define are the boundary conditions. For the Timoshenko beam model it is self-explanatory. However, the Euler-Bernoulli beam model is a rotation-free beam formulation and keeping the beam from rotating is a bit more complicated. A degree of freedom on the control point adjacent to the control point at the clamped end must be fixed for the isogeometric analysis. Since the NURBS curve lies in the xy -plane it is the degree of freedom in the x -direction that must be locked into place to prevent rotation.

4.6.2 Analysis model

Besides using different methods for treating locking the IGA and classic FEA analyses are close to identical. The additional steps required in the isogeometric analysis were explained in Ch. 3. With these added the following code structure of the IGA model for 1D beam analysis is obtained:

Algorithm 1: General structure for isogeometric analysis of beams

Input : NURBS data in chosen format

```

1 Load initial NURBS data.
2 Perform  $h$ -refinement.
3 Generate connectivity matrices.
4 for  $i = 1$  to number of elements  $e$  do
5     Get the knot span of  $e$ .
6     Get the connectivity of  $e$ , i.e. indices of active control points.
7     Get coordinates of control points influencing  $e$ .
8     for  $j = 1$  to number of Gauss points  $gp$  do
9         Compute  $\mathbf{K}_f$ ,  $\bar{\mathbf{B}}_s$ ,  $\bar{\mathbf{B}}_m$ ,  $\mathbf{M}_s$  and  $\mathbf{M}_m$ .
10    end
11    Assemble element contributions to global form.
12 end
13 Compute  $\mathbf{K}_{\bar{B}}$ , note that this is done with the global matrices.
14 Apply boundary conditions.
15 Apply point load.
16 Solve the system for  $\mathbf{a} = \mathbf{K}_{\bar{B}}^{-1}\mathbf{F}$ .
17 for  $k = 1$  to number of evaluation points  $\xi$  do
18     Compute the section forces as described in Ch. 4.5.
19     Compute displacement in the physical domain by:
        - applying change of coordinate system as described in Appendix C,
        - and calculating the displacement as described in Ch. 3.5.
20 end

```

Output : Displacements $u, w, (\varphi)$ and section forces

4.6.3 Results

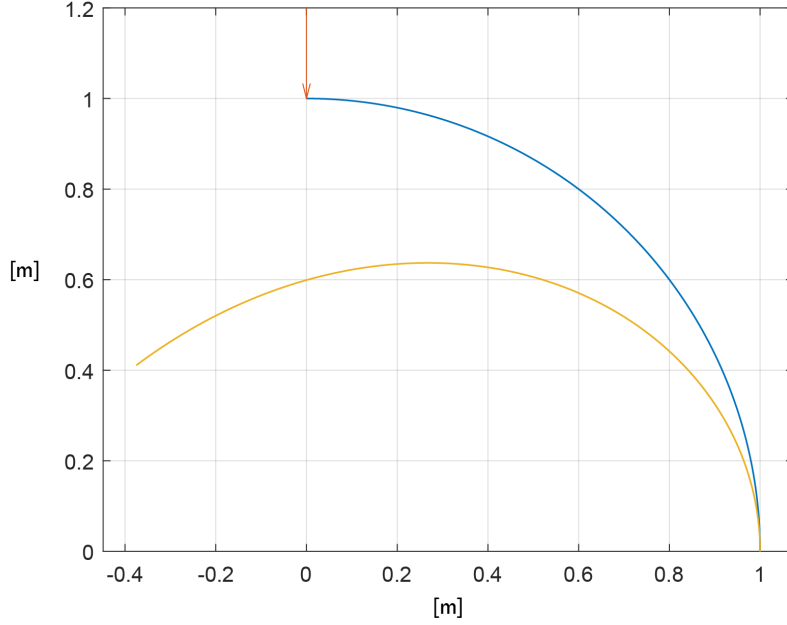


Figure 4.5: Undeformed (blue line) and deformed (yellow line) state. Maximum deflection at free end is 0.58 m, this aligns well with analytic solution for both beam models.

Verification of IGA beam model

The results show that deflections and section forces computed with the Timoshenko and the Euler-Bernoulli beam models align well with the analytical solution. Here, results for the IGA based models are shown. As can be seen in Fig. 4.5 the maximum deflection for given data is 0.58 m. The same value is given by Eq. (4.21). Thus, the displacement in the tangential direction u is verified.

The section forces are also compared to the analytical solution, given by Eqs. (4.24)-(4.26). See Fig. 4.6 on the next page for the results. The plots include moment, normal force and shear force diagrams for the Timoshenko beam model. The shear force diagram for the Euler-Bernoulli beam model is omitted for two reasons. Primarily because one can argue whether it actually is of interest to see the shear force plot for a theory that inherently neglects shear strain. Secondly because computing the shear force based on the equilibrium equations, see Eq. (4.16), is a time-consuming and tedious task. It requires the third derivative of the shape functions with regards to the physical space coordinate.

In the force plots on the next page, the light blue lines represents the analytical solution. The values given with isogeometric analysis are represented by the dark blue lines. The two lines almost completely overlap. Note that the forces are plotted over a straight line as if the curved beam was unrolled with the clamped end on the left and the free end on the right.

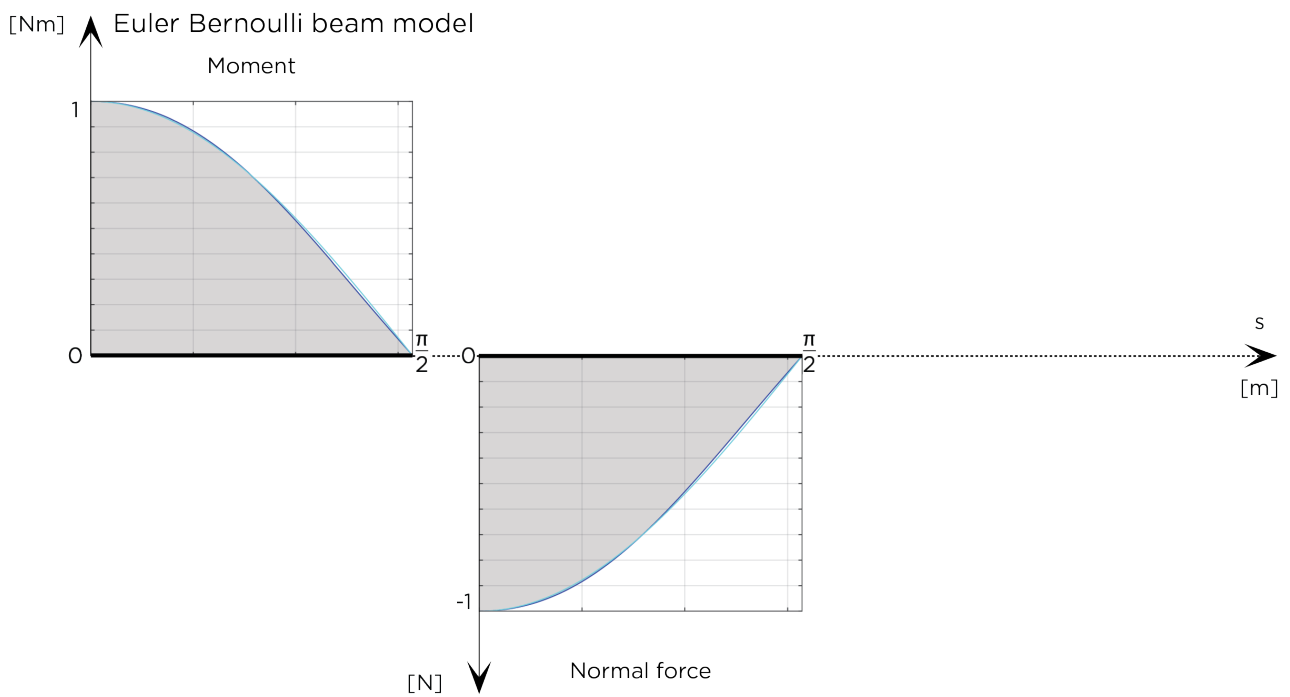
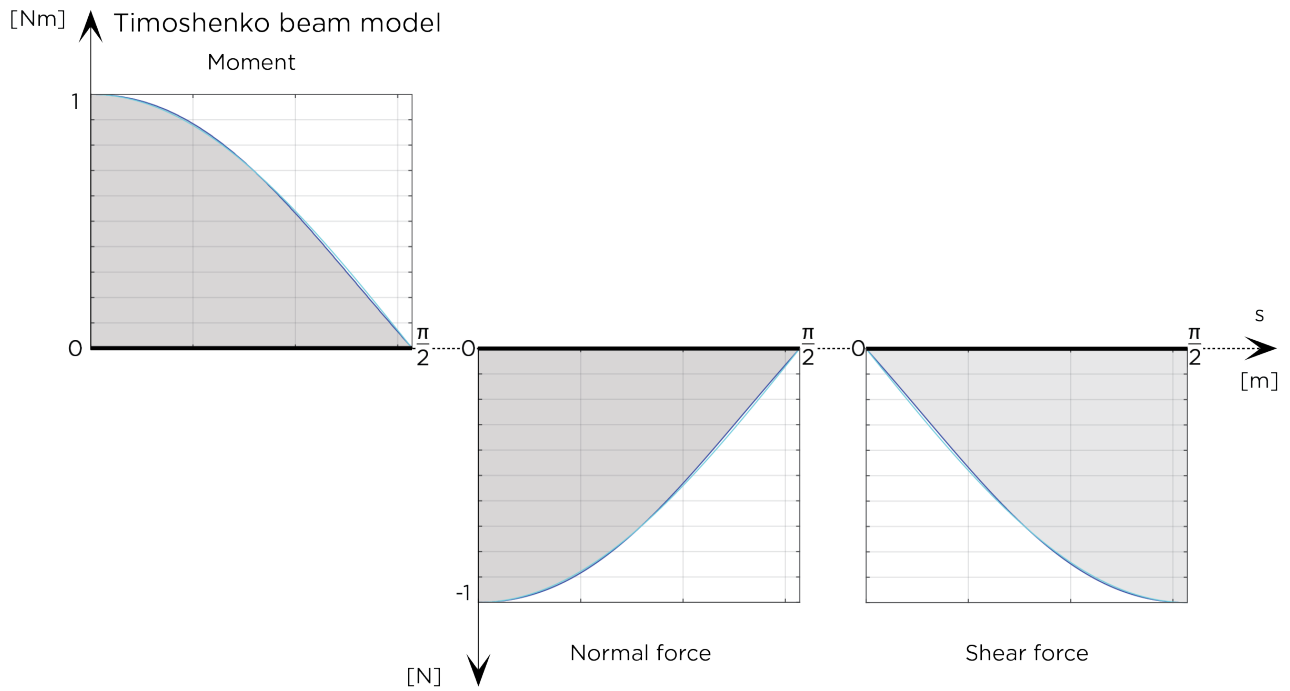


Figure 4.6: Moment, normal force and shear force diagram for respective beam model. Dark blue lines represent numerical solution and light blue analytical solution. The distribution over the length of the beam is shown with the clamped end on the left at "0" and the free end on the right at " L ". Here L denotes the length of the beam.

Comparison between classic FEA and IGA beam models

The results include the beam problem analyzed with classic FEM with straight and curved beam elements as well as IGA solved with curved beam elements. Three error plots are produced based on the results.

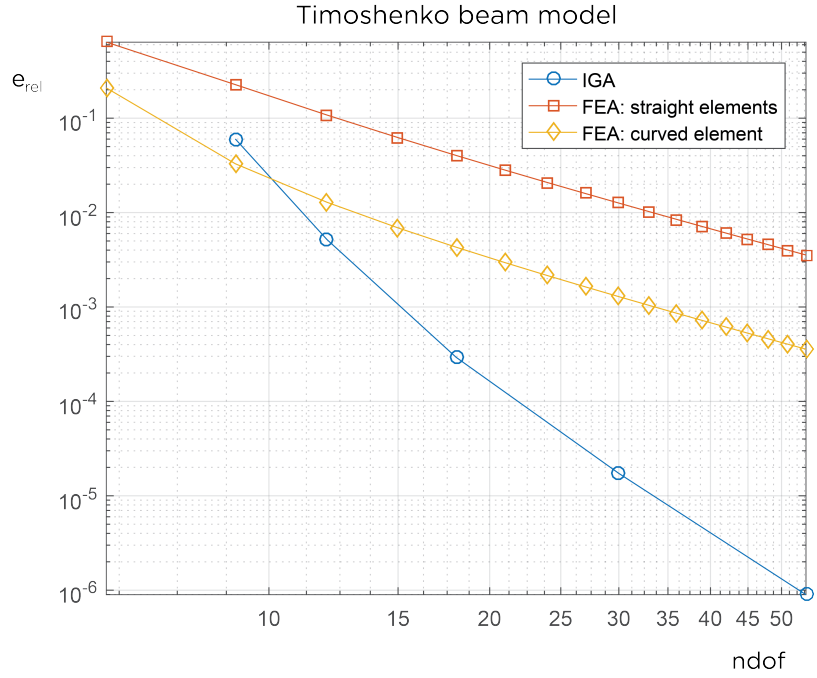


Figure 4.7: Relative error for the Timoshenko beam model.

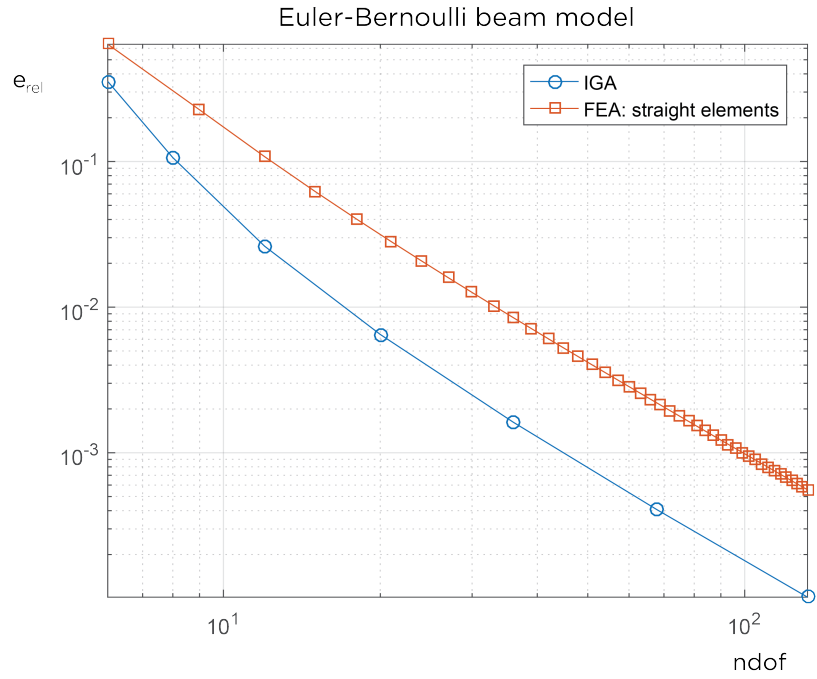


Figure 4.8: Relative error for the Euler-Bernoulli beam model.

The first two plots, Fig. 4.8 and 4.7, display the relative error in strain energy on a per-degree-of-freedom basis. The number of degrees of freedom is denoted $ndof$. The analyses are compared to the analytical solution. The classic FE analysis with straight beam elements is performed with CALFEM[®], an educational finite element Matlab[®] package created at Lund University [23]. In the following, we study the strain energy of the solution on the form:

$$E = \frac{1}{2} \|u\|^2, \quad (4.29)$$

$$\|\cdot\| = \sqrt{a(\cdot, \cdot)}, \quad (4.30)$$

where $a(\cdot, \cdot)$ denotes the "energy" product pertinent to the left-hand side of the weak form. For a discrete approximation, we then obtain:

$$E_h = \frac{1}{2} a(u_h, u_h) = \frac{1}{2} \mathbf{a}^T \mathbf{K} \mathbf{a}, \quad (4.31)$$

With h denoting the approximate solution and ref the analytical reference solution, the relative error can be expressed as:

$$e_{rel} = \left| \frac{E_{ref} - E_h}{E_{ref}} \right|. \quad (4.32)$$

The results obtained with the Timoshenko beam model in Fig. 4.7 clearly show that IGA performs better on both points that are investigated, i.e. more accurate results and higher convergence rate. The relative error is displayed on the vertical axis. The horizontal axis shows the number of degrees of freedom that gives said error. Classic FEA with curved beam elements gives the best approximation for few degrees of freedom, but somewhere around $ndof = 10$ the isogeometric solution outperforms classic FEA in this regard. IGA also shows a higher convergence rate than classic FEA with both straight and curved beam elements. Note the gap between the FEA curves. Classic FEA with curved beam elements give a more accurate solution for all degrees of freedom compared to classic FEA with straight beam elements. The additional error that comes from using straight elements reflect the geometrical error introduced by approximating a curved geometry with a mesh consisting of straight segments.

The results obtained with the Euler-Bernoulli beam model are seen in Fig. 4.8. Again it is clearly shown that IGA continuously gives a better approximation compared to classic FEA with straight beam elements. These results are what was expected and indicate possible fulfillment of the initial hypothesis.

The third error plot, see Fig. 4.9, is used to investigate accuracy of the Euler-Bernoulli and Timoshenko beam models with IGA as the height of the beam changes. A height variation gives the slenderness ratio on the horizontal axis. Given on the vertical axis is the difference in converged results for said slenderness of the two beam models. On the left end of the curve, where the beam is thin, the solutions approach each other. For thick beams, towards the right end of the curve, the error in the converged result of the Euler-Bernoulli beam increases. This is expected and verifies the model. The difference between the models is defined as:

$$u_{\Delta} = \frac{u_T - u_{EB}}{u_T}, \quad (4.33)$$

where T denotes the Timoshenko beam solution and EB the corresponding Euler-Bernoulli solution. The height of the beam is denoted h and varies from 10 mm to 90 mm. The radius R is set to 1 m.

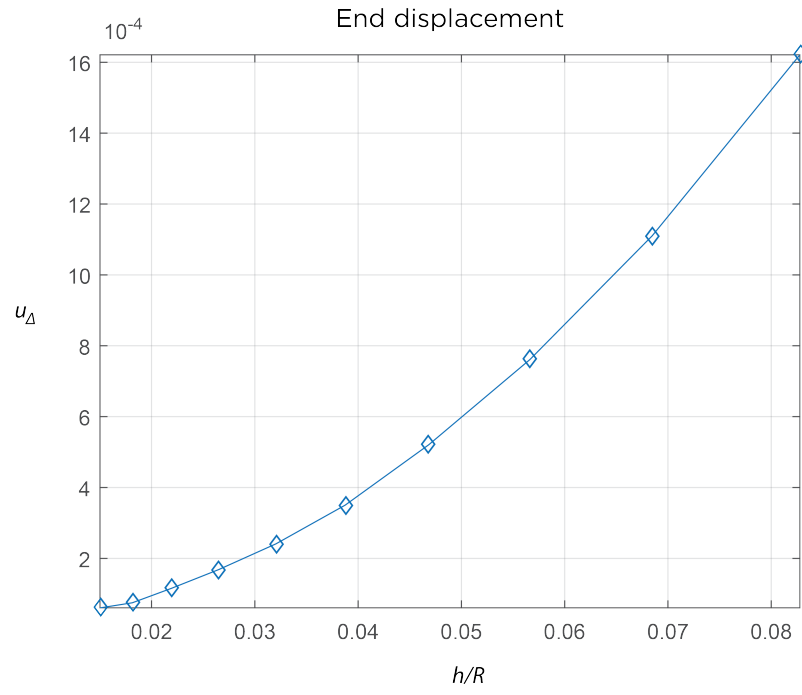


Figure 4.9: Accuracy of end displacement for the Euler-Bernoulli beam model u_{EB} and the Timoshenko beam model u_T for varying thickness of beam.

5 Thin shells

Shell structures are defined by their geometry. In general, they are curved surfaces with a thickness that is very small compared to the other dimensions. The variations in curvature and shape are limitless, making it a popular structural element choice for a wide variety of applications. In architecture shells are found historically in for example masonry vaults. The shell structures designed today are often more organic in their architectural expression. As described by Williams [24], the general strive in structural design of shells is for structures that carry load in pure compression or through a combination of compression and tension. Pure tension structures are often omitted as the term shell implies something relatively rigid. Pure tension structures include for example fabric structures and they are considered a separate category even though they fulfill the geometrical definition of a shell. Shells can also carry load in bending. In that case the thickness is used in order to give the shell bending stiffness. In this thesis the focus is on the study of thin shells. They are, naturally given their slenderness, little or not at all dependent on bending. This leads to a less redundant structure that is more susceptible to geometric imperfections. A heightened geometric accuracy requirement is one of the reasons why isogeometric analysis is especially suitable for analysis of thin shells.

5.1 Notations

Notations and variables are described as they appear but throughout the chapter the following applies:

- the usual convention for Latin and Greek indices, i.e. $i = 1, 2, 3$ and $\alpha = 1, 2$.
- the Einstein summation rules.
- a comma for partial differentiation, for example $f_{,i} = \frac{\partial f}{\partial x_i}$.
- subscripts for covariant components.
- superscripts for contravariant components.
- the subscript 0 for quantities in the reference configuration, for example a point on the reference middle surface is denoted φ_0 .

5.2 Kirchhoff-Love shell theory

5.2.1 Preliminaries

Classic Kirchhoff-Love shell theory is widely accepted as *the theory* for analysis of thin shells. However, the Kirchhoff formulations are not as widespread in classic finite element analysis as formulations based on Reissner-Mindlin theory. This has to do with continuity requirements over element boundaries. In IGA it is easy to control continuity through the knot vector and available refinement methods, a property that becomes highly useful now.

The shell model implemented in this chapter is valid for linear elastic analysis of arbitrarily shaped thin shells. Shearing and stretching of the normal to the mid-surface of the shell are omitted and the shell director is restricted to remain normal to the mid-surface in both the undeformed and the deformed state. This is analogous to the Bernoulli assumption about plane cross-sections remaining planar and perpendicular to the line of deflection.

The derivations of the discretized model equations follow Kiendl [11] and restrictions to small strain kinematics according to Millán et al. [25] are applied. First, the kinematics of the shell body is formulated. Secondly, the Kirchhoff-Love hypothesis and assumptions are introduced. Through the equilibrium configuration and elastic constitutive relations the weak form is obtained and the linearized system of equations for the shell model is given by Galerkin discretization. To be able to follow the derivations a basic understanding of differential geometry of surfaces is necessary. Bathe [10] provides a thorough description of the geometrical preliminaries. A short introduction to the shell geometry is also given below.

5.2.2 The shell geometry

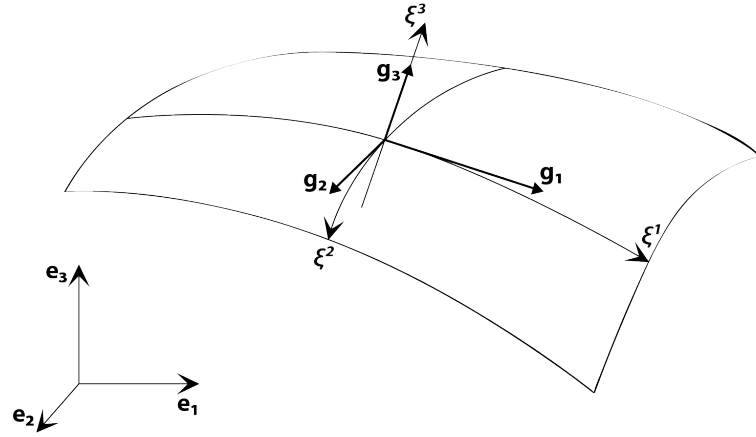


Figure 5.1: Surface in three-dimensional space with curvilinear coordinates, local basis and reference orthonormal basis.

Basics

A point in three-dimensional space is identified with its coordinates (x^1, x^2, x^3) and unit vectors of a reference orthonormal basis such as $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$:

$$\mathbf{x} = x^1 \mathbf{e}_1 + x^2 \mathbf{e}_2 + x^3 \mathbf{e}_3 , \quad (5.1)$$

where \mathbf{x} constitutes a position vector. In this example the basis \mathbf{e}_i denotes the global Cartesian basis, but for describing free-form surfaces it is helpful to use curvilinear coordinates and local bases. Fig. 5.1 shows an example of a surface in three-dimensional space with these coordinates and bases. Two important bases are the covariant basis \mathbf{g}_i and the contravariant basis \mathbf{g}^i . Corresponding covariant and contravariant coordinates are denoted ξ_i and ξ^i . The position vector expressed in terms of these coordinates $\mathbf{x} = \mathbf{x}(\xi^1, \xi^2, \xi^3)$ is related to the covariant basis by:

$$d\mathbf{x} = d\xi^i \mathbf{g}_i . \quad (5.2)$$

Thus, the covariant base vectors are defined as:

$$\mathbf{g}_i = \frac{\partial \mathbf{x}}{\partial \xi^i} = \mathbf{x}_{,i} . \quad (5.3)$$

The covariant basis is related to the contravariant base vectors by:

$$\mathbf{g}_i \cdot \mathbf{g}^j = \delta_i^j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} . \quad (5.4)$$

A surface is a two-dimensional geometry and thus defined parametrically by two curvilinear coordinates (ξ^1, ξ^2) . For the case of surfaces the first two covariant base vectors \mathbf{g}_α can thereby be computed as described in Eq. (5.3). The third base vector \mathbf{g}_3 is defined as a normalized vector orthogonal to \mathbf{g}_1 and \mathbf{g}_2 :

$$\mathbf{g}_3 = \frac{\mathbf{g}_1 \times \mathbf{g}_2}{\|\mathbf{g}_1 \times \mathbf{g}_2\|} . \quad (5.5)$$

From Eq. (5.3) it can be reasoned that the contravariant base vectors \mathbf{g}^α lie in the tangent plane spanned by the covariant base vectors \mathbf{g}_α . The third contravariant base vector must thereby be equal to the third covariant base vector $\mathbf{g}^3 = \mathbf{g}_3$.

The metric tensor \mathbf{g} , also called the identity tensor, is an important quantity for the description of surfaces. It can be expressed in both covariant and contravariant bases:

$$\mathbf{g} = g^{\alpha\beta} \mathbf{g}_\alpha \otimes \mathbf{g}_\beta = g_{\alpha\beta} \mathbf{g}^\alpha \otimes \mathbf{g}^\beta . \quad (5.6)$$

The metric coefficients $g_{\alpha\beta}$ are given by the scalar product of the covariant base vectors:

$$g_{\alpha\beta} = \mathbf{g}_\alpha \cdot \mathbf{g}_\beta . \quad (5.7)$$

This expression is also called *the first fundamental form of surfaces* and contains important properties of the surface such as length of and angles between base vectors. The contravariant metric coefficients are given by:

$$[g^{\alpha\beta}] = [g_{\alpha\beta}]^{-1} . \quad (5.8)$$

Given these metric coefficient matrices the contravariant base vectors can be computed from the covariant base vectors and the other way around:

$$\mathbf{g}^\alpha = g^{\alpha\beta} \mathbf{g}_\beta , \quad (5.9)$$

$$\mathbf{g}_\alpha = g_{\alpha\beta} \mathbf{g}^\beta . \quad (5.10)$$

The curvature properties of the surface are described by *the second fundamental form of surfaces* with the curvature tensor coefficients defined by:

$$b_{\alpha\beta} = \mathbf{g}_\alpha \cdot \mathbf{g}_{3,\beta} = -\mathbf{g}_{\alpha,\beta} \cdot \mathbf{g}_3 . \quad (5.11)$$

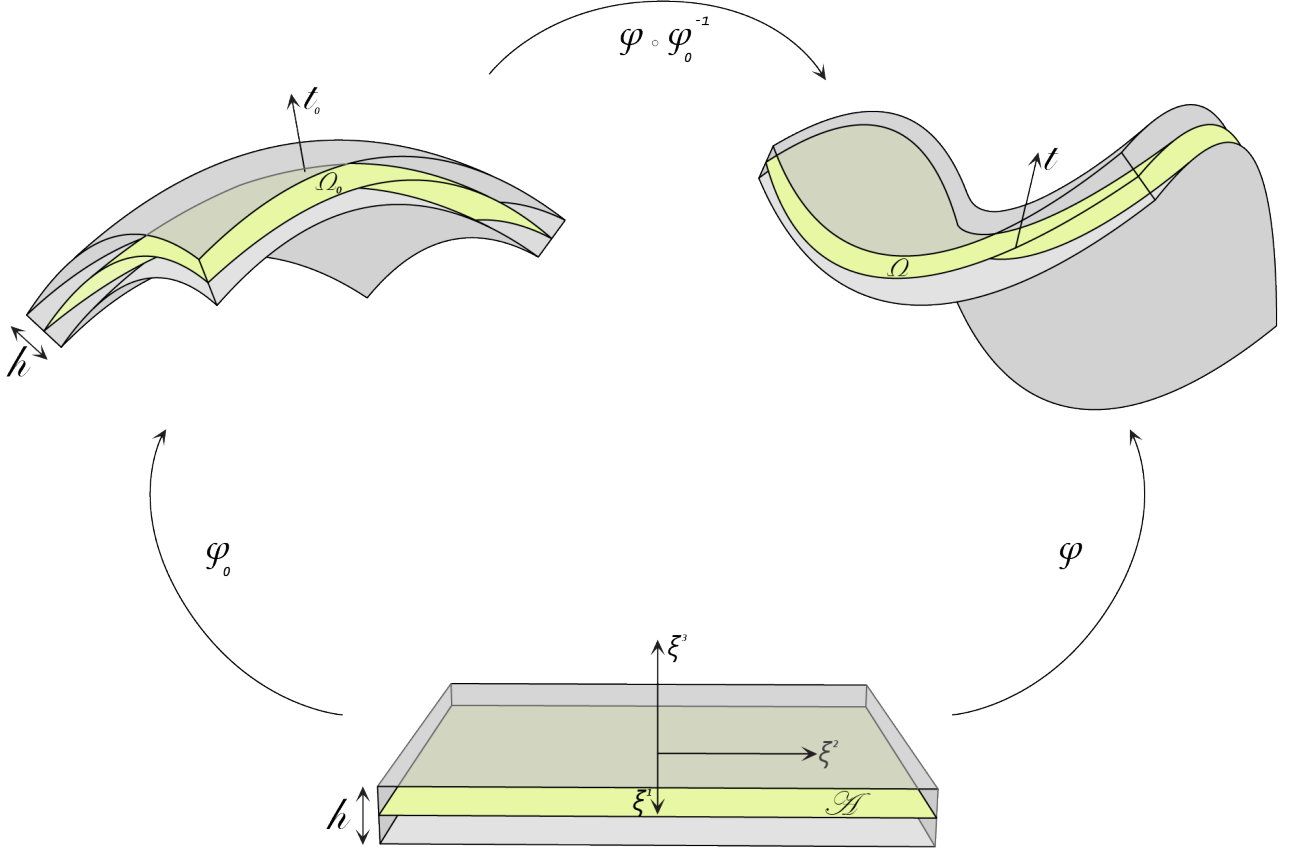


Figure 5.2: Mappings between undeformed Ω_0 state, deformed Ω state and parametric space \mathcal{A} .

The shell body

The mathematical model of a shell is analyzed as a midsurface with a parameter defining the thickness. Following the notations in Fig. 5.2 the midsurface is denoted Ω_0 in the undeformed state, also called the reference state. The thickness is denoted h . The mapping denoted φ_0 is the mapping between the parametric space of the midsurface, defined as $\mathcal{A} \subset \mathbb{R}^2$, and the undeformed configuration. Furthermore, the unit vector field normal to the midsurface is denoted \mathbf{t}_0 . An area element of the middle surface can be computed as $d\Omega_0 = \bar{j}_0 d\xi^1 d\xi^2$ where $\bar{j}_0 = \|\varphi_{0,1} \times \varphi_{0,2}\|$.

The shell body, here denoted ζ , lies in a three-dimensional reference domain such that $\zeta \subset \mathbb{R}^3$. If a configuration \mathbf{x}_0 is defined as the mapping from the parametric domain $\mathcal{A} \times [-h/2, h/2]$ into \mathbb{R}^3 it can be expressed by:

$$\mathbf{x}_0(\xi^1, \xi^2, \xi^3) = \varphi_0(\xi^1, \xi^2) + \xi^3 \mathbf{t}_0(\xi^1, \xi^2). \quad (5.12)$$

The body can then be geometrically defined as:

$$\zeta = \left\{ \mathbf{x}_0 \in \mathbb{R}^3 \mid \mathbf{x}_0(\xi^1, \xi^2, \xi^3) = \varphi_0(\xi^1, \xi^2) + \xi^3 \mathbf{t}_0(\xi^1, \xi^2), -\frac{h}{2} \leq \xi^3 \leq \frac{h}{2}, (\xi^1, \xi^2) \in \mathcal{A} \right\} \quad (5.13)$$

5.2.3 Kinematics

Kinematics are what describe the deformations of the shell body. The deformed state, also called the actual configuration, is defined in the same way as the undeformed state but without the subscript $(\cdot)_0$. The midsurface is denoted Ω and the unit vector field normal to the midsurface \mathbf{t} . The mapping between the parametric space and the deformed configuration is given by φ and an area element on the deformed midsurface is computed as $d\Omega = \bar{j} d\xi^1 d\xi^2$ with $\bar{j} = \|\varphi_{,1} \times \varphi_{,2}\|$. Thus, each point on the body can be described in both the undeformed state by φ_0 and the deformed state by φ . The deformation \mathbf{u} is described by these configurations such that:

$$\mathbf{u} = \varphi - \varphi_0 . \quad (5.14)$$

Furthermore, the mapping from the reference configuration $d\varphi_0$ to the actual state $d\varphi$ can be used to describe a deformation gradient \mathbf{F} :

$$d\varphi = \mathbf{F} \cdot d\varphi_0 . \quad (5.15)$$

This deformation gradient includes rigid body movements. It is defined by the base vectors in the reference and the actual configuration, for example by $\mathbf{F} = \mathbf{g}_i \otimes \mathbf{g}_0^i$ and $\mathbf{F}^T = \mathbf{g}_0^i \otimes \mathbf{g}_i$, and can be used for mapping between undeformed and deformed base vectors. The Jacobian of the mapping is given by

$$J = \det(\mathbf{F}) = \frac{j}{j_0} \quad \text{where} \quad j = \mathbf{g}_3 \cdot (\mathbf{g}_1 \times \mathbf{g}_2) . \quad (5.16)$$

Since the gradient includes rigid body motions it cannot be used directly as a measure of strains. Different strain tensors exist but for this project the Green-Lagrange strain tensor \mathbf{E} is used. It is defined by the deformation gradient and the identity tensor \mathbf{I} :

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) = E_{ij} \mathbf{g}_0^i \otimes \mathbf{g}_0^j . \quad (5.17)$$

By inserting the expressions for \mathbf{F} and \mathbf{F}^T into Eq. (5.17) and recalling that the identity tensor is identical to the metric tensor the following is obtained:

$$\mathbf{E} = \frac{1}{2}(g_{ij} - g_{0ij}) \mathbf{g}_0^i \otimes \mathbf{g}_0^j . \quad (5.18)$$

The Green-Lagrange strain coefficients E_{ij} are given by the metric coefficients in the deformed and undeformed state:

$$E_{ij} = \frac{1}{2}(g_{ij} - g_{0ij}) . \quad (5.19)$$

The coefficients refer to the undeformed state contravariant basis $\mathbf{g}_0^i \otimes \mathbf{g}_0^j$ and not an orthonormal basis.

5.2.4 The Kirchhoff-Love hypothesis

With the kinematics of the surface in place Kirchhoff-Love theory is introduced. The theory is based on the "direct approach" meaning that the shell is regarded as a two-dimensional surface and proper three-dimensional behaviour is assumed. Henceforth the derivations are restricted to thin Kirchhoff-Love theory of shells. As mentioned earlier it is analogous to Euler-Bernoulli beam theory as cross-sections are assumed to remain planar

during deformation. The result is a linear strain distribution through the thickness. It is also assumed that the cross-sections remain normal to the middle surface during deformation. This means that the director is always normal to the midsurface, hence the definition of an independent director is superfluous. The shell can be completely represented by its middle surface. The implication mechanically is that shear strains are neglected, like in the Euler-Bernoulli beam model. The assumption is reasonable for thin structures and a good guideline is that the shell radius to thickness ration should be around $R/h > 20$ [11].

Given that both normal strains and transversal shear strains are neglected only in-plane strain coefficients are considered. Eq. (5.18) can thereby be reduced to:

$$\mathbf{E} = E_{\alpha\beta} \mathbf{g}_0^\alpha \otimes \mathbf{g}_0^\beta, \quad (5.20)$$

giving the strain coefficients $E_{\alpha\beta}$:

$$E_{\alpha\beta} = \frac{1}{2}(g_{\alpha\beta} - g_{0\alpha\beta}). \quad (5.21)$$

Since every point on the shell can be described by the middle surface and normal vector alone the base vectors \mathbf{a}_i are given as in Eqs. (5.3) and (5.5) by:

$$\mathbf{a}_\alpha = \boldsymbol{\varphi}_{,\alpha}, \quad (5.22)$$

$$\mathbf{a}_3 = \frac{\boldsymbol{\varphi}_{,1} \times \boldsymbol{\varphi}_{,2}}{\|\boldsymbol{\varphi}_{,1} \times \boldsymbol{\varphi}_{,2}\|} = \mathbf{t}. \quad (5.23)$$

Note that \mathbf{a}_α differs from \mathbf{g}_α in that $\mathbf{a}_\alpha = \boldsymbol{\varphi}_{,\alpha}(\xi^1, \xi^2)$ whereas $\mathbf{g}_\alpha = \mathbf{x}_{,\alpha}(\xi^1, \xi^2, \xi^3)$. Corresponding to Eqs. (5.7) and (5.11) the metric and curvature coefficients defined on the midsurface are given by:

$$a_{\alpha\beta}(\xi^1, \xi^2) = \boldsymbol{\varphi}_{,\alpha} \cdot \boldsymbol{\varphi}_{,\beta}, \quad (5.24)$$

$$\kappa_{\alpha\beta}(\xi^1, \xi^2) = \boldsymbol{\varphi}_{,\alpha} \cdot \mathbf{t}_{,\beta} = -\boldsymbol{\varphi}_{,\alpha\beta} \cdot \mathbf{t}. \quad (5.25)$$

The position vector \mathbf{x} is then given by $\mathbf{x} = \boldsymbol{\varphi}(\xi^1, \xi^2) + \xi^3 \mathbf{t}(\xi^1, \xi^2)$ which gives the base vectors $\mathbf{g}_\alpha = \boldsymbol{\varphi}_{,\alpha} + \xi^3 \mathbf{t}_{,\alpha}$ and the metric coefficients $g_{\alpha\beta}$:

$$g_{\alpha\beta} = a_{\alpha\beta} - 2\xi^3 \kappa_{\alpha\beta} + (\xi^3)^2 \mathbf{t}_{,\alpha} \cdot \mathbf{t}_{,\beta}. \quad (5.26)$$

However, for thin shells the quadratic term in the equation above is omitted leaving:

$$E_{\alpha\beta} = \frac{1}{2}(a_{\alpha\beta} - a_{0\alpha\beta}) + \xi^3 (\kappa_{\alpha\beta} - \kappa_{0\alpha\beta}). \quad (5.27)$$

The strains are represented by the metric and curvature coefficients of the middle surface. It can be seen from the equations that the strains consist of a constant and a linear part. The constant part correspond to the membrane strains $\varepsilon_{\alpha\beta}$ and the linear part to the effect of bending $\rho_{\alpha\beta}$. Eq. (5.27) can thus be rewritten as:

$$E_{\alpha\beta} = \varepsilon_{\alpha\beta} + \xi^3 \rho_{\alpha\beta}. \quad (5.28)$$

5.2.5 Small strain kinematics

Small displacements are assumed for the shell deformation field. Since the displacement vector of the middle surface of the shell is defined as $\mathbf{u} = \boldsymbol{\varphi} - \boldsymbol{\varphi}_0$ the linear membrane and bending strain measures can be derived from the kinematic variables up to first order in \mathbf{u} on the form:

$$\varepsilon_{\alpha\beta} = \frac{1}{2}(\boldsymbol{\varphi}_{0,\alpha} \cdot \mathbf{u}_{,\beta} + \boldsymbol{\varphi}_{0,\beta} \cdot \mathbf{u}_{,\alpha}) , \quad (5.29)$$

$$\rho_{\alpha\beta} = -\mathbf{t}_0 \cdot \mathbf{u}_{,\alpha\beta} - \boldsymbol{\varphi}_{0,\alpha\beta} \cdot \Delta \mathbf{t} , \quad (5.30)$$

where

$$\mathbf{t}_{0,\alpha} = -\bar{j}_0^{-1} \left\{ \mathbf{t}_0 \times [\mathbf{t}_0 (\boldsymbol{\varphi}_{0,1\alpha} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{0,2\alpha})] \right\} , \quad (5.31)$$

$$\Delta \mathbf{t} = -\bar{j}_0^{-1} \left\{ \mathbf{t}_0 \times [\mathbf{t}_0 (\mathbf{u}_{,1} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \mathbf{u}_{0,2})] \right\} \quad (5.32)$$

and $\Delta \mathbf{t}$ is the approximation to $\mathbf{t} - \mathbf{t}_0$ up to the first order in \mathbf{u} .

5.2.6 Equilibrium configuration

The equilibrium configuration is an expression for the balance between external and internal energy of the body. The potential energy of an elastic shell body is expressed by the functional:

$$\Pi[\mathbf{u}] = \int_{\zeta_0} W(\mathbf{u}) dV_0 + \Pi_{ext}[\mathbf{u}] , \quad (5.33)$$

where ext stands for external and W is the internal energy density:

$$W = \frac{1}{2} C^{ijkl} E_{ij} E_{kl} . \quad (5.34)$$

Here C^{ijkl} denotes the contravariant components of the elasticity tensor. Note that this is considering an isotropic material. It is assumed that the elasticity tensor does not vary through the thickness, and the internal energy density can thus be integrated through the thickness giving:

$$\mathscr{W} = \frac{1}{2} \int_{-h/2}^{h/2} C^{\alpha\beta\gamma\delta} E_{\alpha\beta} E_{\gamma\delta} \frac{j_0}{\bar{j}_0} d\xi \simeq \frac{1}{2} C^{\alpha\beta\gamma\delta} \left(h \varepsilon_{\alpha\beta} \varepsilon_{\gamma\delta} + \frac{h^3}{12} \rho_{\alpha\beta} \rho_{\gamma\delta} \right) \quad (5.35)$$

with

$$C^{\alpha\beta\gamma\delta} = \frac{E}{(1-v^2)} \left[v a_0^{\alpha\beta} a_0^{\gamma\delta} + \frac{1}{2} (1-v) (a_0^{\alpha\gamma} a_0^{\beta\delta} + a_0^{\alpha\delta} a_0^{\beta\gamma}) \right] \quad (5.36)$$

and $a_0^{\alpha\gamma} (a_0)_{\gamma\beta} = \delta_\beta^\alpha$. Note that E denotes Young's modulus and v Poisson's ratio in accordance with standard engineering notations. Now, the internal potential energy can be written as:

$$\Pi_{int}[\mathbf{u}] = \int_{\Omega_0} W(\mathbf{u}) d\Omega_0 \quad (5.37)$$

and external potential energy becomes:

$$\Pi_{ext}[\mathbf{u}] = - \int_{\Omega_0} \mathbf{q} \cdot \mathbf{u} \, d\Omega_0 - \int_{\partial\Omega_0} \mathbf{h} \cdot \mathbf{u} \, d\ell_0 . \quad (5.38)$$

Here \mathbf{q} is the body load per unit area, \mathbf{h} the forces per unit length applied on the boundary of the middle surface and $d\ell_0$ is the line element of the boundary of the middle surface. The equilibrium displacement field is obtained by requiring the total potential energy with respect to all admissible variations $\boldsymbol{\eta}$ to be stationary:

$$\delta\Pi[\mathbf{u}, \boldsymbol{\eta}] = \left. \frac{d}{d\varepsilon} \Pi(\mathbf{u} + \varepsilon \boldsymbol{\eta}) \right|_{\varepsilon=0} = 0 . \quad (5.39)$$

Note that the admissible displacement fields and variations must be consistent with the essential boundary conditions. This statement can be expressed in terms of integrals over the parametric space \mathcal{A} as follows:

$$\delta\Pi_{int}[\mathbf{u}, \boldsymbol{\eta}] = \int_{\mathcal{A}} C^{\alpha\beta\gamma\delta} \left(h \varepsilon_{\gamma\delta}(\mathbf{u}) \varepsilon_{\alpha\beta}(\boldsymbol{\eta}) + \frac{h^3}{12} \rho_{\gamma\delta}(\mathbf{u}) \rho_{\alpha\beta}(\boldsymbol{\eta}) \right) \bar{j}_0 \, d\xi^1 \, d\xi^2 , \quad (5.40)$$

$$\delta\Pi_{ext}[\boldsymbol{\eta}] = - \int_{\mathcal{A}} \mathbf{q} \cdot \boldsymbol{\eta} \, \bar{j}_0 \, d\xi^1 \, d\xi^2 - \int_{\partial\mathcal{A}} \mathbf{h} \cdot \boldsymbol{\eta} \, \|\boldsymbol{\varphi}_{0,t}\| \, d\ell_\xi , \quad (5.41)$$

where $\boldsymbol{\varphi}_{0,t}$ is the partial derivative of the middle surface mapping in the tangential direction to the boundary curve.

5.2.7 Elastic constitutive relations

The constitutive equations describe the relation between generalized strains and stresses via a material law. The two parameters previously mentioned, Young's modulus and Poisson's ratio, are the only ones required for description of an isotropic material. The effective membrane $n^{\alpha\beta}$ and bending $m^{\alpha\beta}$ stresses are given by:

$$n^{\alpha\beta} = \frac{\partial \mathcal{W}}{\partial \varepsilon_{\alpha\beta}} = h C^{\alpha\beta\gamma\delta} \varepsilon_{\gamma\delta} , \quad (5.42)$$

$$m^{\alpha\beta} = \frac{\partial \mathcal{W}}{\partial \rho_{\alpha\beta}} = \frac{h^3}{12} C^{\alpha\beta\gamma\delta} \rho_{\gamma\delta} . \quad (5.43)$$

They can be interpreted as force and moment resultants and, by using Voigt's notation, the following expressions are obtained:

$$\mathbf{n} = \begin{pmatrix} n^{11} \\ n^{22} \\ n^{12} \end{pmatrix} = h \mathbf{C} \boldsymbol{\varepsilon} , \quad \mathbf{m} = \begin{pmatrix} m^{11} \\ m^{22} \\ m^{12} \end{pmatrix} = \frac{h^3}{12} \mathbf{C} \boldsymbol{\rho} , \quad (5.44)$$

with

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon^{11} \\ \varepsilon^{22} \\ 2\varepsilon^{12} \end{pmatrix} , \quad \boldsymbol{\rho} = \begin{pmatrix} \rho^{11} \\ \rho^{22} \\ \rho^{12} \end{pmatrix} . \quad (5.45)$$

The matrix C is given by:

$$C = \frac{E}{1-v^2} \begin{pmatrix} (a_0^{11}) & va_0^{11}a_0^{22} + (1-v)(a_0^{12})^2 & a_0^{11}a_0^{12} \\ \vdots & (a_0^{22})^2 & a_0^{22}a_0^{12} \\ symm & \dots & \frac{1}{2}[(1-v)a_0^{11}a_0^{12} - (1+v)(a_0^{12})^2] \end{pmatrix}. \quad (5.46)$$

Now, the following expression for the principle of virtual work is obtained:

$$\delta\Pi[\mathbf{u}, \delta\mathbf{u}] = \int_{\mathcal{A}} (\delta\boldsymbol{\varepsilon} \cdot \mathbf{n} + \delta\boldsymbol{\rho} \cdot \mathbf{m}) \bar{j}_0 \, d\xi^1 \, d\xi^2 + \delta\Pi_{ext}[\delta\mathbf{u}] = 0 \quad (5.47)$$

5.2.8 Galerkin Discretization

Through the principle of virtual work and Galerkin's method the discretized form is obtained. The membrane and bending strain contributions to the stiffness matrix are denoted \mathbf{M} and \mathbf{B} respectively. The shape functions are denoted N_a where a is the shape function index ranging from 1 to the number of degrees of freedom. The strain contributions are expressed in tensor notation as:

$$\mathbf{M}_\alpha = N_{a,\alpha} \boldsymbol{\varphi}_{0,\alpha}, \quad (5.48)$$

$$\mathbf{M}_3 = N_{a,2} \boldsymbol{\varphi}_{0,1} + N_{a,1} \boldsymbol{\varphi}_{0,2} \quad (5.49)$$

and

$$\begin{aligned} \mathbf{B}_\alpha = & -N_{a,\alpha\alpha} \mathbf{t}_0 + \bar{j}_0^{-1} [(\boldsymbol{\varphi}_{0,\alpha\alpha} \times \boldsymbol{\varphi}_{0,2}) N_{a,1} + (\boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{0,\alpha\alpha}) N_{a,2}] \\ & + \bar{j}_0^{-1} (\mathbf{t}_0 \cdot \boldsymbol{\varphi}_{0,\alpha\alpha}) [(\boldsymbol{\varphi}_{0,2} \times \mathbf{t}_0) N_{a,1} + (\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,1}) N_{a,2}], \end{aligned} \quad (5.50)$$

$$\begin{aligned} \mathbf{B}_3 = & -2N_{a,12} \mathbf{t}_0 + 2\bar{j}_0^{-1} [(\boldsymbol{\varphi}_{0,12} \times \boldsymbol{\varphi}_{0,2}) N_{a,1} + (\boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{0,12})] N_{a,2} \\ & + 2\bar{j}_0^{-1} (\mathbf{t}_0 \cdot \boldsymbol{\varphi}_{0,12}) [(\boldsymbol{\varphi}_{0,2} \times \mathbf{t}_0) N_{a,1} + (\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,1}) N_{a,2}]. \end{aligned} \quad (5.51)$$

Introducing the Voigt notation gives the final form of the Galerkin element stiffness matrix as:

$$K^e = \int_{\mathcal{A}} \left[\left(h\mathbf{M}^T \mathbf{C} \mathbf{M} + \frac{h^3}{12} \mathbf{B}^T \mathbf{C} \mathbf{B} \right) \bar{j}_0 \right] d\xi^1 \, d\xi^2 \quad (5.52)$$

Note that none of the steps in the derivation have been specific to isogeometric analysis. The discretization is the same independent of choice of analysis. However, the ease with which the equation system is solved numerically is completely dependent on the use of IGA. Since the curvature contains the second derivative of the geometric description C^1 continuity is needed to fulfill compatibility requirements. As later shown in the examples this is achieved with k-refinement and higher order polynomials in the isogeometric implementation.

5.3 Membrane locking

Locking issues became evident during development of the beam model in the previous chapter. Unsurprisingly, the shell model developed in this chapter also experiences locking problems. However, the beam problems studied with IGA were limited to NURBS data with degree $p = 2$ whereas in the shell implementation k -refinement is utilized and higher order NURBS geometries are investigated. This control over polynomial degree enables investigation of locking phenomena for higher order polynomials. Similar to what is described by Kiendl et al. [26] it is from the results evident that these locking problems associated with low order shell discretizations is alleviated when the degree is raised $p > 2$. Thus, locking in the shell model is reduced by simply raising the polynomial degree.

5.4 Section forces

The section forces, or the generalized in-plane membrane and bending stresses can be computed given the deformed state. The membrane stress is given by \mathbf{n} and the bending stress by \mathbf{m} in Eq. (5.44):

$$\mathbf{n} = \begin{pmatrix} n^{11} \\ n^{22} \\ n^{12} \end{pmatrix} = h\mathbf{C}\boldsymbol{\varepsilon}, \quad \mathbf{m} = \begin{pmatrix} m^{11} \\ m^{22} \\ m^{12} \end{pmatrix} = \frac{h^3}{12}\mathbf{C}\boldsymbol{\rho}.$$

The matrix \mathbf{C} is given in Eq. (5.46) and the variable h denotes the thickness of the shell. The membrane and bending tensors $\boldsymbol{\varepsilon}$ and $\boldsymbol{\rho}$ are computed from the strain displacement matrices \mathbf{B} , \mathbf{M} and the element displacements. Solving for \mathbf{n} without any post-processing returns the stresses in curvilinear coordinates. However, it is typically the Cartesian components of the stresses that are of interest. How to transfer between these coordinate systems is explained in Appendix D.

5.5 Boundary conditions in the IGA shell model

The question of how to handle boundary conditions including rotational restraints in the Kirchhoff-Love shell model follows the same reasoning as in the Euler-Bernoulli beam model. However, it is a bit trickier in two dimensions and there are several implementation options.

The simplest way to prevent or prescribe rotations at the boundaries is to block degrees of freedom on the second row of control points, meaning the row closest to the boundary where the conditions are enforced. This works well for straight edges, it also works well for curved edges as long as they lie in a plane such as the xy -, xz - or yz -plane. For example, if a curved boundary lies in the xz -plane the x and z degrees of freedom on the adjacent row of control points are blocked to prevent rotation.

However, it is sometimes necessary to impose zero rotation without zero displacement at a boundary. The most common case is symmetry boundaries for which degrees of freedom cannot just be blocked but must be forced to move in the same way. For example, it might be necessary to ensure that the x and z degrees of freedom for each control point in the second row moves the same way as its neighbor in the first row. Additionally, the question of how to enforce conditions on arbitrarily curved edges that do not lie in one plane must also be addressed. In these cases simply blocking degrees of freedom is not enough. Both the problems can be solved with the typical methods for imposing constraints in classic finite element analysis [11]. These include for example Lagrange multipliers or penalty methods and one of these approaches is explained in Appendix E.

5.6 Examples

5.6.1 Problem setup

Two problems are chosen for analysis. Both serve to verify the shell model as well as to compare isogeometric analysis to classic finite element analysis. Accuracy and convergence rate on a per-degree-of-freedom basis is studied. The first problem is similar to the problem that was set up for the beam models, a quarter circle curve. The second problem is a classic shell benchmark problem, the pinched hemisphere. The latter is part of the "Shell obstacle course" and is further explained for example by Kiendl [27].

Problem 1: Quarter-circle strip

The quarter circle curve from the beam problems is extruded 0.2 m in depth to create a shell. By setting Poisson's ratio to zero the shell behaves like the beam and results can be compared with the analytical solution for the Timoshenko beam. Just like the beam problem, the shell equivalent is fixed at one end and subjected to loading at the other end, see Fig. 5.3. A distributed vertical line load is applied by integrating over the boundary. The load is set to $Q = -5 \text{ N/m}$ in z -direction and corresponds to the point load on the beam since $Q \cdot b = -5 \text{ N/m} \cdot 0.2 \text{ m} = -1 \text{ N}$. Clamped edges can be enforced in the same way as for beams since the boundary of interest lies in one plane. By blocking all the degrees of freedom of the control points on the boundary and the ones of interest on adjacent boundary rotation is hindered. Since the bottom boundary lies in the xy -plane rotation is prevented by locking the x and z degrees of freedom on the adjacent row.

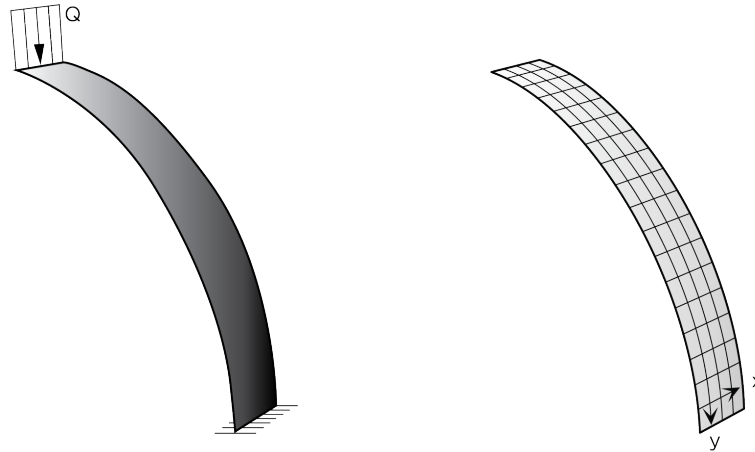


Figure 5.3: *Left:* Setup for *Problem 1*, the shell is fixed at one end and subjected to a distributed vertical line load on the other end. *Right:* Illustration of the local directions on the surface, note that these are not the same as the global coordinate system.

Remaining data for the problem setup is given in the table below:

Variable	Value	Unit
Young's modulus, E	$80 \cdot 10^9$	[Pa]
Thickness of shell, t	0.001	[m]
Width of shell, b	0.2	[m]
Poisson's ratio, ν	0	[-]
Radius, R	1	[m]
Line load, Q	-5	[N/m]

Problem 2: Pinched hemisphere

The second problem consists of a hemisphere that is fixed at its top point and subjected to concentrated forces on the free boundary. The forces are applied in opposite direction along the global x and y axes as shown in Fig. 5.4. The reference for the converged solution is given in terms of radial displacement at the loaded points, $u = 0.0924$ m [27]. Only one quarter of the shell is modeled given the double symmetry. How to apply symmetry conditions in the isogeometric shell model is explained in Appendix E.

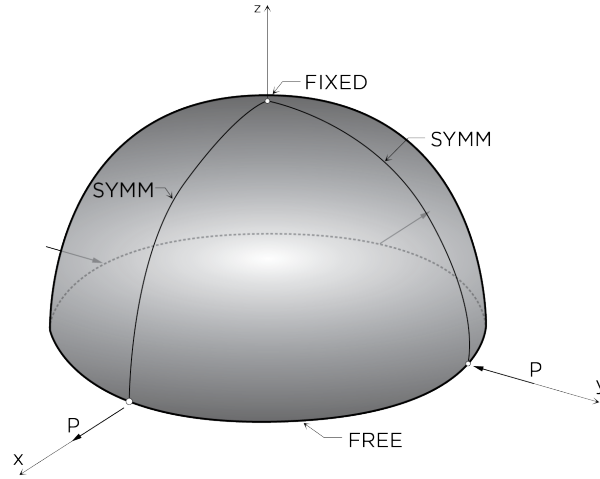


Figure 5.4: Setup for *Problem 2*, the shell is fixed at its top point and subjected to opposite concentrated forces on the free boundary. Only one quarter of the shell is modeled due to symmetry.

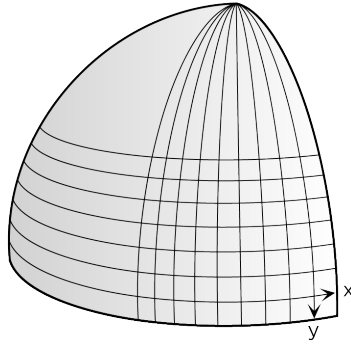


Figure 5.5: Illustration of the local directions on the surface, note that these are not the same as the global coordinate system.

Remaining data for the problem setup is given in the table below:

Variable	Value	Unit
Young's modulus, E	$6.825 \cdot 10^7$	[Pa]
Thickness of shell, t	0.04	[m]
Angle, φ	2π	[rad]
Poisson's ratio, ν	0.3	[-]
Radius, R	10	[m]
Point loads, P	± 2	[N]

5.6.2 Analysis model

The general structure of the code for isogeometric analysis of Kirchhoff-Love shells is given by:

Algorithm 2: General structure for isogeometric analysis of shells

Input : NURBS data in chosen format

```
1 Load initial NURBS data.
2 Perform  $k$ -refinement, i.e. first  $p$ -refinement then  $h$ -refinement.
3 Arrange new control points in control net matrix.
4 Generate connectivity matrices.
5 for  $i = 1$  to number of elements  $e$  do
6   | Get the knot span of  $e$ , both in the  $\xi$  and  $\eta$  directions.
7   | Get the connectivity of  $e$ , i.e. indices of active control points. Both in the  $\xi$  and  $\eta$  directions.
8   | Get coordinates of control points influencing  $e$ .
9   | for  $j = 1$  to number of Gauss points  $gp$  do
10  |   | Compute  $\mathbf{K}^e$  and  $\mathbf{f}^e$ 
11  | end
12  | Assemble element contributions to global form.
13 end
14 Locate control point indices of edges and wherever loads are to be applied. E.g. if one edge is located on the
    |  $x$ -axis the condition  $y = 0$  can be used to obtain this information.
15 for  $i = 1$  to number of elements  $e$  do
16  | Perform operations from rows 6 and 7 again.
17  | Extract control point indices for load application and compute respective degrees of freedom.
18  | for  $j = 1$  to number of Gauss points  $gp$  do
19  |   | Integrate over the boundary to obtain the additions from line loads to  $\mathbf{f}$ .
20  | end
21 end
22 Apply boundary conditions.
23 Apply point loads.
24 Solve the system for  $\mathbf{a} = \mathbf{K} \setminus \mathbf{f}$ .
25 for  $i = 1$  to number of elements  $e$  do
26  | Perform operations from rows 6 and 7 again.
27  | for  $j = 1$  to number of Gauss points  $gp$  do
28  |   | Compute the membrane and bending in-plane stresses as described in Ch. 5.4.
29  |   | Compute the displacements at arbitrary physical coordinates in the same way as discussed in Ch. 3.5.
30  | end
31 end
```

Output : Displacements and in-plane generalized stresses.

5.6.3 Results

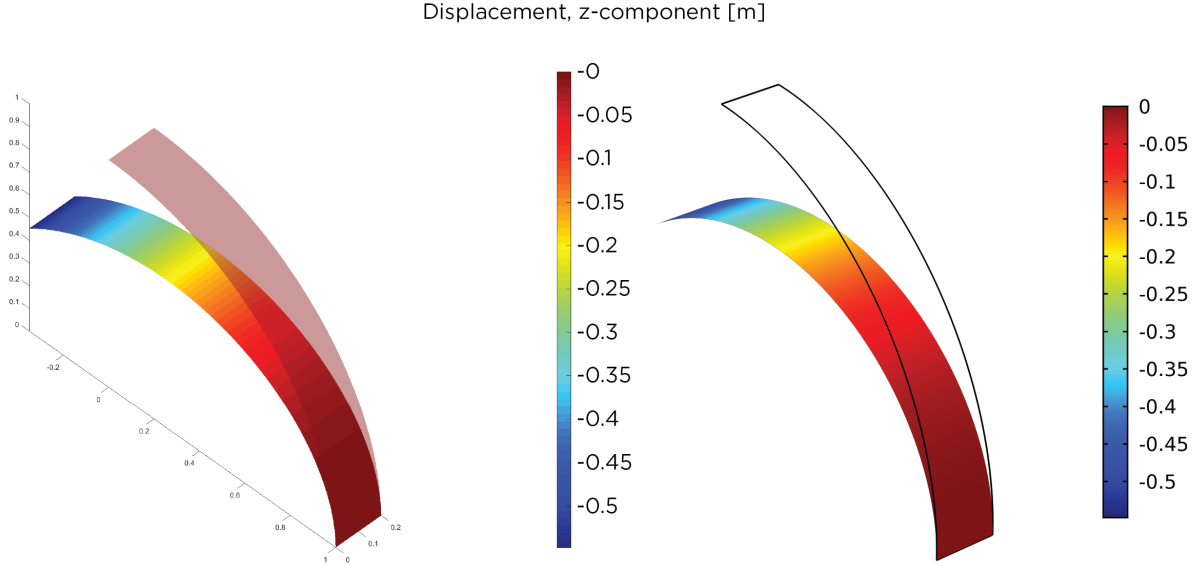


Figure 5.6: Displacement in global z -direction, IGA shell model on the right and FEA solution on the left.

Problem 1: Verification of results

Just as for the beam analysis the analytical solution for the Timoshenko beam is used to verify the IGA shell model. The solution is also used to verify the set-up of the FEA model in COMSOL Multiphysics®. In Fig. 5.6 the displacement plots for converged results of respective models are shown. The end displacement of 0.58 m is the same as was given by the analytical solution for the same material data.

With the displacement corroborated, the FEA model is used for verification of in-plane bending and membrane stresses with Poisson's ratio raised to $\nu = 0.3$. The results from the IGA model for in-plane membrane stress is shown in the left column of Fig. 5.7 - 5.9. Corresponding plots for the shell model with FEA are depicted in the right column. The correlation between the models is very good.

Equivalent plots for bending moments are given in Fig. 5.10 - 5.12. Again the correlation is strong and the values match the solution obtained with FEA very well. Note the slight inaccuracies and oscillations in the FEA plots on the body of the shell, these do not appear in the IGA shell model. Note also that graphical discrepancies might be caused by different definitions of local directions on the surface for the different models, see Fig. 5.3.

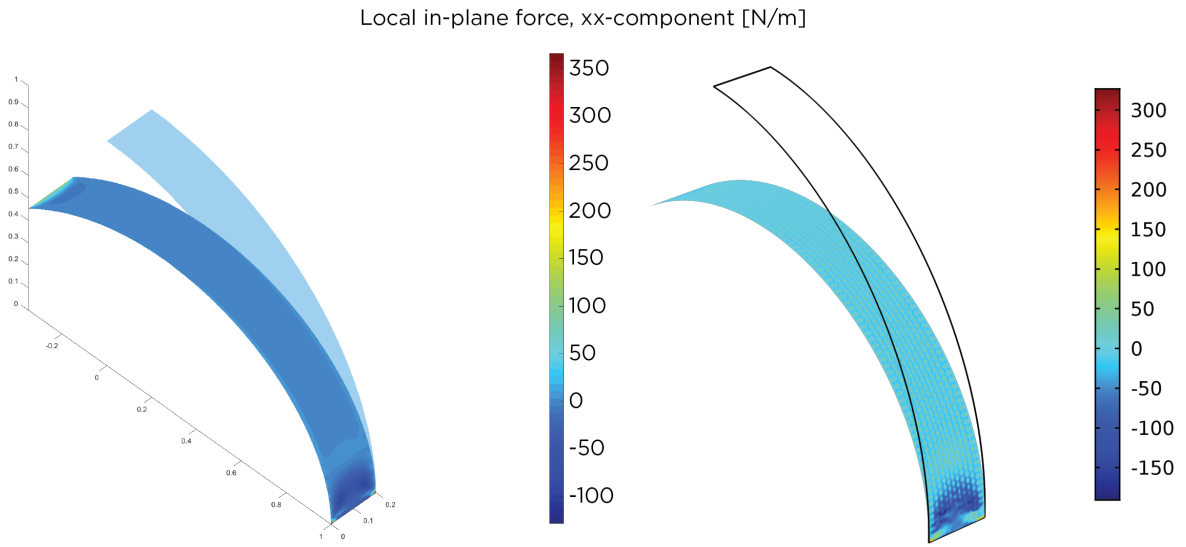


Figure 5.7: Local in-plane force, xx -component. IGA shell model on the left and FEA solution on the right.

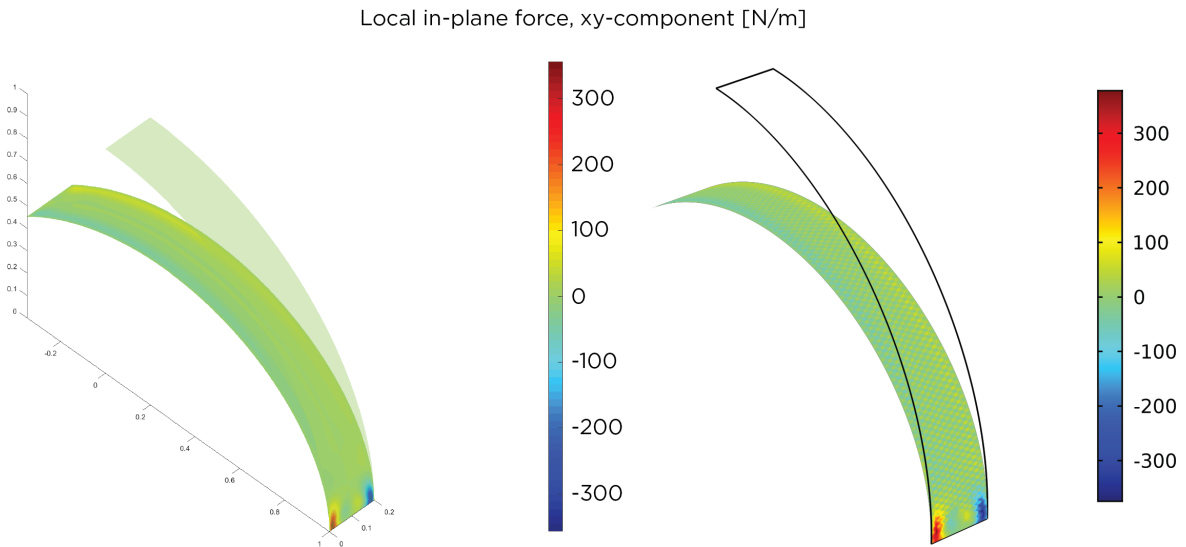


Figure 5.8: Local in-plane force, xy -component. IGA shell model on the left and FEA solution on the right.

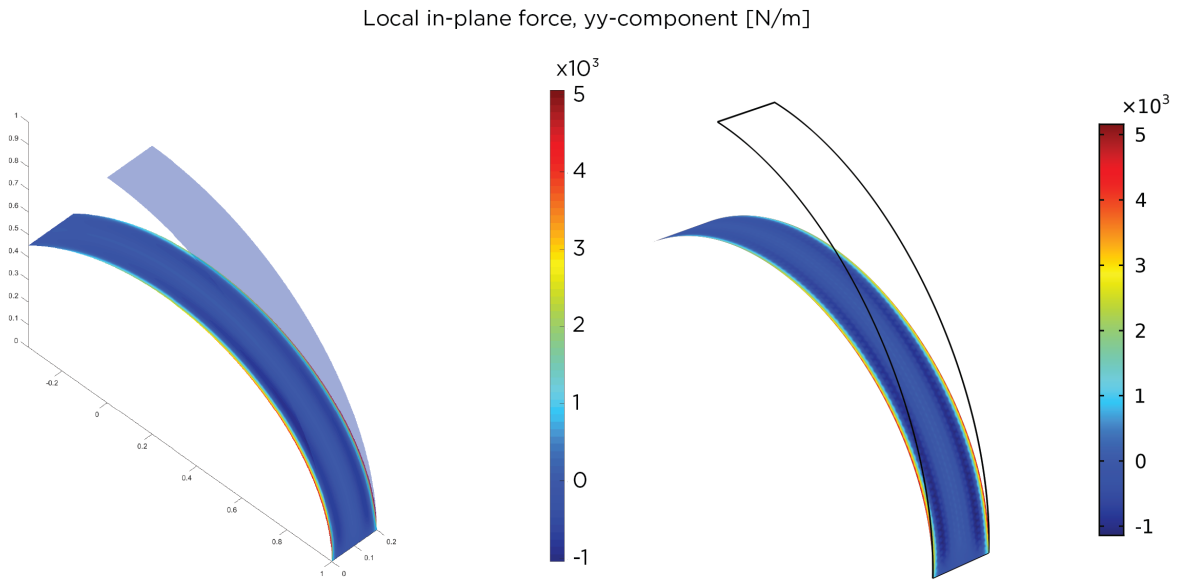


Figure 5.9: Local in-plane force, yy -component. IGA shell model on the left and FEA solution on the right.

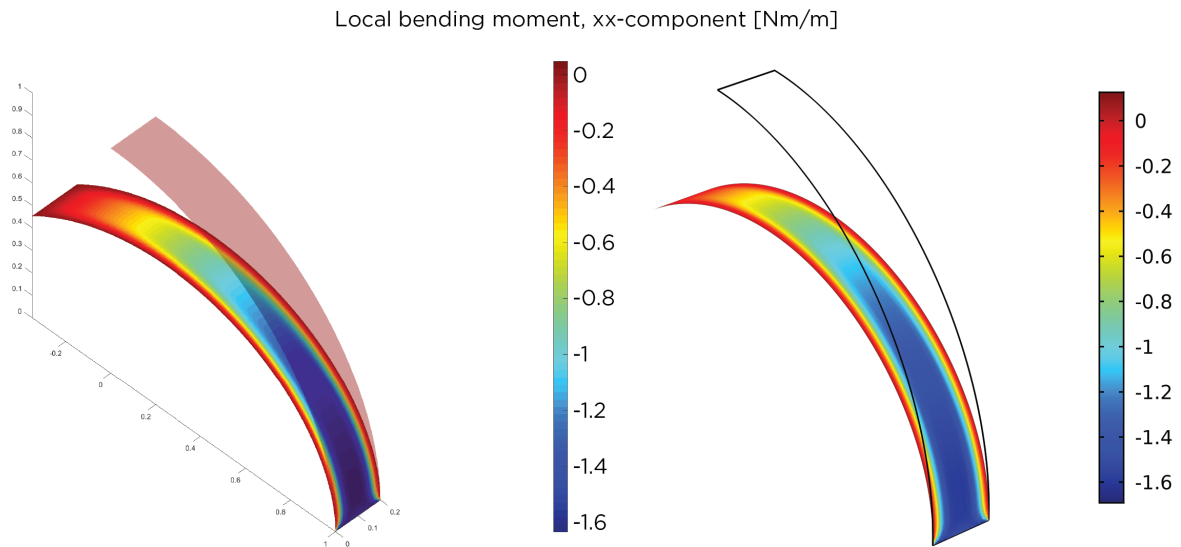


Figure 5.10: Local bending moment, xx -component. IGA shell model on the left and FEA solution on the right.

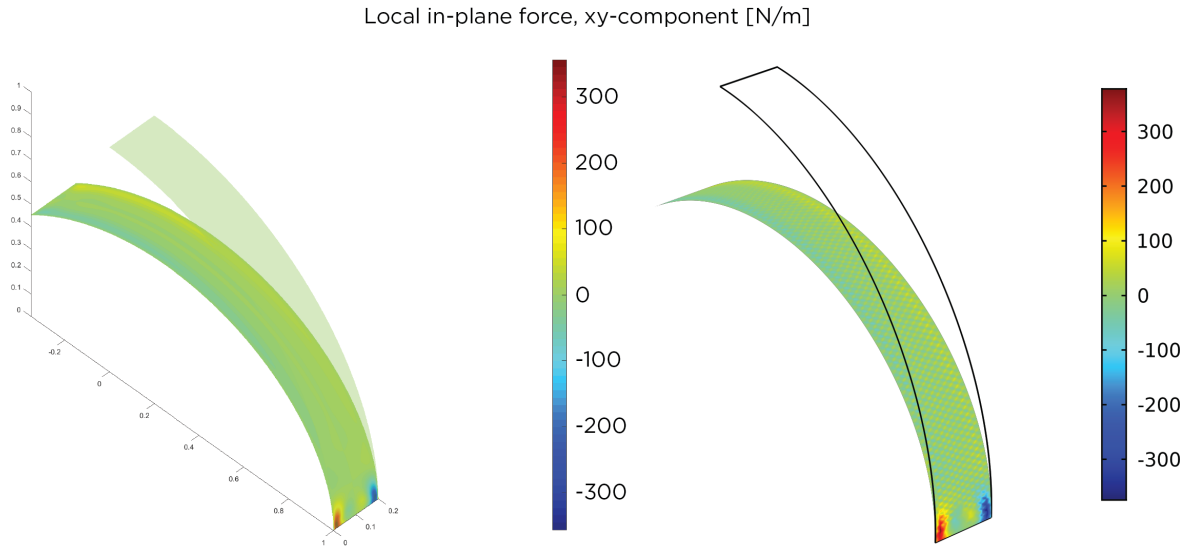


Figure 5.11: Local bending moment, xy -component. IGA shell model on the left and FEA solution on the right.

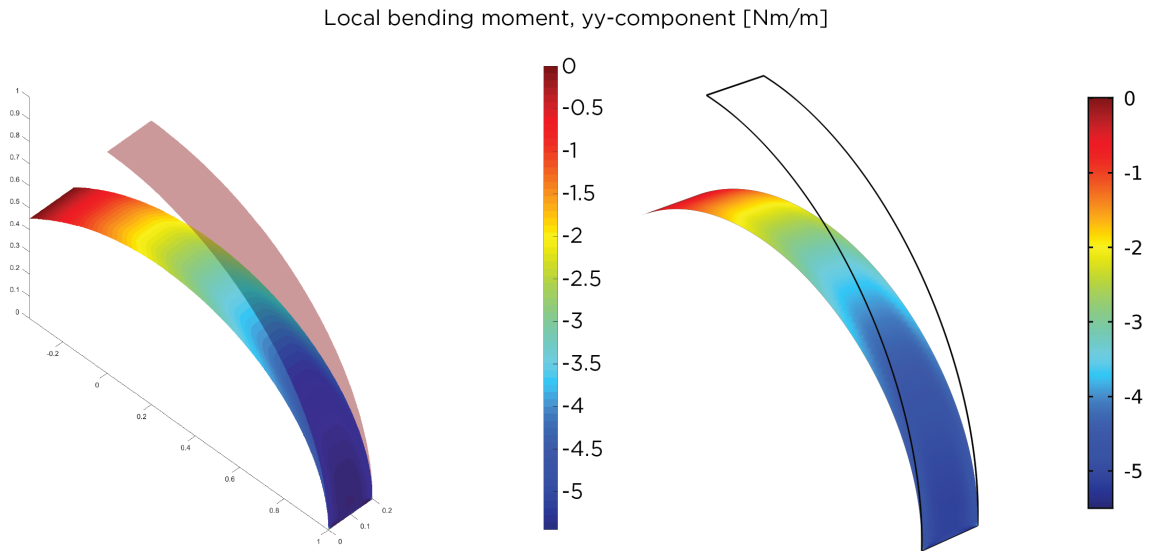


Figure 5.12: Local bending moment, yy -component. IGA shell model on the left and FEA solution on the right.

Problem 1: Comparison of accuracy and convergence rate

The basis for comparison is given by the relative error in strain energy versus number of degrees of freedom $ndof$. Three results are produced with the finite element model: one for a free triangular mesh, the second for a free quadrilateral mesh and the third for a mapped quadrilateral mesh. The model is based on Reissner-Mindlin theory and solved with MITC elements [28]. For the model solved with isogeometric analysis six results are produced. The polynomial degrees range from $p = 2$ to $p = 6$ and $q = 2$ to $q = 5$ where p represents the direction of the curved boundary and q the direction of the straight boundary on the surface.

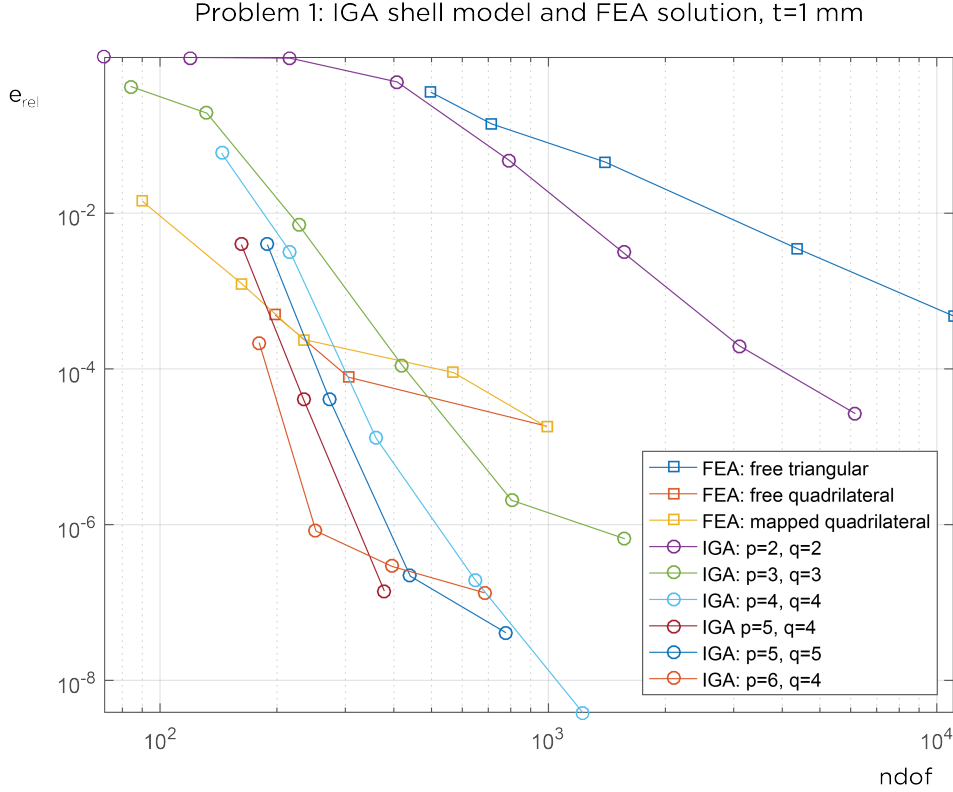


Figure 5.13: Comparison between the FEA and IGA shell models. Isogeometric solutions are denoted by circles and FEA solutions by squares.

For the quantitative comparison, we consider again the strain energy of the solution:

$$E = \frac{1}{2} \|u\|^2, \quad (5.53)$$

$$\|\cdot\| = \sqrt{a(\cdot, \cdot)}, \quad (5.54)$$

where $a(\cdot, \cdot)$ pertains to the "energy" product of the mechanical equilibrium. For a discrete approximation, we then obtain:

$$E_h = \frac{1}{2} a(u_h, u_h) = \frac{1}{2} \mathbf{a}^T \mathbf{K} \mathbf{a}, \quad (5.55)$$

With h denoting the approximate solution and ref the analytical reference solution, the relative error can be expressed as:

$$e_{rel} = \left| \frac{E_{ref} - E_h}{E_{ref}} \right|. \quad (5.56)$$

IGA outperforms the FEA solution with triangular meshing in terms of both accuracy and convergence rate. Even the isogeometric solution with $p = q = 2$ which is known to experience substantial locking problems gives better results than solving the problem with the triangular mesh in FEA. The IGA shell model shows increasing accuracy and convergence rate as k -refinement is performed and higher polynomial degrees are obtained. There seems to be a limit to how much the polynomial degree of the basis functions along η , the direction of the straight boundary, can be raised and still contribute to improving the solution. By comparing the line representing $p = 5$ and $q = 4$ with the one where $p = q = 5$ it can be seen that raising q does not give a better solution, it only translates the line.

All isogeometric solutions except $p = q = 2$ appear to have higher convergence rate than the FEA solutions with mapped and free quadrilateral meshing respectively. However, it is only the isogeometric solution with $p = 6$ and $q = 4$ that outperforms FEA in terms of accuracy for few degrees of freedom. Worth noting is that the first point on the line for FEA with mapped quadrilateral meshing consists of two elements with 45 degrees of freedom each. The general conclusion is that IGA works very well for this problem setup.

Since the theory behind the model is adapted for thin shells it is also of interest to see how the solutions converge for thicker shells. Three different thicknesses are investigated: 1 mm, 10 mm and 100 mm. The results are shown in Fig. 5.14. Noting the values on the vertical axis it is clearly seen that the isogeometric solution converges towards larger errors with increasing thickness. This is expected since the reference solution is based in Timoshenko beam theory whereas the Kirchhoff love shell theory is an extension of the Euler-Bernoulli beam theory. Thus, it is not possible to come closer to the reference solution than the errors shown with this shell theory.

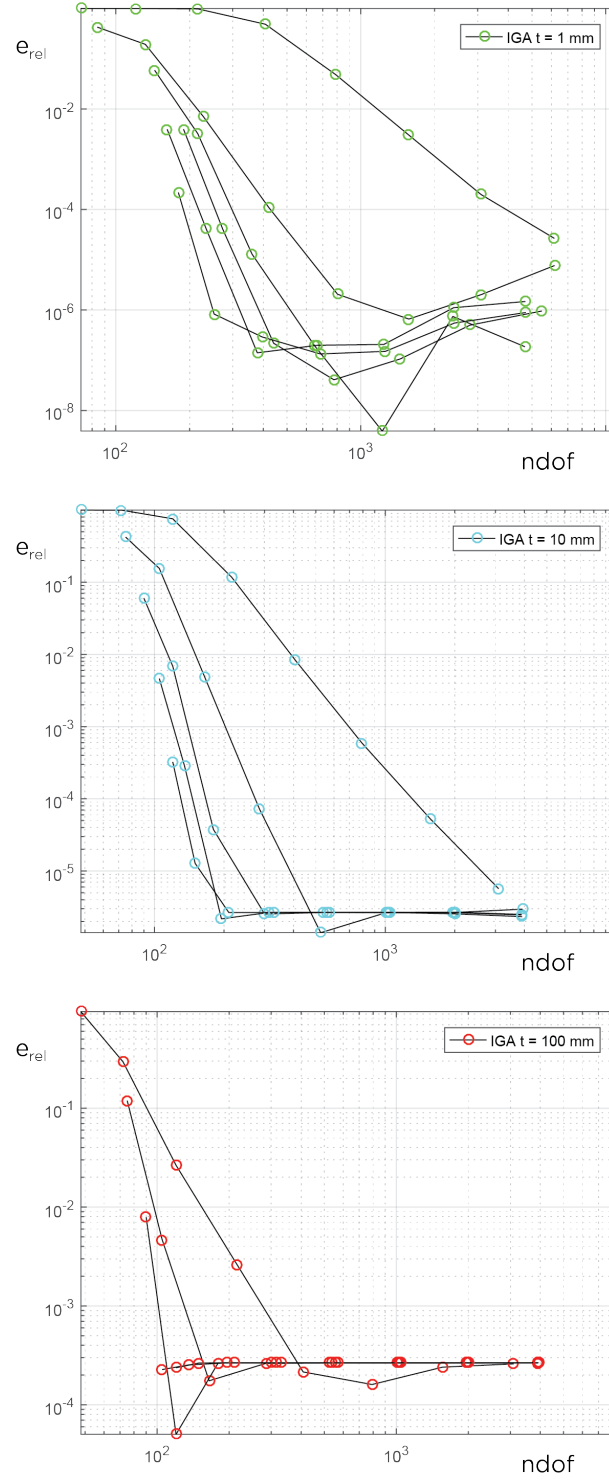


Figure 5.14: From top to bottom: thickness of 1 mm, 10 mm and 100 mm. For the thinnest shell the plot from the IGA shell model converges towards an error of 10^{-7} . For the medium thickness the relative error lands at about 10^{-6} . Finally, for the thickest shell the convergence stops around an error of 10^{-4} .

Problem 2: Verification of results

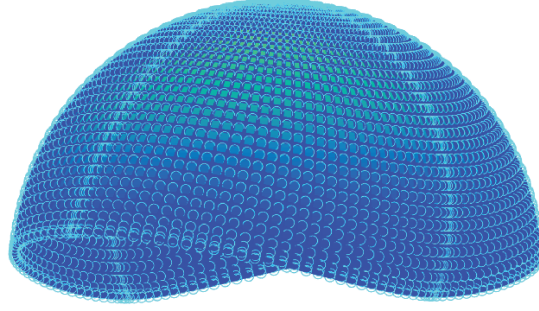


Figure 5.15: Shell after deformation, displacements are scaled by factor 25. The circles on the surface represent the control points.

The reference solution given for the pinched hemisphere benchmark is a radial displacement of $u = 0.0924 \text{ m}$. The deformed configuration of the shell is given in Fig. 5.15. The convergence to the reference value for the loaded point on the x -axis is shown in Fig. 5.16. As can be seen, higher polynomial degrees give both a better initial approximation and stronger convergence rate.

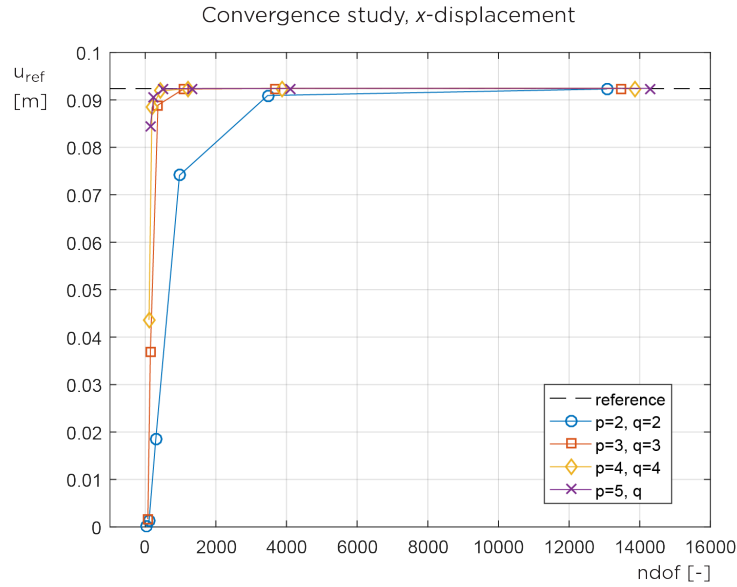


Figure 5.16: Convergence of radial displacement at the loaded point on the global x -axis.

The displacement plots for displacement in x , y and z -direction respectively of the converged models are given in Fig. 5.17 - 5.19. The IGA solution is given on the left and the FEA solution on the right. The plots correlate well. All displacements are scaled with a factor of 25. With the displacement corroborated, the in-plane bending and membrane stresses are compared. The results from the IGA model for in-plane membrane stress are shown in the left column of Fig. 5.20 - 5.22. Corresponding plots for the shell model with FEA are depicted in the right column. The correlation between the models is good. Equivalent plots but for in-plane bending stresses are given in Fig. 5.23 - 5.25. Again the correlation is strong and the values match the solution obtained with FEA well. Here, the FE solution is obtained with the software AbaqusTM. Note that the color maps do not correspond exactly. Differences in extreme values are a result of the color mapping. Other graphical discrepancies might be caused by different definitions of local directions on the surface for the different models, see Fig. 5.5.

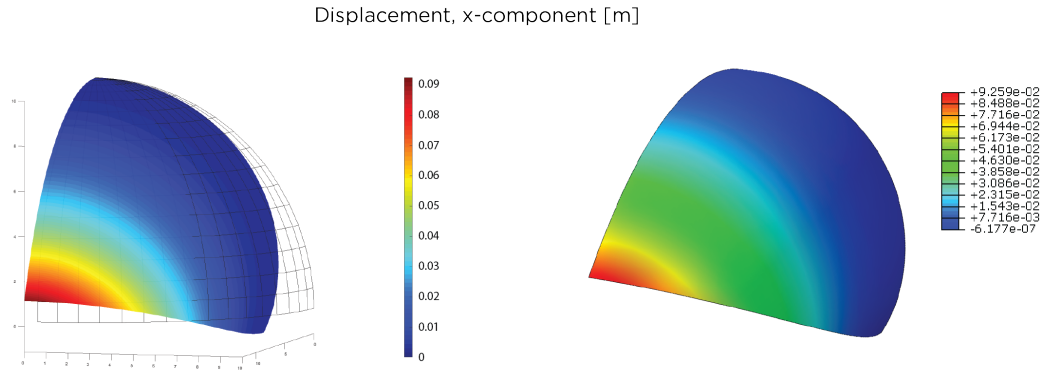


Figure 5.17: Displacement in global x -direction, IGA shell model on the left and FEA solution on the right.

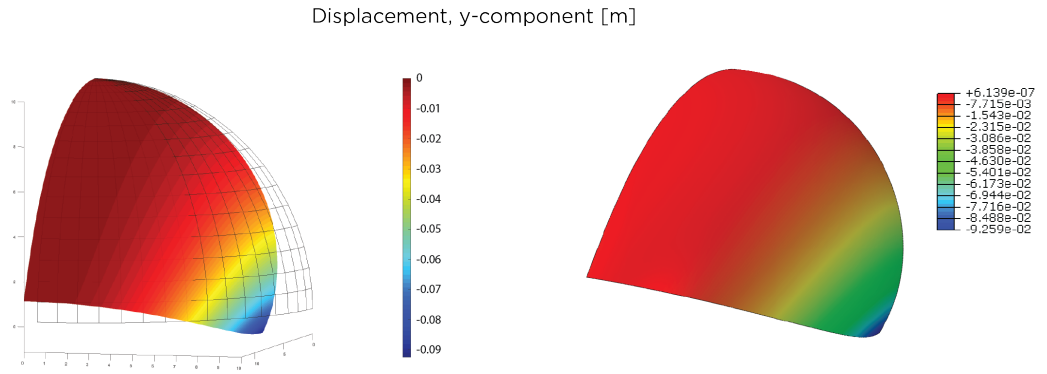


Figure 5.18: Displacement in global y -direction, IGA shell model on the left and FEA solution on the right.

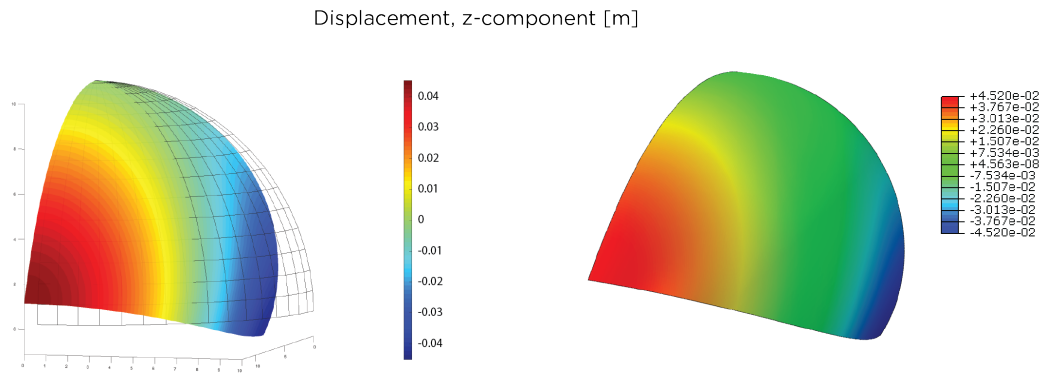


Figure 5.19: Displacement in global z -direction, IGA shell model on the left and FEA solution on the right.

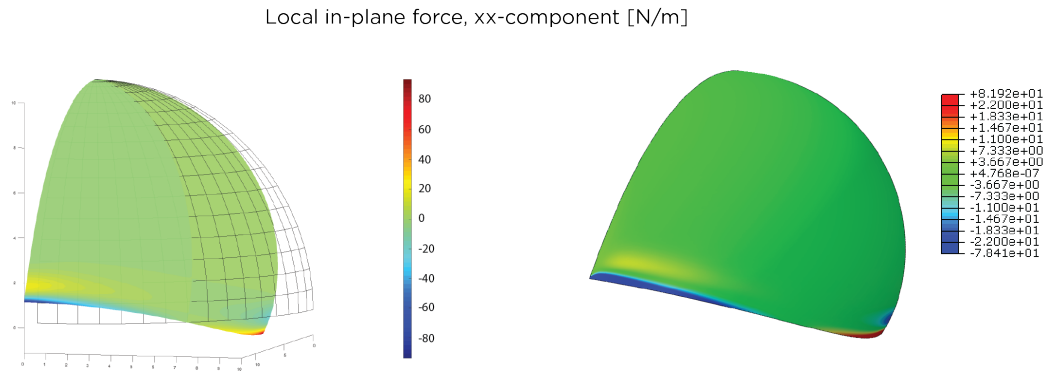


Figure 5.20: Local in-plane force, xx -component. IGA shell model on the left and FEA solution on the right.

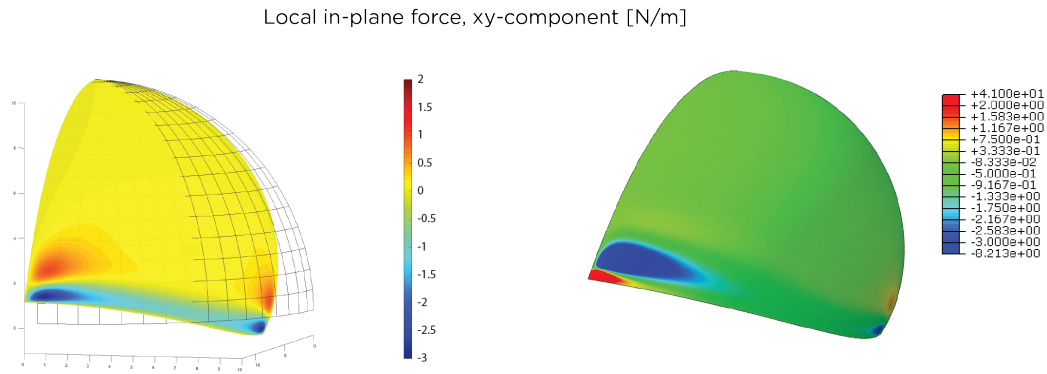


Figure 5.21: Local in-plane force, xy -component. IGA shell model on the left and FEA solution on the right.

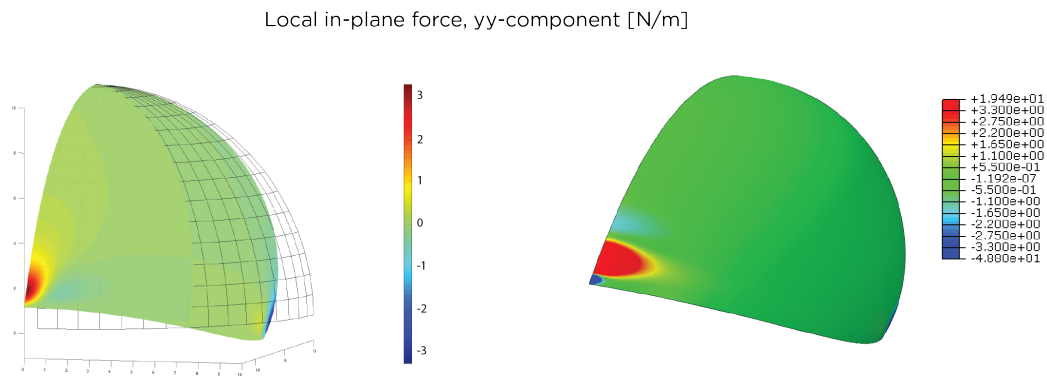


Figure 5.22: Local in-plane force, yy -component. IGA shell model on the left and FEA solution on the right.

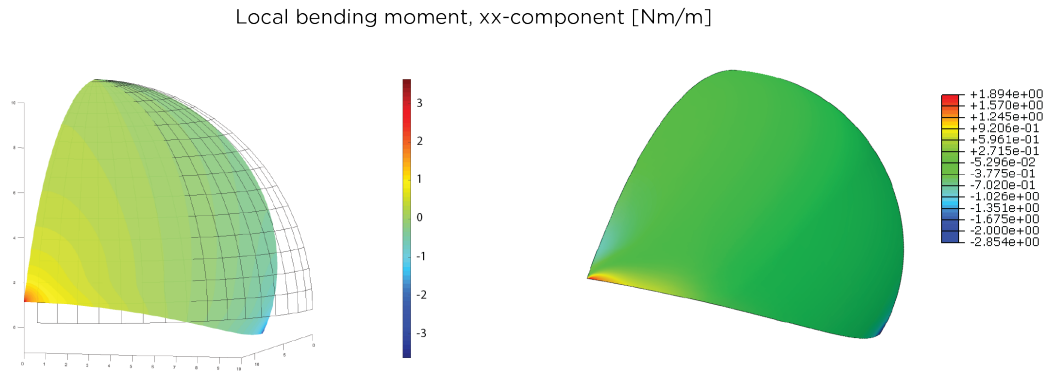


Figure 5.23: Local bending moment, xx -component. IGA shell model on the left and FEA solution on the right.

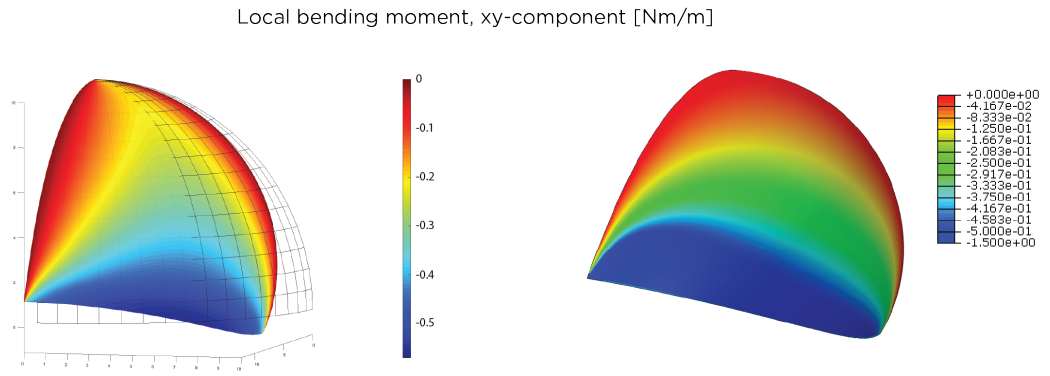


Figure 5.24: Local bending moment, xy -component. IGA shell model on the left and FEA solution on the right.

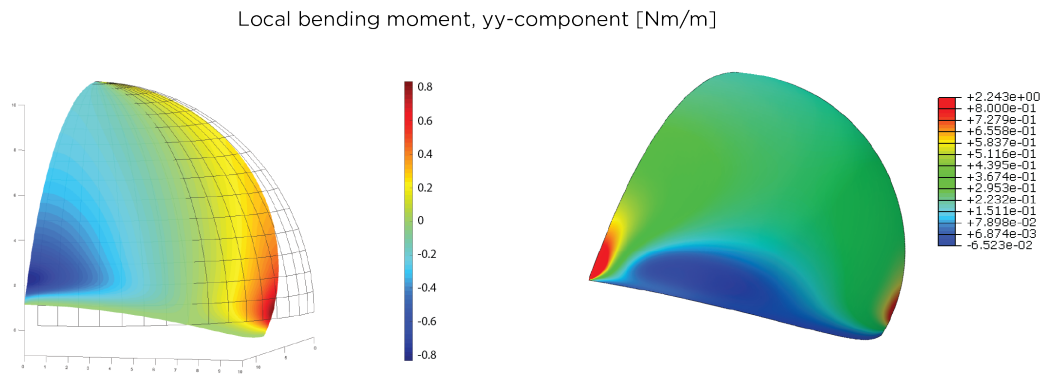


Figure 5.25: Local bending moment, yy -component. IGA shell model on the left and FEA solution on the right.

Problem 2: Comparison of accuracy and convergence rate

For the pinched hemisphere, like for the other examples investigated in this project, the relative error as given in Eq. (5.45) is investigated. Again, the finite element analysis software COMSOL Multiphysics® is used for comparison. Two FE solutions are computed, one with a free triangular mesh and one with a free quadrilateral mesh. The mapped quadrilateral mesh that was used in the previous shell comparison is not applicable for this double curved geometry. The isogeometric analysis solutions range from degree $p = q = 2$ to $p = q = 7$.

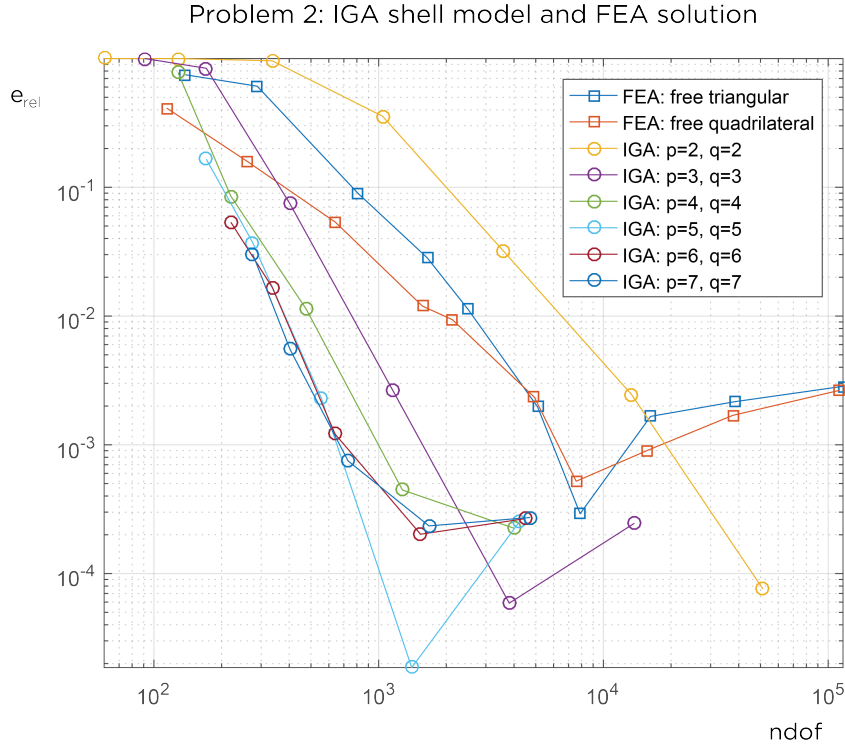


Figure 5.26: Comparison between the FEA and IGA shell models. Isogeometric solutions are denoted by circles and FEA solutions by squares.

The results, seen in Fig. 5.26, show similar accuracy for both IGA and FEA. The locking problems in the isogeometric model with $p = q = 2$ are clearly seen here, it is the least suitable solution. The finite element solution with free quadrilateral mesh performs well, however the convergence is not quite as smooth as with the IGA shell model. The most accurate analysis setup with the highest convergence rate is given by IGA with $p = q = 7$. In general it is seen that the IGA solutions with $p, q > 3$ have better convergence rate than what is given by the finite element analyses.

6 Case Study

A case study is performed to test the IGA shell model developed in the previous chapter. The subject for the study is a thin shell sculpture called "Crest". It is chosen in collaboration with external supervisor Civil Engineer Rasti Bartek for its geometrical and structural properties. One of the objectives is to study how the shell model performs when a complex geometry is analysed instead of a classic benchmark problem. Another objective is to gain insight into what it is like to work with isogeometric analysis from a project process perspective. Finite element analyses are run by the external supervisor in parallel with the isogeometric analysis and the results and the working process are compared.

6.1 Description



Figure 6.1: The sculpture "Crest" designed by Zaha Hadid Architects in collaboration with BuroHappold Engineering and LiteStructures. Photograph courtesy of ©Ed Reeve.

The sculpture "Crest" was a temporary installation at the Victoria and Albert Museum in London the summer of 2014. The design and production method was the result of collaborative work between BuroHappold Engineering, Zaha Hadid Architects and LiteStructures. It was a free standing sculpture only depending on friction and self-weight to counteract critical loads [29].

The design was informed by the surroundings. It spanned an oval fountain elegantly with the shape of the shell edges reflecting the shape of the fountain perimeter, see Fig. 6.2. The design was achieved by cutting out portions from a cone. The resulting shape consists of a single curved developable surface, meaning that it can be formed from a flat sheet and bent into shape [29], [30].

Site limitations led to a set of structural design decisions. Due to restraints regarding the size of sheets that could be transported into the museum courtyard the sculpture was assembled from flat aluminum pieces that were bolted together in two layers on site. However, this panelization is not part of the case study analysis. Site restraints regarding limited impact on the bottom of the fountain was another aspect influencing design decisions. This condition together with results from the load analyses strongly influenced the configuration of the supports.



Figure 6.2: Top view of "Crest". Photograph courtesy of ©Ed Reeve.

The load cases studied included thermal, wind and accidental loading, the latter in the shape of a point load equivalent to a child standing on the surface. A critical load case proved to be wind uplift. To mitigate it the weight of the shell was increased by adding panels at the supports and a rubber band was added to increase friction to the ground. The geometrical and structural design decisions in combination resulted in a stable shell that acted as a highly efficient structural system.

Besides a good design there are multiple other reasons for why this is a good choice for the case study. As required, the geometry is made up from a single patch NURBS surface. Additionally, with a constant thickness of 16 mm and a span of 10 m the sculpture falls within the definition of thin shells. The surface is non-symmetric and has curved boundaries and is therefore complex to mesh using classic FEA. There is also a lot of data available for comparing IGA with the original FE analyses, both of simplified models and of the full structure. Important to note is that there is no known analytical solution for this shell, the only things to go upon are the previous FE analysis, data from the conference article written about the shell and the fact that the installation was successful [29].

6.2 Limitations and simplifications

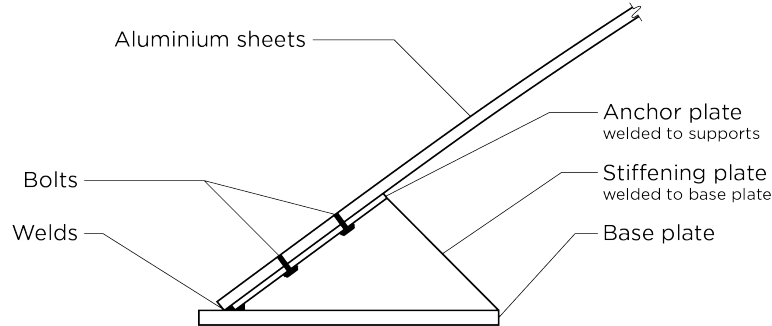


Figure 6.3: Schematic picture of configuration of edge supports for the "Crest" shell.

As mentioned previously, the design of the supports was influenced by site conditions. The resulting configuration is shown in Fig. 6.3. The base plate, added to increase weight, is welded to stiffening plates and together they are supporting the shell. In terms of structural behaviour this results in a connection that hinders rotations at the supports. Thus, it is a reasonable simplification to model the supporting edges as fixed. The actual supports also have two cables connecting the edges, see the left side illustration in Fig. 6.4. These were added as a security measure against horizontal spreading of the shell since the sculpture was located in a public place without supervision. The right side illustration in the same figure shows the modeled simplified supports.

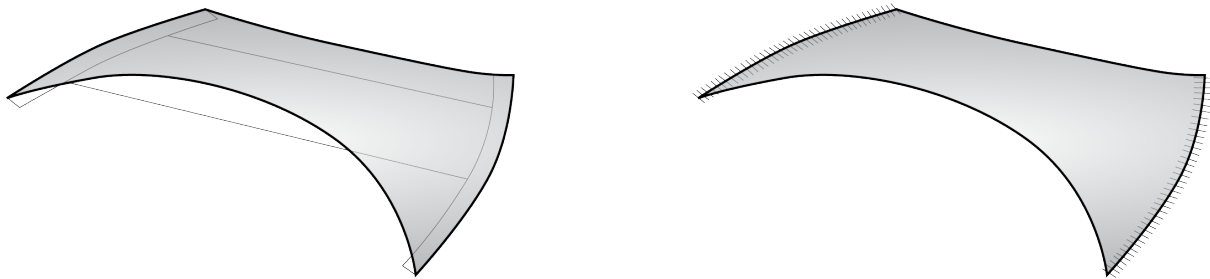


Figure 6.4: Actual boundary conditions (left) and simplified boundary (right) used for analysis of the "Crest" shell.

Another simplification is omitting the panelization. The joints required for the production and installation process are not included in the analysis. The shell is instead treated as a continuous single surface without taking shear transfer between the layers into account. This analysis decision is in line with the initial architectural intent which was to express a continuous surface.

Finally, the analysis was restricted to treat only the self-weight load case. Point loads reflecting accidental loading and wind load effects could easily be added to the IGA model. However, extensive finite element analysis data was accessible for the self-weight load scenario whereas analyses of the other scenarios were limited. Additionally, the main in-use scenario is when the shell is subjected to only self-weight.

6.3 Analysis

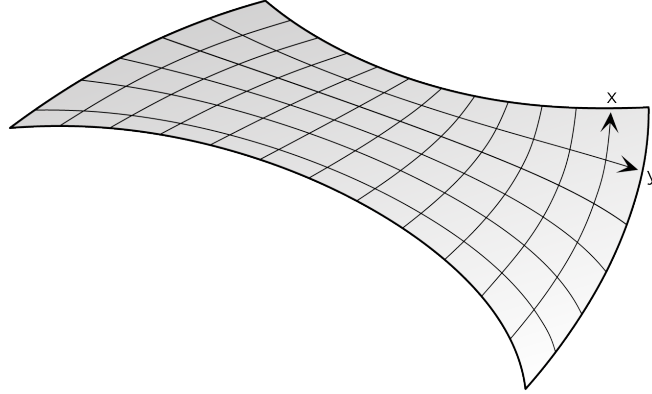


Figure 6.5: Illustration of local directions on the surface, note that these are the same as the global coordinate system.

The geometry input is defined by a single patch NURBS surface delivered as a Rhinoceros[®] model. This reflects one of the most common processes in the industry where a design is delivered in an NURBS based environment. The isogeometric analysis process then starts with the NURBS data being structured and exported using the Grasshopper[®] component that was written for this project. Ideally the entire analysis would take place within the Grasshopper[®] environment, but for the time being it needs to be exported since the shell model is implemented in Matlab[®]. The remaining data required to complete the model is then added in Matlab[®], see the table below:

Variable	Value	Unit
Young's modulus, E	$70 \cdot 10^9$	[Pa]
Density, ρ	2710	[kg/m ³]
Thickness, t	0.008	[m]
Poisson's ratio, ν	0.34	[-]
Face load, G	$-2 \cdot t \cdot 9.81 \cdot \rho$	[N/m ²]

Note the factor 2 on the face load G . It is included since the actual shell consists of two layers of 8 mm panels whereas only one layer is included in the analysis model.

Thereafter, the IGA mesh is refined within the shell model utilizing k -refinement until the maximum and minimum displacements in global z -direction converge. Two different NURBS surfaces of nearly the same geometry are analysed. The model initially received is defined by 601×38 control points. The second model is created by 4×4 control points. The first model was analysed for the real project and is therefore the geometry that the FEA deflection results for comparison originate from. However, there are no original results of in-plane stresses available, thus these must be produced separately. In order to save time a reduced model is created for this stress comparison.

The FEA results for the original geometry are obtained with both the software ETABS[®] by CSI and the software Robot[™] by Autodesk[®] separately. These applications take meshes as input. Meshing of the NURBS surface is done with a Grasshopper[®] component called MeshMachine and then exported for the FE analysis.

The FE analysis of the second geometry is performed with the software Abaqus[™]. The input is the NURBS surface exported from Rhinoceros[®] in SAT-format. Worth noting is that Abaqus[™] alerts the user about imprecise geometry upon importing the file.

6.4 Results

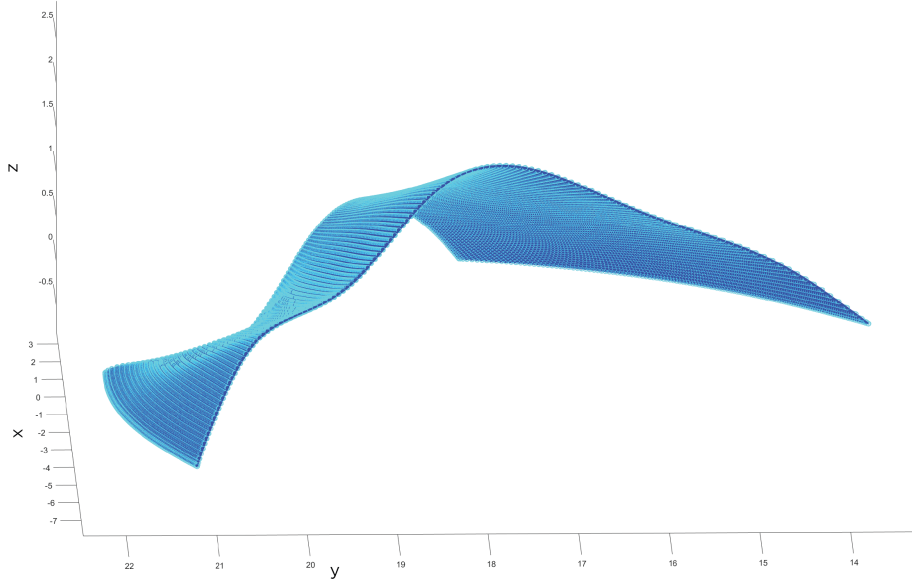


Figure 6.6: Deformed shape of "Crest" obtained with the IGA shell model. The displacements are scaled by factor 50.

Initial geometry

The analysis output for the initial geometry consists of displacements in z -direction. The maximum deflection obtained with the IGA shell model is 13.3 mm and the minimum value -18.8 mm. Corresponding values from the FE models are $u_{max} = 11$ mm and $u_{min} = -17$ mm from RobotTM as well as $u_{max} = 11.4$ mm and $u_{min} = -17.1$ mm from ETABS[®]. The difference is likely related to the meshing process required for FEA. While the surface can be exactly described in IGA the FEA meshes are only geometrical approximations. The results are shown in Fig. 6.7 - 6.9.

Reduced geometry

The analysis output for the second geometry consists of deformations and in-plane membrane and bending stresses. The deformed shape from the isogeometric analysis is depicted in Fig. 6.6. The maximum displacement is 6.5 mm and the minimum displacement -10 mm. The convergence to maximum and minimum displacements of the IGA shell model are shown in Fig. 6.10 - 6.11. It can be seen that higher polynomial degrees give better initial approximations and faster convergence. Furthermore, it is shown that the values correspond well with the FE results from AbaqusTM in Fig. 6.12 - 6.13. However, the FEA model requires approximately 450000 degrees of freedom to reach about the same results that are obtained with 14000 degrees of freedom in the IGA shell model.

The results for in-plane forces and bending moments obtained with IGA and FEA are shown in Fig. 6.14 - 6.25. Note that the color maps do not correspond exactly, but the values and patterns from both analyses match each other well. Note also that graphical discrepancies might be caused by different definitions of local directions on the surface for the different models, see Fig. 6.5.

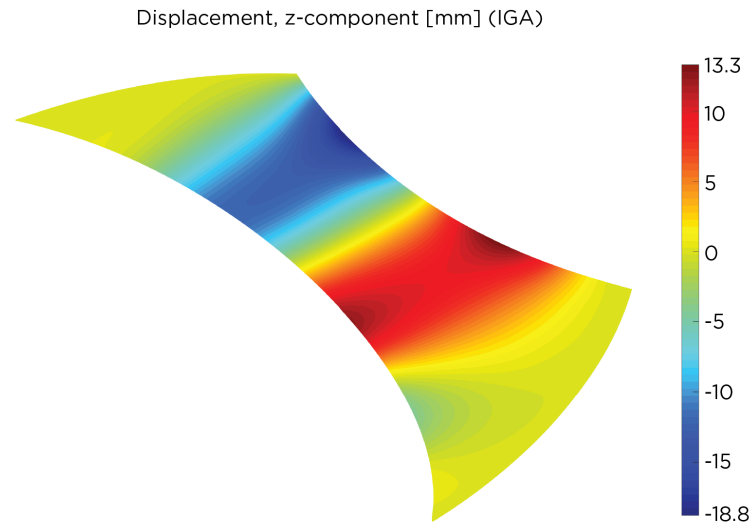


Figure 6.7: Displacement in z -direction. Results from the IGA shell model.

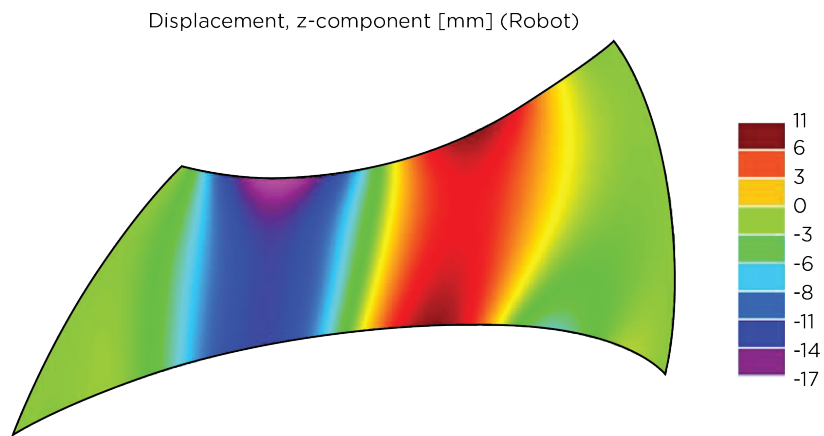


Figure 6.8: Displacement in z -direction. FEA results from RobotTM.

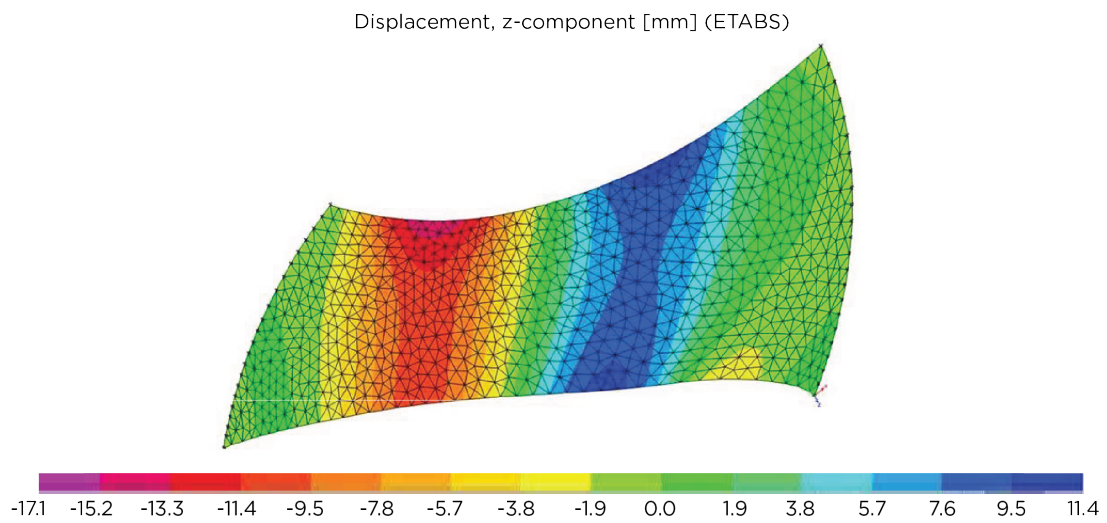


Figure 6.9: Displacement in z -direction. FEA results from ETABS[®].

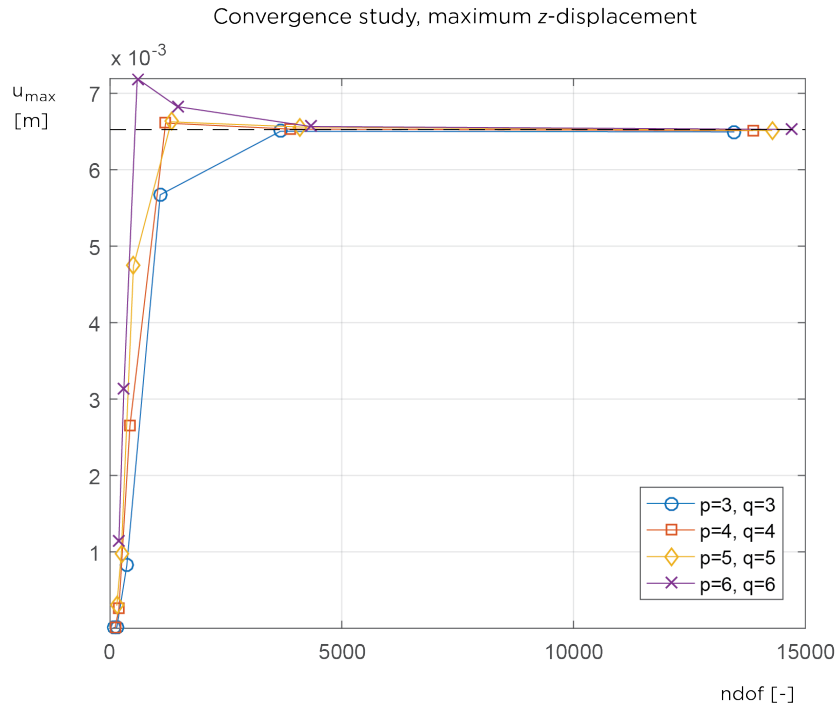


Figure 6.10: Maximum z -displacement convergence for the IGA shell model.

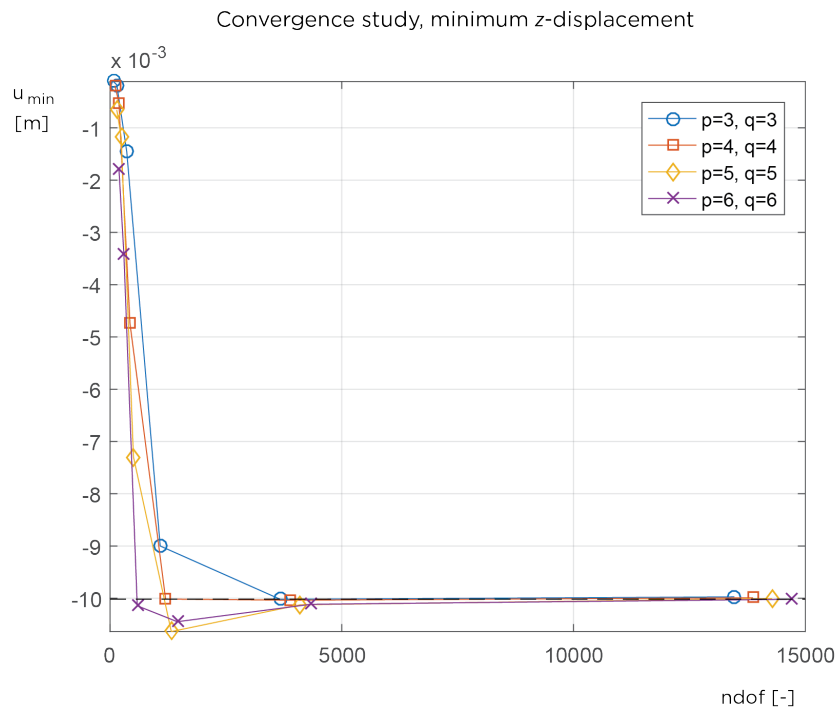


Figure 6.11: Minimum z -displacement convergence for the IGA shell model.

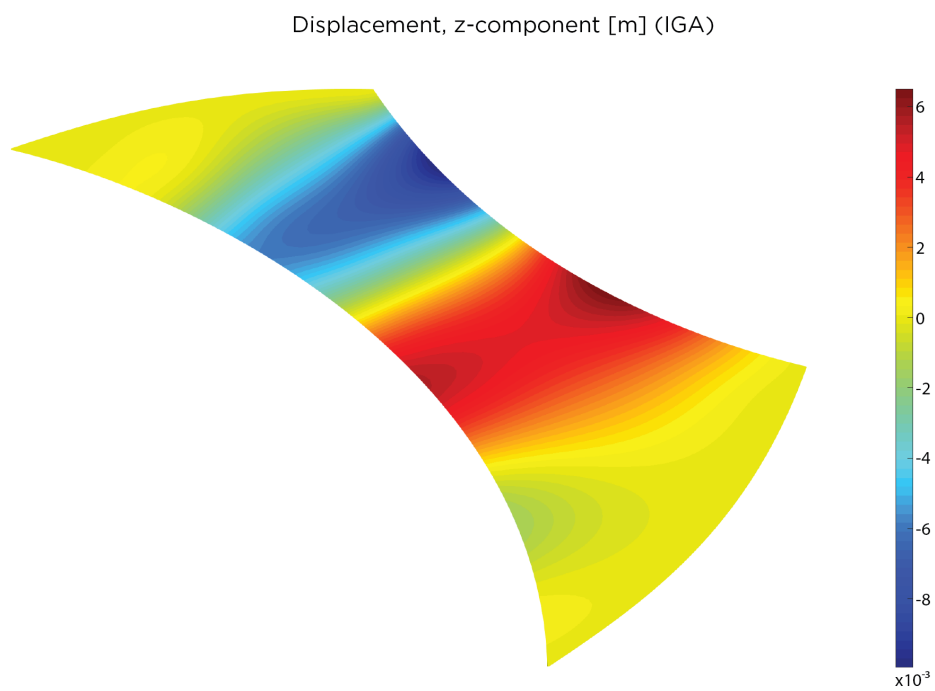


Figure 6.12: Deformation in z -direction obtained with IGA.

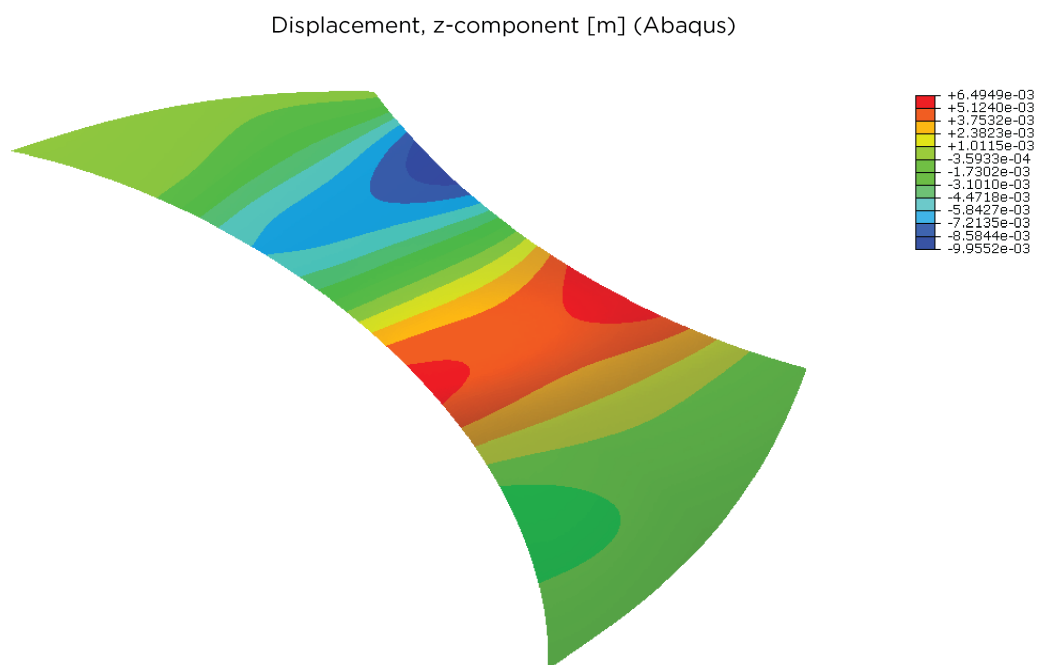


Figure 6.13: Deformation in z -direction obtained with AbaqusTM.

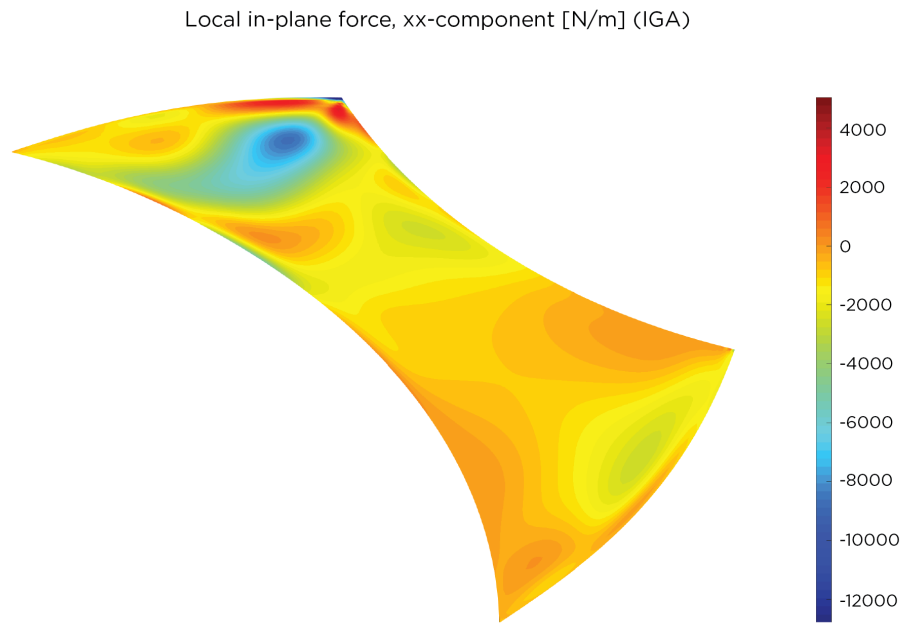


Figure 6.14: Local in-plane force, xx -component. Results from the IGA shell model.

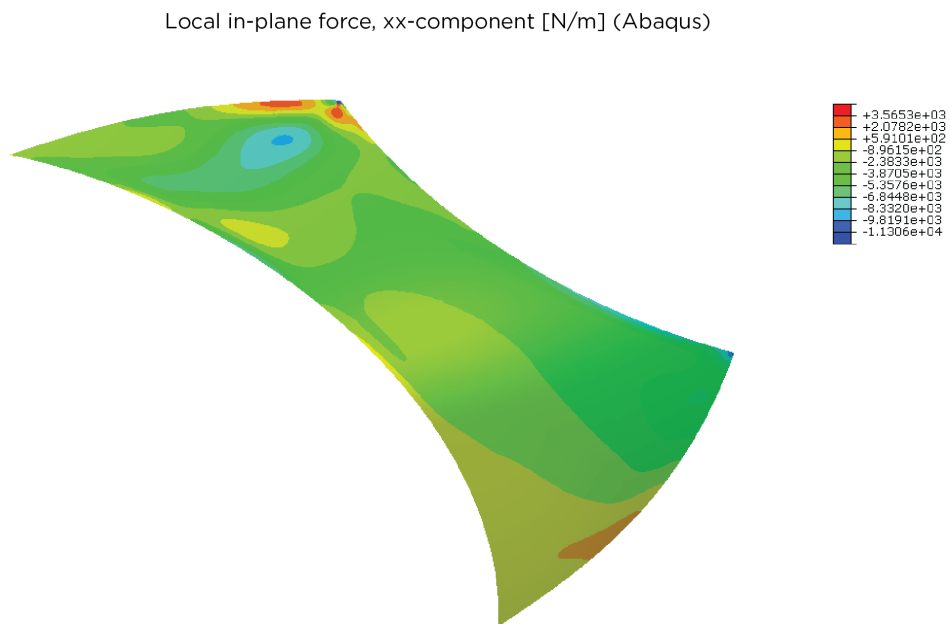


Figure 6.15: Local in-plane force, xx -component. Results from AbaqusTM.

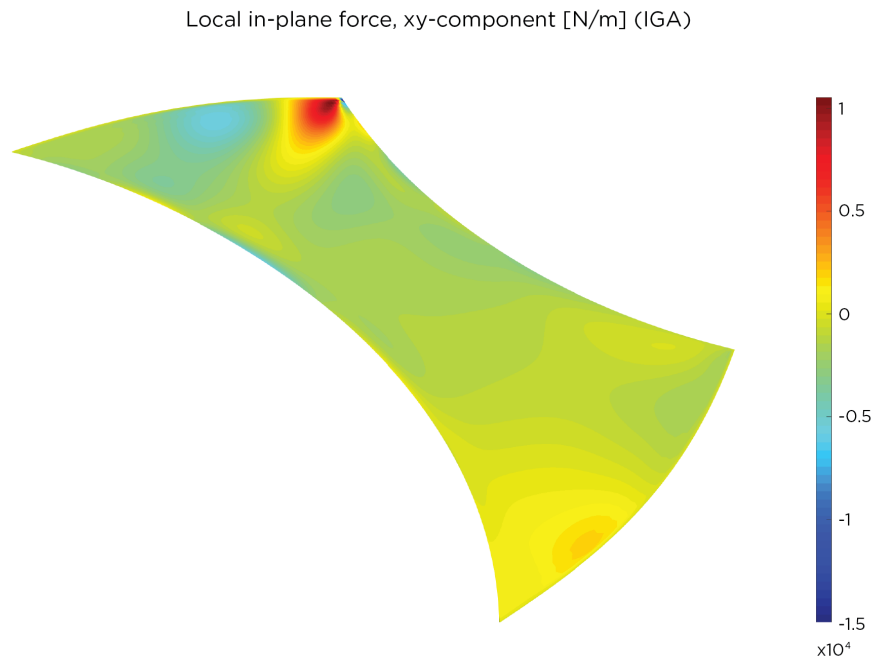


Figure 6.16: Local in-plane force, xy -component. Results from the IGA shell model.

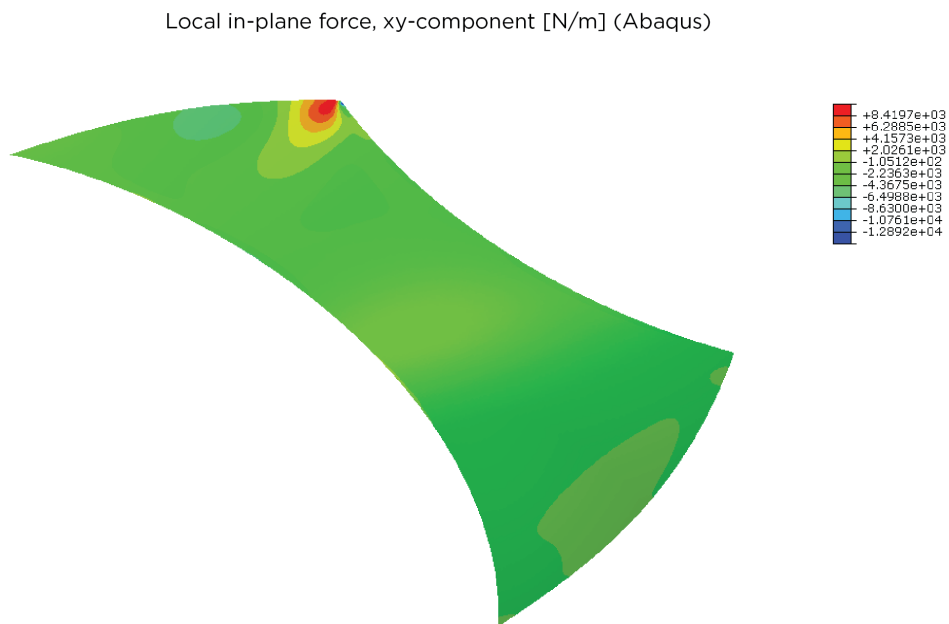


Figure 6.17: Local in-plane force, xy -component. Results from AbaqusTM.

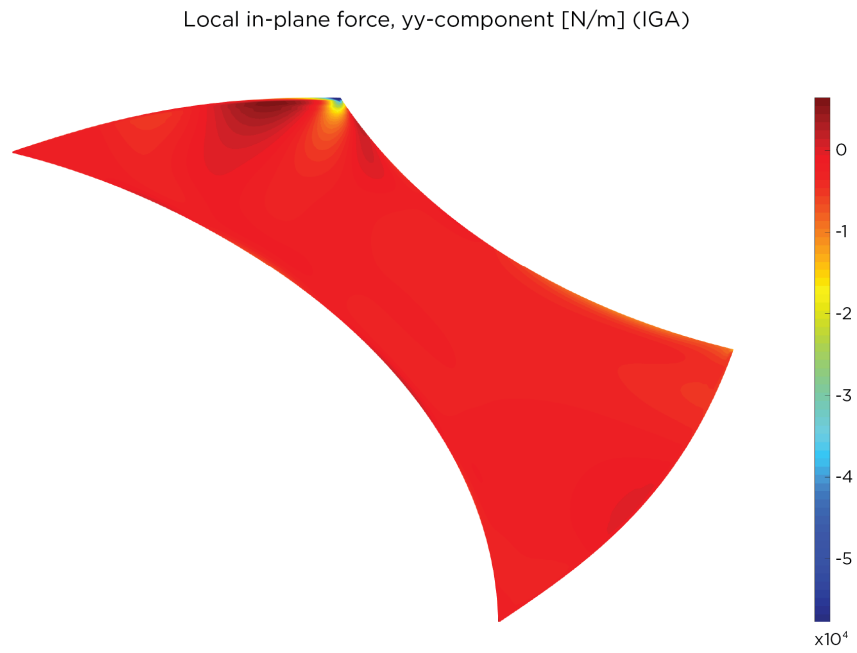


Figure 6.18: Local in-plane force, yy -component. Results from the IGA shell model.

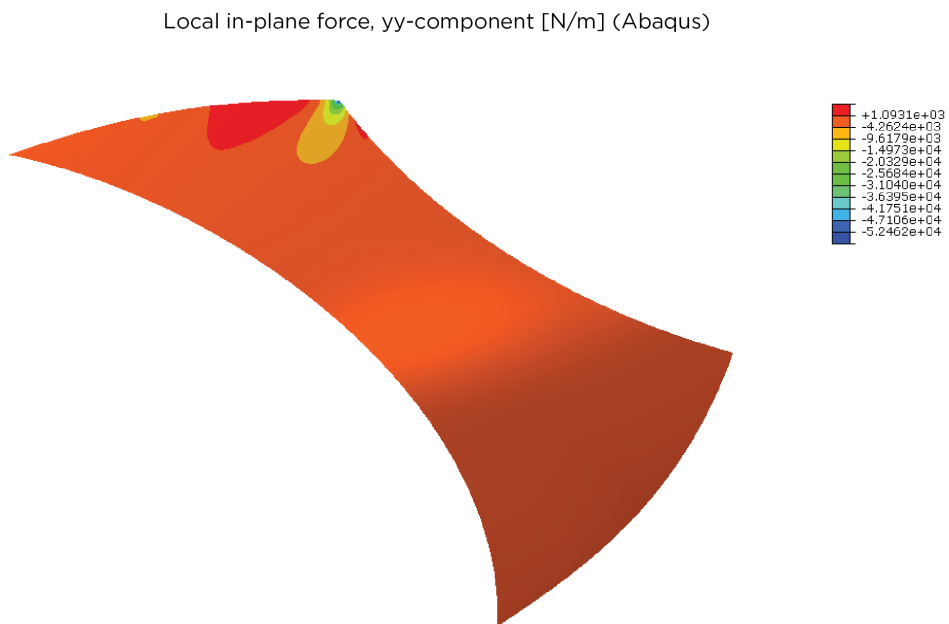


Figure 6.19: Local in-plane force, yy -component. Results from AbaqusTM.

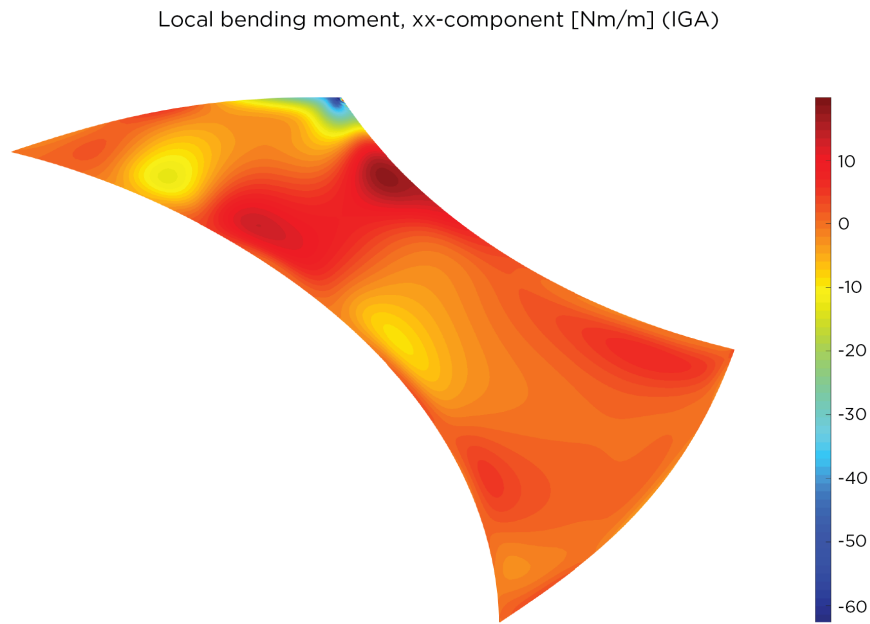


Figure 6.20: Local bending moment, xx -component. Results from the IGA shell model.

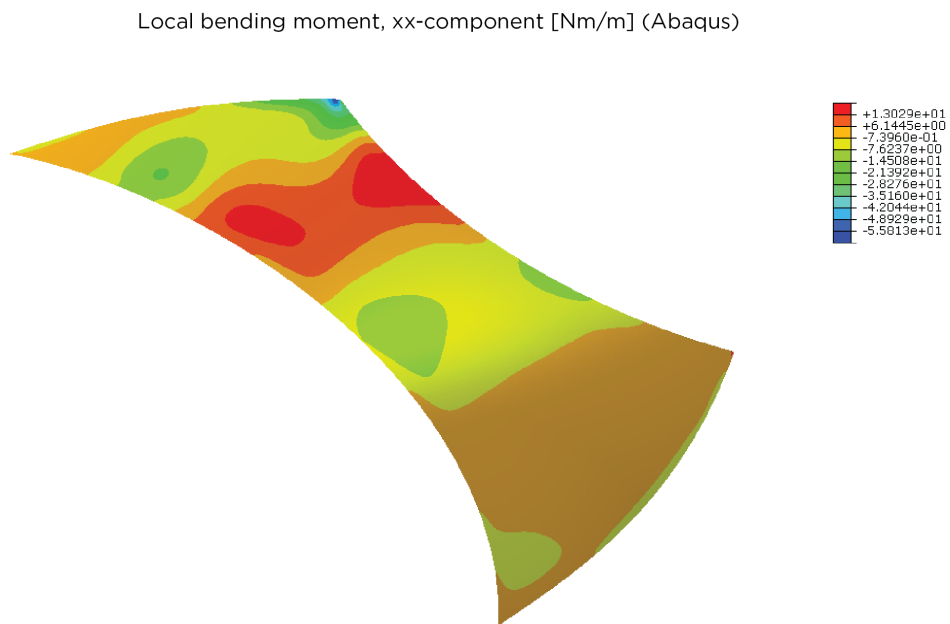


Figure 6.21: Local bending moment, xx -component. Results from AbaqusTM.

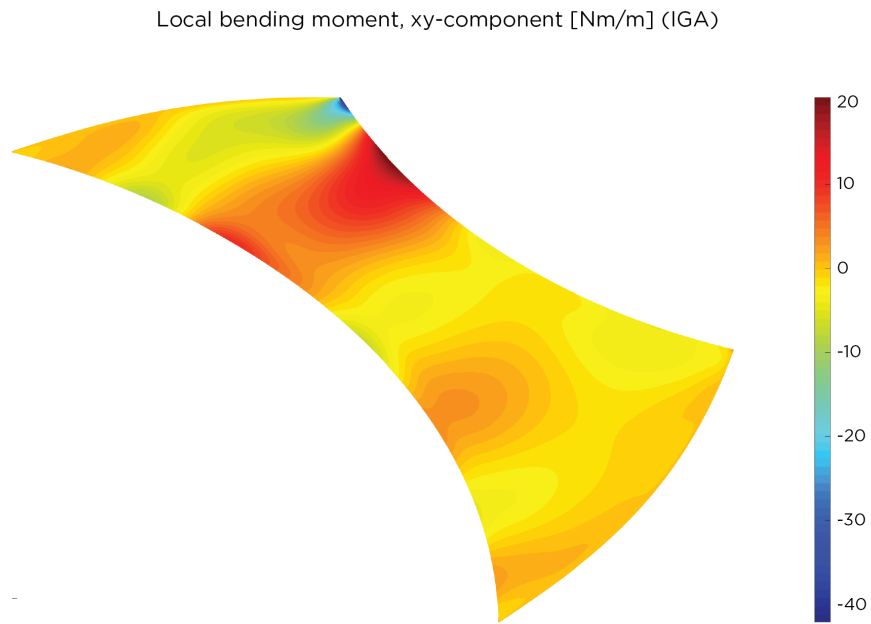


Figure 6.22: Local bending moment, xy -component. Results from the IGA shell model.

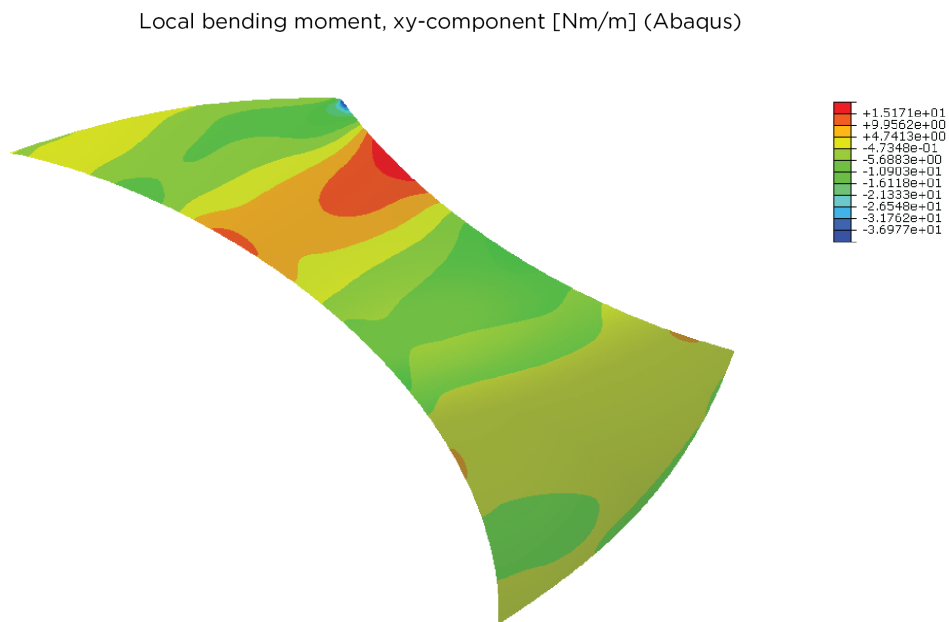


Figure 6.23: Local bending moment, xy -component. Results from AbaqusTM.

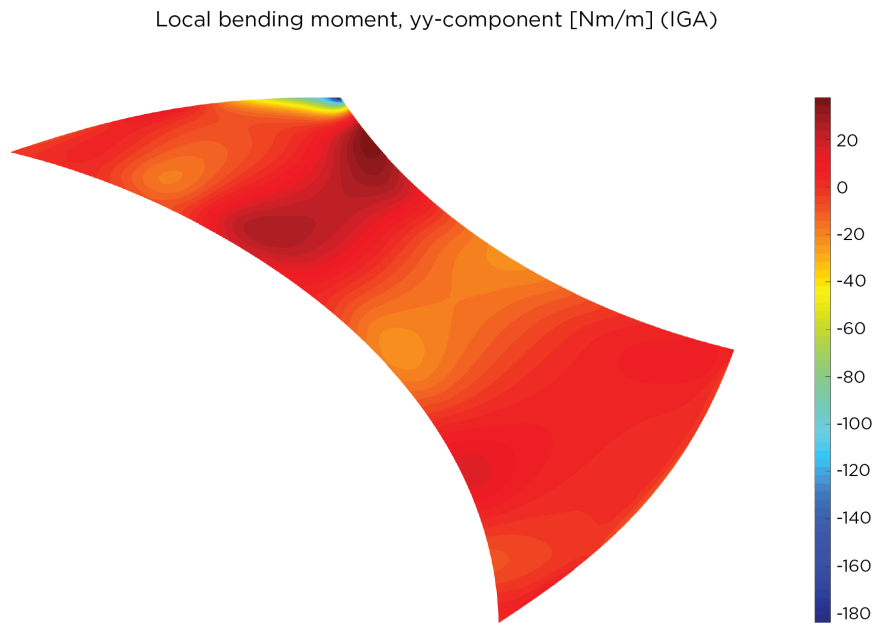


Figure 6.24: Local bending moment, yy -component. Results from the IGA shell model.

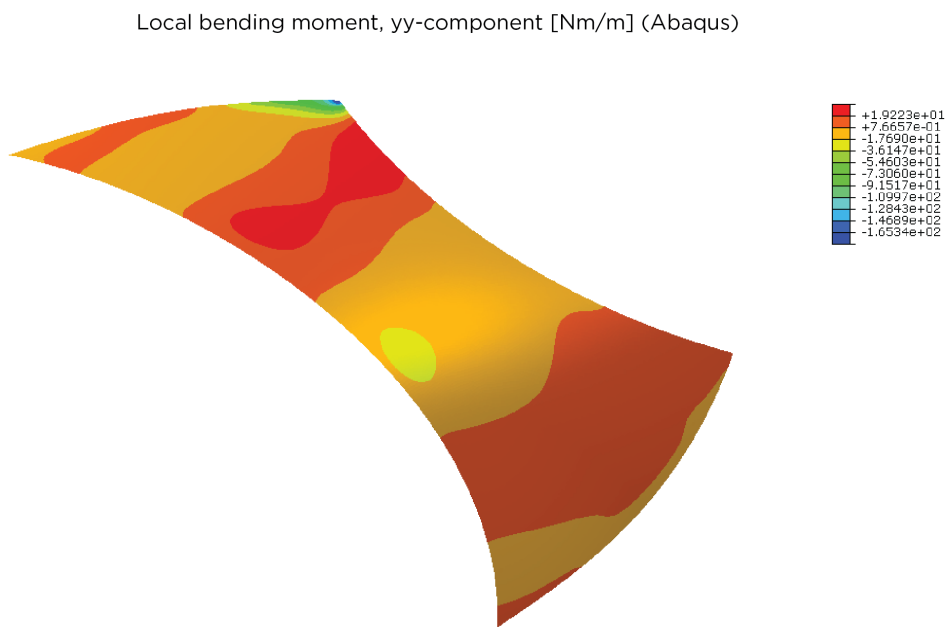


Figure 6.25: Local bending moment, yy -component. Results from AbaqusTM.

6.5 Workflow

There are strong upsides to working with IGA besides more accurate results for fewer degrees of freedom compared to classic FEA. These advantages lie within the workflow that results from utilizing isogeometric analysis.

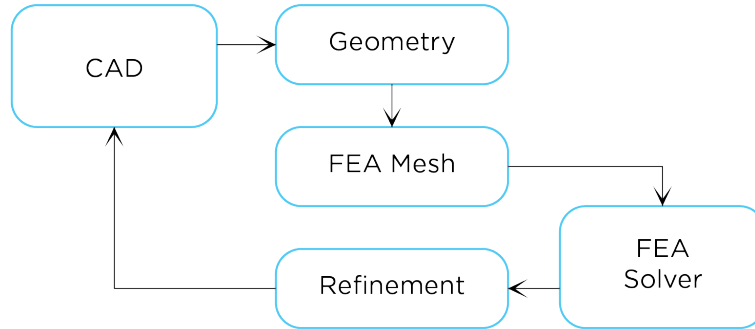


Figure 6.26: Flow of iterative FEA process.

It is well-known within the industry, and also noted during the case study, that meshing for FEA compatibility is tedious. For example, for the analyses run in RobotTM and ETABS[®] it was necessary to return to the CAD interface to generate a finer mesh each time refinement was necessary, see Fig. 6.26. As mentioned previously the meshes for the case study were generated with the Grasshopper[®] component MeshMachine. Achieving a mesh with this component tends to be a time consuming task. Additionally, a mesh quality check is required before sending off the geometry for analysis. Thus, the iterative procedure required to produce an acceptable mesh becomes slow.

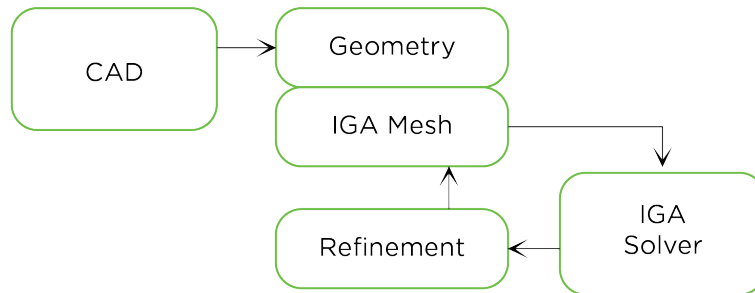


Figure 6.27: Flow of reduced IGA process.

Meantime, IGA meshes reflecting the exact geometry are automatically generated. The geometry and the mesh are inherently connected by sharing the same underlying description. Thus, the meshing step is reduced and the only thing remaining is to define the method for mesh refinement, in this case k -refinement. The iterative procedure is reduced as shown in Fig. 6.27.

Industry reports state that about 20% of analysis time consists of meshing [2]. Huge potential for time-savings can be seen since IGA does not need the meshing step that is essential in the FEA process. According to the same reports 60% of analysis time consists of generating analysis-suitable geometries. Potential for time-savings here was also noted during the case study in that setting up the model for the IGA solver was far more straightforward than for the FEA solver.

7 Conclusions

The question addressed at the start of this project is whether IGA is more time efficient and yields more accurate results than classic FEA on a per-degree-of-freedom basis. To investigate this, models for beams were implemented in 1D and a shell model based on Kirchhoff-Love theory was implemented for analysis in 2D. The models were verified against analytical reference solutions. Furthermore, classic FEA solutions were obtained through self-implemented models as well as commercial software applications. These were then used as basis for comparing the efficiency and accuracy advantages of IGA compared to classic FEA. Finally, a case study was carried out to test the shell analysis model on a built structure, as well as to obtain experience from using IGA in a real project scenario. Based on all of this, some conclusions are drawn.

7.1 Accuracy and efficiency

Results from examples and benchmark tests imply fulfillment of the initial hypothesis. It can be concluded that IGA gives superior results in terms of both accuracy and convergence rate on a per-degree-of-freedom basis. A key factor seems to be the ability to utilize higher continuity over element boundaries compared to classic FEA. However, it is also noted that increased continuity results in computationally expensive analyses. More coupling terms are introduced along with increasing polynomial degree, thus computational time increases. However, it is necessary to investigate the influence of different numerical solution methods before conclusions regarding computational efficiency can be drawn. Since Gauss point integration is suboptimal for isogeometric analysis it does not provide a good basis for comparing efficiency.

While it is difficult to judge computational efficiency, more can be said regarding time savings during the entire design process. Since about 80% of the entire analysis time today is spent on generating FEA meshes and analysis-suitable models the potential for time savings is larger outside of the actual solver. About 20% of those 80% is immediately omitted because geometry and mesh are inherently connected in IGA. How advantageous this is was noticed during the case study. Setting up the analysis model for IGA was straightforward in comparison with the process of exporting geometries and generating meshes for FE analysis. Additionally, different file formats for exporting the geometrical data gave slight differences in the end result obtained with the FE analysis. However, how much these geometrical discrepancies would affect the project in the end remains an open question since errors are introduced further ahead in the process as well, such as during the construction phase.

7.2 Potential for integrated design

Of special interest in this project is integrated structural and architectural design. Architects and engineers have different ways of working. They are involved at different stages of a project and use different software programs in their respective design processes. This severely limits the ability for architectural and structural design to inspire and influence each other. Efforts made in the industry to bridge this gap and enable tighter collaboration over professional boundaries include for example Building Information Modeling (BIM) where each project party contributes with their design information. However, this often happens at the cost of freedom of design as the programs are severely limited in terms of handling complex geometries. Additionally, the process remains time-consuming even with the use of BIM. On the other hand, if the problem is approached by letting analysis move towards the design process there is a true chance of integrating structural engineering with architectural design. This is what isogeometric analysis is about.

Isogeometric analysis enables a common platform for analysis and design and thereby a unified design process. Greater geometric complexity can be described as the geometry description is maintained throughout the

process. Furthermore, interactive design could also be enabled by setting up the model based on parameters rather than static values, so called parametric analysis.

There is potential of using IGA for all parts of the process, but the usability in one design phase stands out. Isogeometric analysis shows great potential in early stage conceptual design. By being able to directly take a design model and analyze it, without time consuming transformation of data, one can early on give indications of feasibility. The structural engineering perspective can influence the project to take on directions where the aspects of design and engineering strengthen each other.

7.3 Suggestions for further work

Based on the findings in this thesis project the following suggestions for further work are concluded:

- Include torsion in the IGA beam model and thus complete a model for 3D beams.
- Include large deformations in the IGA shell model.
- Explore different options for buckling analysis of shells.
- Complement the IGA shell model by implementing Reissner-Mindlin theory.
- Investigate the influence on efficiency in IGA of different numerical solution methods, for example collocation.
- Implement T-spline basis instead of NURBS basis to enable analysis of multiple patch geometries.
- Investigate methods for local and adaptive refinement in IGA.
- Implement models for analysis of contact problems with IGA.

References

- [1] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* **194** (2005), 4135–4195.
- [2] J. A. Cottrell, T. J. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009. ISBN: 978-0-470-74873-2.
- [3] T. J. Hughes. *Isogeometric analysis: Where we are and where we are going*. 2014. URL: <https://www.youtube.com/watch?v=Vw169Ds69y0> (visited on 05/03/2017).
- [4] S. Adriaenssens et al. *Shell Structures for Architecture. Form Finding and Optimization*. Routledge, 2014. ISBN: 978-0-415-84059-0.
- [5] M. I. Design. *Roof structure for L'Oceanografic. Valencia. Spain*. 2015. URL: <http://www.muvprojects.com/en/proyectos/puentes-y-estructuras/cubiertas-oceanografico-valencia/>.
- [6] UweRohwedder. image. 2017. URL: <https://upload.wikimedia.org/wikipedia/commons/4/4b/Alsterschwimmhalle-Hamburg.JPG>.
- [7] C. Faber. *Estructuras de Candela*. Editorial Continental, 1970.
- [8] V. P. Nguyen et al. Isogeometric analysis: an overview and computer implementation aspects. *Mathematics and Computers in Simulation* **117** (2015), 89–116.
- [9] N. S. Ottosen and H. Petersson. *Introduction to the Finite Element Method*. Pearson Education Limited, 1992. ISBN: 0-13-473877-2.
- [10] D. Chapelle and K. Bathe. *The Finite Element Analysis of Shells - Fundamentals*. Springer, 2003. ISBN: 3-540-41339-1.
- [11] J. M. Kiendl. “Isogeometric Analysis and Shape Optimal Design of Shell Structures”. PhD thesis. Technische Universität München, 2011.
- [12] J. Kiendl et al. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering* **198** (2009), 3902–3914.
- [13] H. Casquero et al. Arbitrary-degree T-splines for isogeometric analysis of fully nonlinear Kirchhoff–Love shells. *Computer-Aided Design* **82** (2017), 140–153.
- [14] L. Piegl and W. Tiller. *The NURBS Book*. Second. Springer, 1997. ISBN: 3-540-61545-8.
- [15] R. McNeel and Associates. *What are NURBS?* 2017. URL: <https://www.rhino3d.com/nurbs> (visited on 03/06/2017).
- [16] T. J. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* **199** (2010), 301–313.
- [17] CIMNE. *Invited sessions*. 2017. URL: <http://congress.cimne.com/iga2017/frontal/Invited.asp>.
- [18] V. Alic. *Isogeometric Analysis toolbox for CALFEM*. 2017. URL: <https://github.com/CALFEM/calfem-matlab-iga>.
- [19] A. Cazzani, M. Malagù, and E. Turco. Isogeometric Analysis of Plane-Curved Beams. *Mathematics and Mechanics of Solids* **21(5)** (2016), 562–577.
- [20] C. Lim, C. Wang, and S. Kitipornchai. Timoshenko curved beam bending solutions in terms of Euler-Bernoulli solutions. *Archive of Applied Mechanics* **67** (1997), 179–190.
- [21] R. Bouclier, T. Elguedj, and A. Combescure. Locking free isogeometric formulations of curved thick beams. *Computer Methods in Applied Mechanics and Engineering* **245–246** (2012), 144–162.
- [22] G. Cowper. The shear coefficient in Timoshenko’s beam theory. *Journal of Applied Mechanics* **33** (1966), 335–340.
- [23] B. Zadig. *CALFEM*. 2017. URL: <http://www.byggmek.lth.se/resurser/programvara/calfem/>.
- [24] C. J. Williams. *Shell structures*. 2017. URL: <http://people.bath.ac.uk/absckjkw/LectureNotes/what-is-a-shell.pdf>.
- [25] D. Millán, A. Rosolen, and M. Arroyo. Thin shell analysis from scattered points with maximum-entropy approximants. *International journal for numerical methods in engineering* **85** (2011), 723–751.
- [26] J. Kiendl et al. The bending strip method for isogeometric analysis of Kirchhoff-Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering* **199** (2010), 2403–2416.
- [27] T. Belytschko et al. Stress projection for membrane and shear locking in shell finite elements. *Computer Methods in Applied Mechanics and Engineering* **51** (1985), 221–258.

- [28] COMSOL. *Structural Mechanics Module. User's Guide*. 2012. URL: <https://lost-contact.mit.edu/afs/enea.it/user/e/ebolese/StructuralMechanicsModuleUsersGuide.pdf>.
- [29] R. Bartek et al. "Design and realization of "Crest", a slender bending-active aluminium shell". *Future Visions*. (Aug. 17–20, 2015). International Association for Shell and Spatial Structures (IASS). 2015.
- [30] E. Weisstein. *Developable surface*. 2017. URL: <http://mathworld.wolfram.com/DevelopableSurface.html>.
- [31] J. Kiendl et al. Isogeometric Kirchhoff-Love shell formulations for general hyper-elastic materials. *Computer Methods in Applied Mechanics and Engineering* **291** (2015), 280–303.
- [32] C. Felippa. *MultiFreedom Constraints II*. 2017. URL: <http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/>.

A Second derivative of shape functions w.r.t. physical space coordinates

The second derivative of the NURBS basis functions with regards to a coordinate in physical space is required for some model formulations. The first derivative is stated by Hughes et al. [2] but the second derivative is not as easily accessible in literature. For that reason, the derivation is given here. Initially, it was required for the curved finite element Euler-Bernoulli beam formulation, but it is not specific to that case. The final expression, given by Eq. (A.10) below, is general for the NURBS basis and any physical space coordinate.

It is d^2N/ds^2 that is sought where N denotes the NURBS basis functions and s the physical space coordinate. The coordinate s can be expressed in Cartesian coordinates x and y where each coordinate corresponds to a parameter space coordinate ξ . This can be expressed by:

$$s(\xi) = \begin{bmatrix} x(\xi) \\ y(\xi) \end{bmatrix} . \quad (\text{A.1})$$

Furthermore, the Jacobian of the transformation from parameter to physical space is given by:

$$ds = J d\xi . \quad (\text{A.2})$$

This is further described by Hughes et al. [2]. Additionally, the following expression for the Jacobian is given:

$$J = \sqrt{\left(\frac{dx}{d\xi}\right)^2 + \left(\frac{dy}{d\xi}\right)^2} , \quad (\text{A.3})$$

and the shape functions can be denoted by:

$$N(s) = N(\xi(s)) . \quad (\text{A.4})$$

Thus, dN can be expressed in the following five ways:

$$dN = \frac{dN}{ds} ds = \frac{dN}{ds} J d\xi = \frac{dN}{ds} \frac{ds}{d\xi} d\xi = \frac{dN}{d\xi} d\xi = dN , \quad (\text{A.5})$$

where the expressions after the second and fourth equal signs, i.e. these ones:

$$\frac{dN}{ds} J d\xi = \frac{dN}{d\xi} d\xi , \quad (\text{A.6})$$

gives the following relationship:

$$\frac{dN}{ds} = \frac{1}{J} \frac{dN}{d\xi} . \quad (\text{A.7})$$

Now, the expression for the second derivative of N can be set up. Utilizing Eqs. (A.2) and (A.7) gives:

$$\frac{d^2 N}{ds^2} = \frac{d}{ds} \frac{dN}{ds} = \frac{d}{ds} \left(\frac{1}{J} \frac{dN}{d\xi} \right) = -\frac{1}{J^2} \frac{dJ}{ds} \frac{dN}{d\xi} + \frac{1}{J} \frac{d}{ds} \left(\frac{dN}{d\xi} \right) = -\frac{1}{J^3} \frac{dJ}{d\xi} \frac{dN}{d\xi} + \frac{1}{J^2} \frac{d^2 N}{d\xi^2} . \quad (\text{A.8})$$

There is one unknown in Eq. (A.8) that has not yet been encountered and that is the derivative of the Jacobian with regards to ξ . Utilizing Eq. (A.3) returns:

$$\frac{dJ}{d\xi} = \frac{1}{2} \frac{1}{\sqrt{\frac{\partial x^2}{\partial \xi} + \frac{\partial y^2}{\partial \xi}}} \left(2 \left(\frac{\partial x}{\partial \xi} \right) \frac{\partial^2 x}{\partial \xi^2} + 2 \left(\frac{\partial y}{\partial \xi} \right) \frac{\partial^2 y}{\partial \xi^2} \right) = \frac{1}{J} \left(\frac{\partial x}{\partial \xi} \frac{\partial^2 x}{\partial \xi^2} + \frac{\partial y}{\partial \xi} \frac{\partial^2 y}{\partial \xi^2} \right) . \quad (\text{A.9})$$

Finally, by inserting Eq. (A.9) into (A.8) the expression for the second derivative is given by:

$$\frac{d^2 N}{ds^2} = \frac{1}{J^2} \left[-\frac{\frac{\partial x}{\partial \xi} \frac{\partial^2 x}{\partial \xi^2} + \frac{\partial y}{\partial \xi} \frac{\partial^2 y}{\partial \xi^2}}{J^2} \frac{dN}{d\xi} + \frac{d^2 N}{d\xi^2} \right] , \quad (\text{A.10})$$

thus, all the parts required for implementation are defined.

B Method for eliminating shear and membrane locking in classic FEA

The method for eliminating locking in the classic FEA beam model based on Timoshenko beam theory is given in this appendix. The method utilizes a stable discretization of a mixed problem and the system is solved by static condensation at element level. Each element is assigned three nodes with basis function properties of degree p , as given in Fig. B.1, and continuity C^0 .

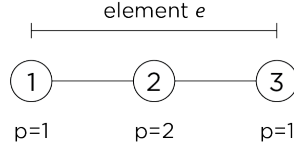


Figure B.1: Node number assignment with corresponding properties for element e .

The basis functions are given by:

$$N_1^e = \frac{1}{2}(1 - \xi) , \quad (\text{B.1})$$

$$N_2^e = (1 - \xi^2) , \quad (\text{B.2})$$

$$N_3^e = \frac{1}{2}(1 + \xi) . \quad (\text{B.3})$$

Let a denote the axial part, s the shear part and m the membrane part of the strain displacement matrix of the Timoshenko beam theory. Then \mathbf{B} can be separated into three parts and for the end nodes that are given by:

$$\mathbf{B}_a^e = \begin{bmatrix} \frac{dN_1^e}{ds} & \frac{N_1^e}{R} & 0 & \frac{dN_3^e}{ds} & \frac{N_3^e}{R} & 0 \end{bmatrix} , \quad (\text{B.4})$$

$$\mathbf{B}_m^e = \begin{bmatrix} 0 & 0 & \frac{dN_1^e}{ds} & 0 & 0 & \frac{dN_3^e}{ds} \end{bmatrix} , \quad (\text{B.5})$$

$$\mathbf{B}_s^e = \begin{bmatrix} -\frac{N_1^e}{R} & \frac{dN_1^e}{ds} & N_1^e & -\frac{N_3^e}{R} & \frac{dN_3^e}{ds} & N_3^e \end{bmatrix} . \quad (\text{B.6})$$

Here R denotes the radius and E , A , I , G and κ are defined as in the report. Thus, the element stiffness contributions from the end nodes can be expressed as:

$$\mathbf{K}_a^e = \int_{\xi} \mathbf{B}_a^{eT} E A \mathbf{B}_a^e J d\xi , \quad (\text{B.7})$$

$$\mathbf{K}_m^e = \int_{\xi} \mathbf{B}_m^{eT} E I \mathbf{B}_m^e J d\xi , \quad (\text{B.8})$$

$$\mathbf{K}_s^e = \int_{\xi} \mathbf{B}_s^{e^T} GA\kappa \mathbf{B}_s^e J d\xi . \quad (\text{B.9})$$

The strain displacement parts for the mid node are denoted by $(\tilde{\cdot})$ and the contributions are defined by:

$$\tilde{\mathbf{M}}_a^e = \int_{\xi} \begin{bmatrix} \frac{dN_2^e}{ds} & \frac{N_2^e}{R} & 0 \end{bmatrix} \begin{bmatrix} EA & 0 & 0 \\ 0 & EI & 0 \\ 0 & 0 & GA\kappa \end{bmatrix} \begin{bmatrix} \frac{dN_2^e}{ds} \\ \frac{N_2^e}{R} \\ 0 \end{bmatrix} J d\xi , \quad (\text{B.10})$$

$$\tilde{\mathbf{M}}_s^e = \int_{\xi} \begin{bmatrix} -\frac{N_2^e}{R} & \frac{dN_2^e}{ds} & 0 \end{bmatrix} \begin{bmatrix} EA & 0 & 0 \\ 0 & EI & 0 \\ 0 & 0 & GA\kappa \end{bmatrix} \begin{bmatrix} -\frac{N_2^e}{R} \\ \frac{dN_2^e}{ds} \\ 0 \end{bmatrix} J d\xi , \quad (\text{B.11})$$

$$\tilde{\mathbf{B}}_a^e = \int_{\xi} [\mathbf{B}_a^e \quad \mathbf{B}_m^e \quad \mathbf{B}_s^e] \begin{bmatrix} EA & 0 & 0 \\ 0 & EI & 0 \\ 0 & 0 & GA\kappa \end{bmatrix} \begin{bmatrix} \frac{dN_2^e}{ds} \\ \frac{N_2^e}{R} \\ 0 \end{bmatrix} J d\xi , \quad (\text{B.12})$$

$$\tilde{\mathbf{B}}_s^e = \int_{\xi} [\mathbf{B}_a^e \quad \mathbf{B}_m^e \quad \mathbf{B}_s^e] \begin{bmatrix} EA & 0 & 0 \\ 0 & EI & 0 \\ 0 & 0 & GA\kappa \end{bmatrix} \begin{bmatrix} -\frac{N_2^e}{R} \\ \frac{dN_2^e}{ds} \\ 0 \end{bmatrix} J d\xi . \quad (\text{B.13})$$

The element stiffness matrix contributions $\tilde{\mathbf{K}}_a^e$ and $\tilde{\mathbf{K}}_s^e$ are then given by static condensation:

$$\tilde{\mathbf{K}}_a^e = \mathbf{K}_a^e - \tilde{\mathbf{B}}_a^e \tilde{\mathbf{M}}_a^{e^{-1}} \tilde{\mathbf{B}}_a^{e^T} , \quad (\text{B.14})$$

$$\tilde{\mathbf{K}}_s^e = \mathbf{K}_s^e - \tilde{\mathbf{B}}_s^e \tilde{\mathbf{M}}_s^{e^{-1}} \tilde{\mathbf{B}}_s^{e^T} , \quad (\text{B.15})$$

resulting in the element stiffness matrix \mathbf{K}^e :

$$\mathbf{K}^e = \tilde{\mathbf{K}}_a^e + \mathbf{K}_m^e + \tilde{\mathbf{K}}_s^e . \quad (\text{B.16})$$

C Change of coordinate system in the beam models

One of the solution fields obtained with both beam models is displacements. These are given in local coordinates, i.e. tangential u and radial v . In general it is more interesting to have the displacement field in Cartesian coordinates x and y . The equations required for the transformation are stated in this appendix. Here the out-of-plane displacement, w in local coordinates and z in Cartesian coordinates are included for generality. However, these components become zero for beam models in 2D.

Firstly, the tangent to the curve at a given evaluation point ξ is defined:

$$\mathbf{t} = \frac{1}{\sqrt{\left(\frac{dN}{d\xi}x(\xi)\right)^2 + \left(\frac{dN}{d\xi}y(\xi)\right)^2 + \left(\frac{dN}{d\xi}z(\xi)\right)^2}} \begin{bmatrix} \frac{dN}{d\xi}x(\xi) & \frac{dN}{d\xi}y(\xi) & \frac{dN}{d\xi}z(\xi) \end{bmatrix}. \quad (\text{C.1})$$

The tangent vector is normalized to unit vector length:

$$\mathbf{e}_t = \frac{\mathbf{t}}{\|\mathbf{t}\|}. \quad (\text{C.2})$$

Secondly, the normal to the curve at ξ is calculated. The unit vector in z -direction, given by $\mathbf{e}_z = [0 \ 0 \ 1]$, points into the 2D xy -plane of the model. Taking the cross product of \mathbf{e}_z and the tangent gives the normal vector at ξ :

$$\mathbf{e}_n = \mathbf{e}_t \times \mathbf{e}_z. \quad (\text{C.3})$$

As was seen in Ch. 3.5 the displacements of a NURBS curve at a given evaluation point are obtained by multiplying the shape functions at the point with the displacement coefficients of the control points active on the same element e :

$$\mathbf{a}_\xi^{uvw} = \mathbf{N}(\xi)a_\xi^e \quad (\text{C.4})$$

Thus, the displacements in Cartesian coordinates x , y and z are given by:

$$\mathbf{a}_\xi^{xyz} = \mathbf{e}_t \mathbf{a}_\xi^{uvw}(1) + \mathbf{e}_n \mathbf{a}_\xi^{uvw}(2) + \mathbf{e}_z \mathbf{a}_\xi^{uvw}(3). \quad (\text{C.5})$$

D Change of coordinate system in the shell model

Solving the shell problem with IGA or classic FEA returns results in terms of displacements. From these stresses and strains can be computed, i.e. other important quantities in structural analysis. The procedure is called stress recovery. For the Kirchhoff-Love shell model presented in this thesis the quantities of interest are in-plane stresses $\boldsymbol{\sigma}$ and strains $\boldsymbol{\varepsilon}$ (note differences in notation compared to the derivations of the shell finite element form). Obtaining meaningful values for these quantities requires a couple of post-processing steps. The trickiest step consists of going from curvilinear to Cartesian coordinates. This appendix covers the equations needed to switch between these coordinate systems.

Firstly, in curvilinear coordinates the stresses and strains are expressed by:

$$\boldsymbol{\sigma} = \sigma^{ij}(\mathbf{a}_i \otimes \mathbf{a}_j), \quad (\text{D.1})$$

$$\boldsymbol{\varepsilon} = \varepsilon_{ij}(\mathbf{a}^i \otimes \mathbf{a}^j). \quad (\text{D.2})$$

Secondly, the Cartesian frame is defined as:

$$\mathbf{e}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|}, \quad (\text{D.3})$$

$$\mathbf{e}_2 = \frac{(\mathbf{a}_2 - (\mathbf{e}_1 \cdot \mathbf{a}_2)\mathbf{e}_1)}{\|\mathbf{a}_2 - (\mathbf{e}_1 \cdot \mathbf{a}_2)\mathbf{e}_1\|}, \quad (\text{D.4})$$

$$\mathbf{e}_3 = \mathbf{a}_3. \quad (\text{D.5})$$

Additionally, the contravariant basis is given by:

$$\mathbf{a}^1 = a^{11}\mathbf{a}_1 + a^{12}\mathbf{a}_2, \quad (\text{D.6})$$

$$\mathbf{a}^2 = a^{21}\mathbf{a}_1 + a^{22}\mathbf{a}_2, \quad (\text{D.7})$$

$$\mathbf{a}^3 = \mathbf{a}_3. \quad (\text{D.8})$$

Here $[a^{ij}] = [a_{ij}]^{-1}$ and $a_{ij} = \mathbf{a}_i \cdot \mathbf{a}_j$.

Thus, the stresses and strains in Cartesian coordinates can be expressed by:

$$\boldsymbol{\sigma} = \bar{\sigma}^{ij}(\mathbf{e}_i \otimes \mathbf{e}_j), \quad (\text{D.9})$$

$$\boldsymbol{\varepsilon} = \bar{\varepsilon}_{ij}(\mathbf{e}^i \otimes \mathbf{e}^j). \quad (\text{D.10})$$

The symbol $\bar{\cdot}$ denotes Cartesian components. Finally, by utilizing tensor properties the following expressions for Cartesian stress and strain components are given:

$$\bar{\sigma}^{kl} = \mathbf{e}^k \sigma^{ij}(\mathbf{a}_i \otimes \mathbf{a}_j) \cdot \mathbf{e}^l = \sigma^{ij}(\mathbf{e}^k \cdot \mathbf{a}_i)(\mathbf{a}_j \cdot \mathbf{e}^l) \quad (\text{D.11})$$

$$\bar{\varepsilon}_{kl} = \mathbf{e}_k \varepsilon_{ij}(\mathbf{a}^i \otimes \mathbf{a}^j) \cdot \mathbf{e}_l = \varepsilon_{ij}(\mathbf{e}_k \cdot \mathbf{a}^i)(\mathbf{a}^j \cdot \mathbf{e}_l) \quad (\text{D.12})$$

E Enforcing fixed boundary and symmetry conditions in IGA

Fixed boundary and symmetry conditions can be enforced in isogeometric analysis by procedures originally developed for classic FEA. These include for example the Lagrange multiplier and penalty methods. Here one approach to the Lagrange multiplier method is restated. The procedure and theory is further described in the course literature for a finite element analysis course at the University of Colorado at Boulder [32].

On symmetry boundaries the objective is to restrict certain movements to be the same on both sides of the boundary. These can be translations and/or rotations. To restrict rotations in IGA degrees of freedom on multiple rows that are parallel to the symmetry edge must be addressed. Start by imagining that two degrees of freedom are connected by a rigid link, for example $a_2 = a_5$, and thus must move in the same manner. This constraint can be replaced with a reaction force pair $(-\lambda, +\lambda)$ called the restraint forces.

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & \cdots & K_{1n} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & \cdots & K_{2n} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & \cdots & K_{3n} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & \cdots & K_{4n} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & \cdots & K_{5n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ K_{n1} & K_{n2} & K_{n3} & K_{n4} & K_{n5} & \cdots & K_{nn} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 - \lambda \\ f_3 \\ f_4 \\ f_5 + \lambda \\ \vdots \\ f_n \end{bmatrix}. \quad (\text{E.1})$$

The variable λ is the so called Lagrange multiplier. Since λ is unknown it is better to express Eq. (E.1) by:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & \cdots & K_{1n} & 0 \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & \cdots & K_{2n} & 1 \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & \cdots & K_{3n} & 0 \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & \cdots & K_{4n} & 0 \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & \cdots & K_{5n} & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{n1} & K_{n2} & K_{n3} & K_{n4} & K_{n5} & \cdots & K_{nn} & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ \vdots \\ a_n \\ \lambda \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ \vdots \\ f_n \end{bmatrix}. \quad (\text{E.2})$$

To create a determinate system an additional row is also appended:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & \cdots & K_{1n} & 0 \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & \cdots & K_{2n} & 1 \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & \cdots & K_{3n} & 0 \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & \cdots & K_{4n} & 0 \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & \cdots & K_{5n} & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{n1} & K_{n2} & K_{n3} & K_{n4} & K_{n5} & \cdots & K_{nn} & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & \cdots & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ \vdots \\ a_n \\ \lambda \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ \vdots \\ f_n \\ 0 \end{bmatrix}. \quad (\text{E.3})$$

Solving this system, i.e. Eq. (E.3), instead of the original system enforces the desired movement of a_2 and a_5 .

For multiple freedom constraints the procedure is repeated and more rows and columns are added together with multiple Lagrange multipliers. The multiplier-augmented form, i.e. the system expressed as in Eq. (E.3), is compactly expressed as:

$$\begin{bmatrix} \mathbf{K} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} . \quad (\text{E.4})$$

For the pinched hemisphere shell benchmark problem the "rigid link" is applied by adding 1/10 of the maximum stiffness to affected degrees of freedom. These include the translational degrees of freedom of the control points on the symmetry boundary and adjacent to it.