Czech Technical University in Prague
Department of Mechanics
Prof. Ing. Jiří Máca, CSc.

RWTH Aachen University
Chair of Structural Analysis and Dynamics
Univ.-Prof. Dr.-Ing. habil. S. Klinkel

MASTER THESIS

Development and implementation of an isogeometric scaled boundary shell formulation

presented by

Mathias Reichle, B.Sc.

Matr.-Nr.: 345942

Supervisors:

Dr.-Ing. Markus Klassen

Prof. Ing. Jan Zeman Ph.D.

Aachen, May 5, 2021

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Reichle Mathias Ferdinand Maria** Personal ID number: **490473**

Faculty / Institute: **Faculty of Civil Engineering**

Department / Institute: **Department of Mechanics**

Study program: **Civil Engineering**

Branch of study: **Building Structures**

II. Master's thesis details

Master's thesis title in English:

Development and implementation of an isogeometric scaled boundary shell formulation

Master's thesis title in Czech:

Development and implementation of an isogeometric scaled boundary shell formulation

Guidelines:

Bibliography / sources:

Name and workplace of master's thesis supervisor:

Markus Klassen, Dr.Ing., RWTH Aachen University

Name and workplace of second master's thesis supervisor or consultant:

prof. Ing. Jan Zeman, Ph.D., Department of Mechanics, FCE

Date of master's thesis assignment: **19.11.2020** Deadline for master's thesis submission: **05.05.2021**

Assignment valid until: _____

Jan Zeman (v.z.)

Markus Klassen, Dr.Ing.
Supervisor's signature

Head of department's signature

prof. Ing. Jiří Máca, CSc.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

05.05.2021
Date of assignment receipt

M. Reichle
Student's signature

Abstract

In the field of numerical structural analysis, shell formulations are generally proposed to model the mechanical response of thin curved structures. Typical approaches consist in the numerical approximation of the surface of the structure in order to develop the element stiffness matrix that is used in the framework of the finite element method (FEM). In this context, the thickness direction of the shell is included in the derivation of the element stiffness matrix which is then associated to translational and rotational degrees of freedom.

In recent years, the scaled boundary method has been proposed to model thin structures such as plates and shells. The main idea of this approach consists in a solid shell formulation with scale separation. The in-plane direction is approximated in a classical sense by shape functions, but for the thickness direction, the analytical solution is taken into account. By these means, the nodal degrees of freedom are given by the displacements of the top and bottom surface of the shell.

The objective of the present thesis is to develop and implement a scaled boundary shell formulation into the framework of isogeometric analysis (IGA). This framework offers the advantage of an exact approximation of a shell structure, since it is based on the NURBS functions that are used for the geometrical description. Furthermore, a higher continuity of the solution field is given naturally by means of these functions.

The initial tasks of this project consist in the familiarization with the scaled boundary shell formulation as well as with IGA. Afterwards the element formulation needs to be derived and implemented into the IGA framework. To conclude the work, standard shell element benchmarks such as the pinched cylinder are performed and documented.

Kurzfassung

Im Bereich der numerischen Strukturanalyse werden im Allgemeinen Schalenformulierungen verwendet, um das mechanische Verhalten von dünnen gekrümmten Strukturen zu modellieren. Typische Ansätze bestehen in der numerischen Approximation der Oberfläche der Struktur, um die Elementsteifigkeitsmatrix zu entwickeln, die im Rahmen der Finite-Elemente-Methode (FEM) verwendet wird. In diesem Zusammenhang wird die Dickenrichtung der Schale in die Ableitung der Elementsteifigkeitsmatrix einbezogen, die mit translatorischen und rotatorischen Freiheitsgraden verknüpft ist.

In den letzten Jahren wurde die Scaled Boundary Methode entwickelt, um dünne Strukturen wie Platten und Schalen zu modellieren. Die Hauptidee dieses Ansatzes besteht in einer festen Schalenformulierung mit Skalentrennung. Die Richtung in der Ebene wird im klassischen Sinne durch Ansatzfunktionen approximiert, aber für die Richtung der Dicke wird die analytische Lösung berücksichtigt. Auf diese Weise sind die Knotenfreiheitsgrade durch die Verschiebungen der Ober- und Unterseite der Schale gegeben.

Das Ziel der vorliegenden Arbeit ist die Entwicklung und Implementierung einer Scaled Boundary Schalenformulierung im Rahmen der isogeometrischen Analyse (IGA). Dieser Rahmen bietet den Vorteil einer exakten Approximation einer Schalenstruktur, da er auf den NURBS-Funktionen basiert, die für die geometrische Beschreibung verwendet werden. Weiterhin ist durch diese Funktionen inhärent eine höhere Stetigkeit des Lösungsfeldes gegeben.

Die ersten Aufgaben dieses Projektes bestehen in der Einarbeitung in die Scaled Boundary Schalenformulierung sowie in die IGA. Danach muss die Elementformulierung abgeleitet und in das IGA-Framework implementiert werden. Zum Abschluss der Arbeiten sind Standard-Schalenelement-Benchmarks wie z.B. der eingespannte Zylinder durchzuführen.

Acknowledgement

Throughout the whole thesis I have received a great deal of support and assistance, for which I would like to express my gratitude.

I would first like to thank my supervisor Dr.-Ing. Markus Klassen, who supervised me from the side of RWTH Aachen and supported me at any time with his experience and good advice. Due to his uncomplicated and prescient guidance, the meetings were always beneficial and targeted.

I would also like to thank Prof. Jan Zeman Ph.D. for supervising me from the side of CTU in Prague and the uncomplicated communication and helpful support during the time of my thesis. He did not only supervise me, but also provided me with the opportunity to participate in a research project during my year abroad in Prague, which gave me an interesting insight into research at the university.

Furthermore, I would like to thank Jianghuai Li Ph.D. for helping me with questions about SBFEM and always giving very helpful hints in implementing the formulation. The support greatly enriched the work.

In addition, I would like to thank the whole staff of the Chair of Structural Analysis and Dynamics, where I was able to work during my time in the master's program and gained a lot of knowledge and experience in many areas.

Finally, I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of writing this thesis.

Mathias Reichle

CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations

CAD	Computer-Aided Design
DOF	Degree Of Freedom
EAP	Electroactive Polymers
FDM	Finite Difference Method
FEA	Finite Element Analysis
FEM	Finite Element Method
IGA	Isogeometric Analysis
NURBS	Non-Uniform Rational B-Splines
SBFEM	Scaled Boundary Finite Element Method
SBIGA	Isogeometric Scaled Boundary Analysis
SBM	Scaled Boundary Method

Nomenclature

Latin Symbols

$\mathbf{B}_i$	Control points of a curve
$\mathbf{B}_{i,j}$	Control net of a surface
$\mathbb{C}$	Elasticity matrix
E	Young's modulus
$\mathbf{E}_i^{00}$ - $\mathbf{E}_{02}^{2}$	Coefficient matrix
$\mathbf{F}$	Load vector
h	Thickness
i	Knot index
$\hat{\mathbf{J}}_s$	Jacobian matrix
j	Knot index
$\mathbf{K}$	Stiffness matrix
k	Multiplicity of a knot
L	Length
l_{au}	Element length in in parametric space
m	Number of basis functions in η
m_k	Number of knots on knot vector
n	Number of basis functions in ξ
$\mathbf{n}$	Normal vector
n_{gp}	Number of Gauss points per parametric direction
p	Polynomial order of NURBS and B-splines
q	Polynomial order of NURBS and B-splines
R	Radius
$\mathbf{r}$	Position vector to point A
$\hat{\mathbf{r}}$	Position vector of a point in the shell
$\mathbf{T}$	Transformation matrix
$\mathbf{u}$	displacement vector
w	Gauss quadrature weight

Greek Symbols

γ	Axis of the local coordinate system
ϵ	Linearized strain vector
H	Knot vector in η-direction
η	Axis of the local coordinate system
θ	Rotational degrees of freedom
κ	Curvature
λ	Lamé constant
μ	Lamé constant
ν	Poisson's ratio
Ξ	Knot vector in ξ-direction
ξ	Axis of the local coordinate system
σ	Cauchy stress tensor

Functions

$\lceil \cdot \rceil$	Ceiling function
$N_{i,p}(\xi)$	B-spline basis function in ξ-direction
$M_{j,q}(\eta)$	B-spline basis function in η-direction
$\mathbf{C}(\xi)$	NURBS or B-spline curve
$\mathbf{S}(\xi, \eta)$	NURBS or B-spline surface
$R_i^p(\xi)$	NURBS curve basis function
$R_{i,j}^{p,q}(\xi, \eta)$	NURBS surface basis function

Operators

$\mathbf{L}$	Differential operator

1. Introduction

1.1. Motivation

Since time immemorial, shell structures have been designed and built for special architectural buildings. Buildings such as the Olympic Center in Munich or the Pantheon in Rome have a very special character. However, these buildings not only place special architectural demands on the builders, but also present challenges for structural design. Over time, the calculation of shells has changed a lot. While the first dome structures still depended heavily on the experience of the master builders, today's modern numerical methods can be used to precisely determine the stress distributions and forces. With the increasing use of computers, more and more complex shells can be calculated with more and more accurate methods.

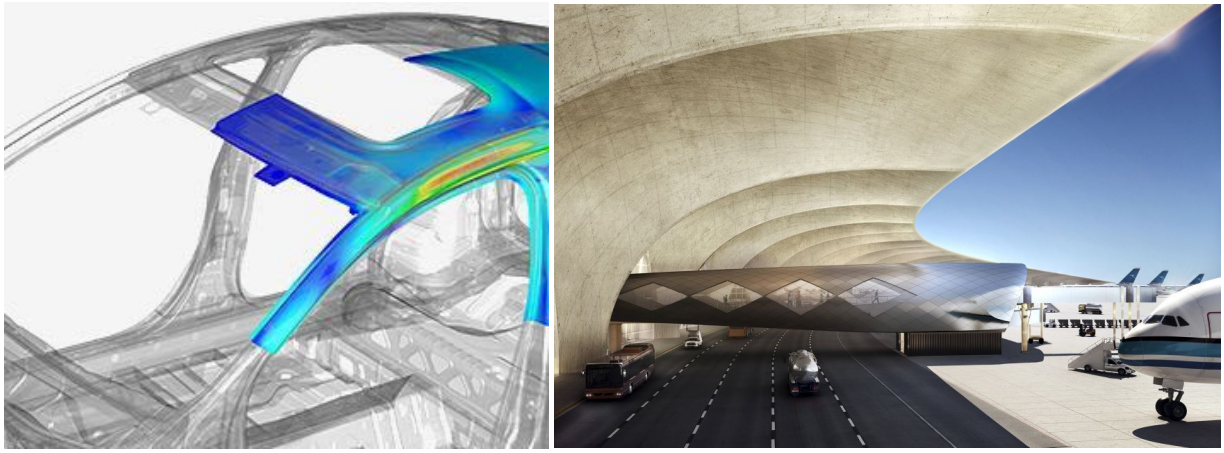


Figure 1.1: Examples of shell structures in engineering problems for a car body [24] (left) and the roof of Terminal 2 at the Kuwait International Airport [6] (right).

In particular, numerical methods, which require a high computational effort, have proven to be very useful due to the development of powerful computers. Nowadays shell structures are not only considered for civil engineering applications but have a wide range of usage. Thus, car bodies, hulls of satellites and airplanes or packaging are analyzed as shells for example.

One method to analyze complicated geometries is called isogeometric analysis (IGA) and is characterized by the fact that the geometric description of a shell is represented exactly at the nodes. In this thesis, the isogeometric analysis is combined with the scaled boundary method, a technique to discretize only the boundary of a body and extrapolate the solution to the interior by a scaling parameter. Thereby, especially computational time can be saved, besides other advantages such as increasing accuracy.

1.2. State of art

Since the advance of computational power, shells have been increasingly calculated using numerical methods, and the FEM has proven to be very powerful. To represent the geometry and the solution field Lagrangian polynomials are employed as basis functions. These functions are normally built up on a mesh, which needs to be generated from a CAD model developed from a design process. This generation of the mesh often causes that the geometry gets changed especially for curved entities. Furthermore, the continuity is of low order at element boundaries which influences the quality of the solution. To overcome this gap between the design process

and the analysis the (IGA) was developed. Basic concept is to use the same basis functions, which are called NURBS, for the design as for the analysis, which is applicable for many kinds of body types.

So far, IGA was applied to several shell formulations. The main theories are the Kirchhoff-Love theory and the Reissner-Mindlin theory. The Kirchhoff-Love shell is rotation-free and only valid for thin shells since the transverse shear deformations are neglected. The shell formulation requires a continuity also on the derivative of the basis functions (C^1-continuity), which is hardly possible to reach for finite elements, but one of the features of NURBS. Therefore, Kiendl et al. [11] developed a NURBS-based Kirchhoff-Love shell.

The Reissner-Mindlin shell plays the dominant role when it comes to finite element. In contrast to the Kirchhoff-Love shell, the transverse shear deformations are considered and thus, the shell is also applicable for thick shells. This formulation was also developed as a NURBS-based shell. Unfortunately, the two shell formulations suffer from locking for thin structures for both approaches, IGA and FEM, especially for low order basis functions.

Therefore, Li et al. [14] developed a shell based on the Scaled Boundary Finite Element Method (SBFEM), which is a solid shell. This method provides an analytical solution in the thickness direction of the shell, for which yields an additional analysis along the thickness. This work is concerned with the development and implementation of an Isogeometric Scaled Boundary shell formulation (SBIGA), which combines the isogeometric analysis with the scaled boundary method for shells.

1.3. Organization of this Work

The thesis consists of six chapters and the appendices. The structure is as follows:

At first in Chapter 2 nowadays problems of the interaction of computer aided design (CAD) and computational methods are described and the advantages of a NURBS-based description of the geometry are pointed out. All the necessary mathematical background of B-splines and NURBS is provided and explained. Finally, a framework for shells using isogeometric analysis is presented, which uses NURBS as basis functions and reduces the gap between design and calculation.

In Chapter 3 an element description using the scaled boundary method for shells with an analytical approach in thickness direction is presented and the crucial points to couple this formulation with the isogeomentric analysis are pointed out. In particular, the increased difficulty of the formulation due to curved surfaces and the associated changes for the scaled boundary transformation of shells are addressed.

Chapter 4 shows the numerical implementation of the isogeometric scaled boundary formulation. At first, a general architecture of a numerical simulation is explained and separated in its features. The key functions and processes are explained and the important parts of the code are visualized.

The benchmarking is documented in Chapter 5. Four examples are presented as they are the simply supported square plate, the pinched cylinder, the Scordelis-Lo roof and the pinched hemisphere with hole. All the models are evaluated conducting an order elevation and a mesh refinement and the accuracy of the solution is critically evaluated.

The summary of the results of this work is given in Chapter 6. Furthermore and outlook for future investigations is presented. Further information is given in two appendices to present additional technical details related to the main part of the thesis.

2. Isogeometric analysis

Different methods are provided to describe the interpolation of the geometry and the solution field of an object in the Euclidean space. In this thesis Non-Uniform Rational B-Splines (NURBS) are used as shape functions, which is a quite powerful tool in the framework of isogeometric analysis (IGA) for structural analysis. The method was first introduced by Hughes et al. in 2005 in [9]. The main motivation to use NURBS as basis functions in numerical analysis is that the geometry describing functions of the design process can be used directly in the simulation. In the model generation and simulation analysis with the usual Finite Element Analysis (FEA), especially the model generation takes a very high portion of the time, since an entirely new model has to be created for the simulation, which in turn has to be generated with different meshes.

Figure 2.1 shows all important steps for a simulation with numerical methods and the relative portions of the total time of the process. The data was published in the paper [5] in 2005 and were collected by Sandia National Laboratories. It can be seen that almost 70% of the time was spent on building the model (highlighted in yellow). If, on the other hand, NURBS rather than Lagrange polynomials are used for the basis functions, this ratio can be significantly reduced, since the functions are already stored in the CAD model. Thus, the process of model generation cannot be eliminated completely, but the necessary time is reduced considerably.

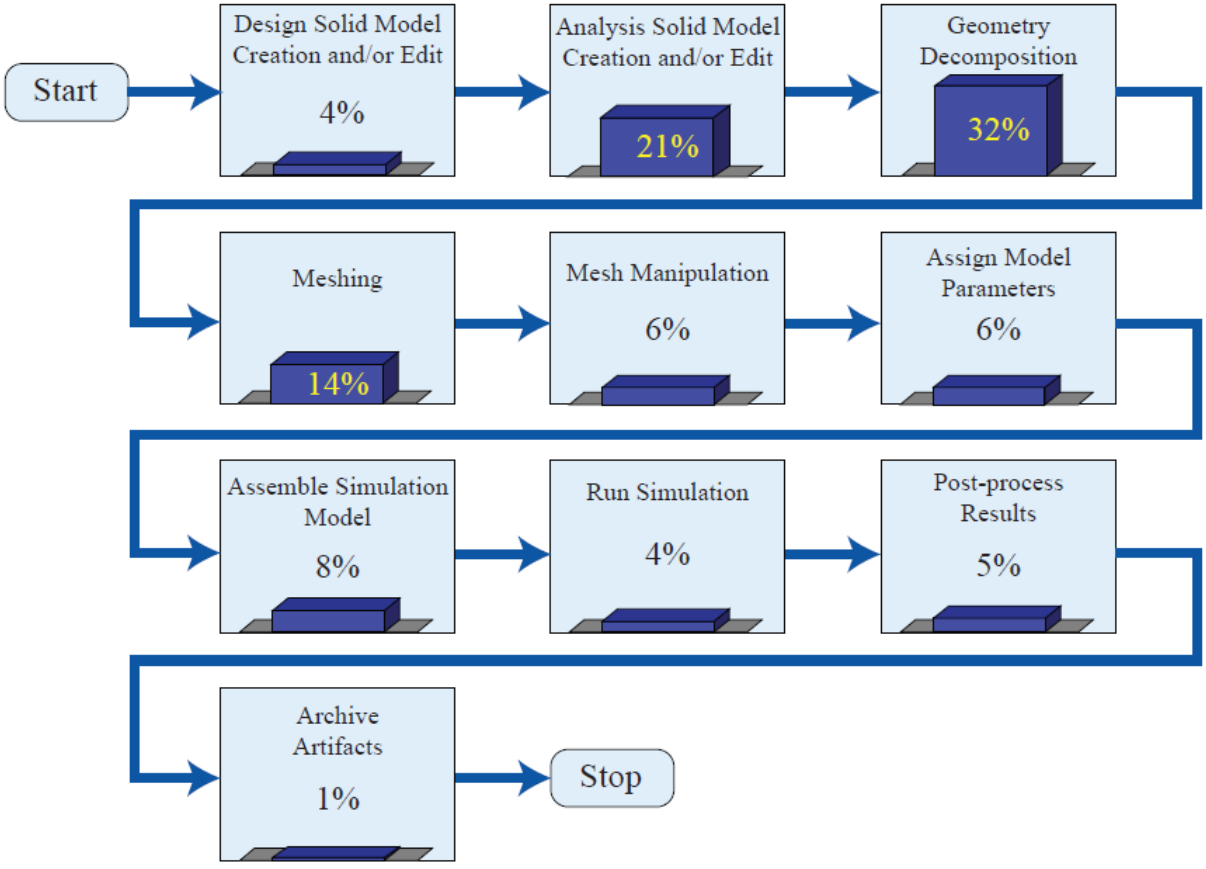


Figure 2.1: Exemplary estimation of relative time consumption of various steps in a simulation process at Sandia National Laboratories [5].

But not only the model generation becomes less time consuming - also in other areas the IGA

offers advantages, because it combines features of the finite element analysis and meshless methods, namely:

1. It takes the geometry more into account by evaluation of the NURBS and control points instead of Lagrange polynomials and nodes. No matter how rough the mesh is discretized, the geometric description is always exact.
2. By saving the geometry within the NURBS basis functions, the mesh refinement is simplified once the initial mesh is constructed.
3. The mesh refinement can be used to increase the continuity of the model which provides better stability of the calculation for specific problems.

Therefore, in the following sections the fundamentals of describing geometries with B-splines and NURBS are explained and their applications, properties, advantages and disadvantages are shown. After describing the basis functions, curves and surfaces, the framework for numerical analysis with NURBS (IGA) is introduced as a conclusion of this chapter.

2.1. Geometric description using B-splines and NURBS

Since NURBS consist of B-splines, it is important to explain their concept first. For this purpose, a step-by-step explanation of the geometric description of entities is given. At first, the B-spline basis functions are presented which can then be used to generate B-spline curves and surfaces. After having a look on the specific properties regarding their continuity, NURBS are introduced and so their derivatives. With this in hand, all the theory that is necessary to know for the geometric description in IGA is provided.

2.1.1. Fundamentals of B-splines

At first, a knot vector is introduced, which is a vector in one dimension and contains a set of non-decreasing coordinates in parametric space and is presented as $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$, where $\xi_i \in \mathbb{R}$ is the i-th knot, i is the knot index with $i = 1, 2, ..., n + p + 1$, n is the number of basis functions that compose the B-spline and p is the polynomial order which is usually referred to as degree in the computational geometry literature. These knots can be considered as the division points that subdivide the interval $\{\xi_1, , ..., \xi_{n+p+1}\}$ into knot spans. Each B-spline basis function is supposed to have its own domain on the interval. If the distance between the knots is not changing in the parametric space, which means they are equally spaced, it is called *uniform*. If they are unequally spaced, the knot vector is called *non-uniform*. This is of special interest when it comes to NURBS, see Section 2.1.5. Furthermore, if a knot is located more than once on the same coordinate in parametric space, it is called a *repeated* knot. These repeated knots determine the continuity of the splines and are discussed in Section 2.1.3. The most commonly used *open* knot vectors in structural analysis have its first and last knots $p + 1$ times repeated. Thus, in one dimension the B-spline curves are interpolatory at the ends of the parametric space interval and at the corners of patches in multiple dimensions. However, it is important to notice that B-splines not necessarily interpolate at interior knots, which is a big difference between knots and so called nodes which are used in the finite element analysis.

The B-spline basis functions are usually determined recursively by the *Cox-de Boor recursive formula*. It starts with piecewise constant basis functions ($p = 0$) and defined as

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}. \tag{2.1}$$

For higher polynomial orders of $p = 1, 2, 3, ..$ the basis functions are defined as

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{2.2}$$

With the knot vector and the recursive formula in hand, the B-spline basis functions can be constructed. Therefore, the knot vector defines the polynomial order such as the start and end points of the basis function.

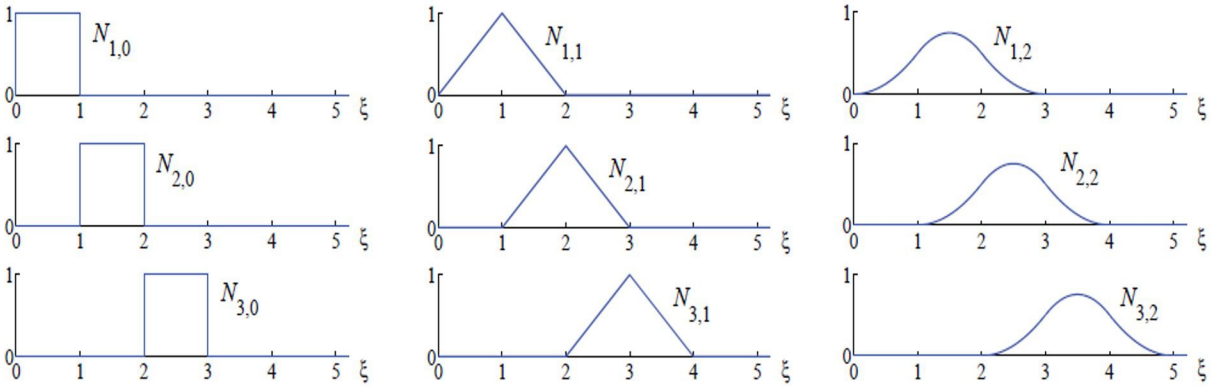


Figure 2.2: B-spline basis functions of orders $p = 0$, $p = 1$ and $p = 2$, defined by open knot vector $\Xi = [0, 1, 2, 3, 4, ...]$ [9].

Consecutively, essential properties of B-spline basis functions can be stated, which are visible in Figure 2.2:

1. All functions are non-negative, which means that $N_{i,p}(\xi) \geq 0, \forall \xi$.
2. The sum of all values of the basis functions at an arbitrary point ξ is equal to one. This is called the partition of unity, where $\forall \xi$, it holds that $\sum_{i=1}^{n} N_{i,p}(\xi) = 1$.
3. Each basis function $N_{i,p}(\xi)$ has a compact support that is contained in the interval $[\xi_i, \xi_{i+p+1}]$.
4. For constant and linear piecewise polynomials $(p = 0, 1)$ the basis functions are the same as for the standard constant and linear basis functions of finite elements. However, for higher polynomial orders the basis functions differ.

With the basis functions and the knot vector in hand, entire B-spline curves can be defined in the following subsection.

2.1.2. B-Spline curves

B-spline curves are constructed by calculation a linear combination of B-spline functions and coefficients in any dimension $\mathbb{R}^d$. These coefficients are called control points and can be considered as similar to *nodal coordinates* used in finite element analysis. However, the control points do not necessarily need to be interpolated and act more in a way of a scaffolding. A piecewise-polynomial B-spline curve is defined in equation (2.3) for n basis functions $N_{i,p}$ and the referring control points $\mathbf{B}_i \in \mathbb{R}^d$, where $i = 1, 2, ..., n$:

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\mathbf{B}_i. \tag{2.3}$$

An example of a B-spline curve with its basis functions is given in Figure 2.3. This curve will be used as an example for all the modifications which are presented in this chapter to have a simple and common visualization. The curve consists of an open knot vector $\Xi = [0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5]$ and consecutively of quadratic basis functions ($p = 2$). Furthermore, a repeated knot is shown at $\xi = 4$ in parametric space, therefore the interior control point is interpolated. By piecewise linear interpolation of the control points the convex hull is drawn, which is also called *control polygon*.

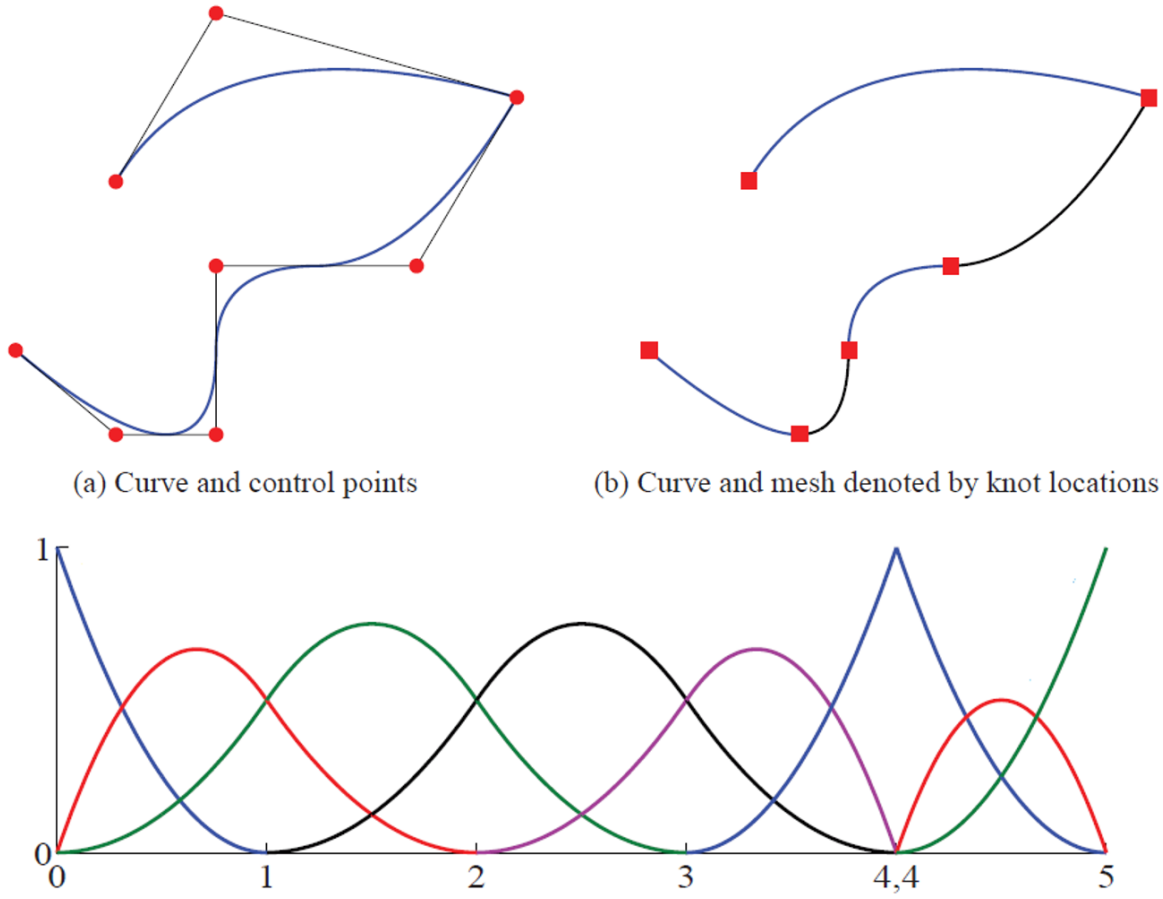


Figure 2.3: Exemplary B-spline curve with a non-uniform, open knot vector of $\Xi = [0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5]$ [8].

Again, some important properties of B-spline curves can be defined regarding continuity etc.

1. If the knot vector only consists of the numbers zero and one and $n = p$ then the curve is a Beziér curve.
2. The degree of the curve p, number of basis functions n and number of knots m_k are related by $m_k = n + p + 1$.
3. The end points of a B-spline are always interpolated since the knot vector is open.
4. B-splines provide the local modification scheme, which means that the moving of a control point $\mathbf{B}_i$ only changes the curve on the interval $[u_i, u_{i+p+1})$. This fundamentally

differs from Beziér curves and increases the stability of the calculations and decreases the calculation costs.

Furthermore, special properties with respect to the continuity of B-splines are explained in the following passages.

2.1.3. Continuity

Continuity means - in a mathematical sense - that a function varies with no abrupt breaks or jumps within the domain it is defined. The continuity is denoted as C^d, where d is the number of continuous derivatives that can be obtained. In terms of structural analysis the continuity has a high impact on the quality of the solution field, since in plate bending patches for example, C^0-continuity, where only the basis functions but not their derivatives are continuous, means that the displacements are continuous but the stresses are not. C^1-continuity means that both displacement field and stress field are continuous, in the case of continuous material.

Figure 2.4 shows the exemplary basis functions of Section 2.1.2 using quadratic basis functions with an open knot vector. In between the knots and at all the knots except at $\xi = 4$ in parametric space, the function is C^1-continuous. At $\xi = 0$, $\xi = 4$ and $\xi = 5$ the function has repeated knots and is therefore only C^0-continuous on $\xi = 4$ and C^{-1}-continuous on $\xi = 0$ and $\xi = 5$. In this special case, where the multiplicity of the knot k is equal to the polynomial order p, the basis function is interpolatory.

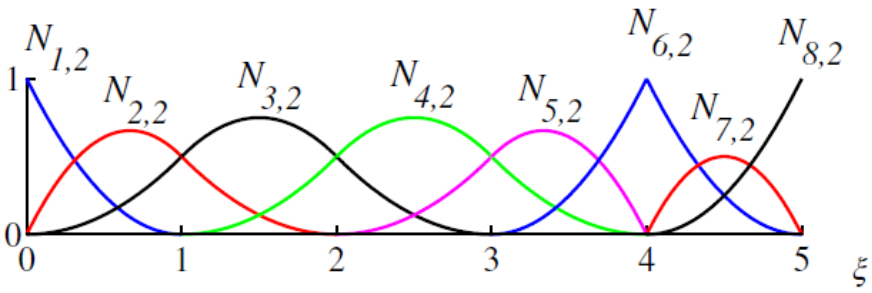


Figure 2.4: B-spline basis functions for knot vector $\Xi = [0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5]$ [9].

So in general the continuity of basis functions can be summarized as that a every basis function is C^p-continuous in between knots. However, since the multiplicity of the parametric points in the knot vector influence the continuity, the basis function has over the whole domain $p - k_i$ continuous derivatives, where k_i is the multiplicity of a knot at the point ξ_i. If the multiplicity exceeds the order p, the basis becomes discontinuous and the patch boundary is formed. However, even though the multiplicity of knots has an influence on the continuity, it does not influence the support, since the basis function $N_{i,p}$ still begins at knot ξ_i and ends at ξ_{i+p+1}.

2.1.4. B-spline surfaces

Since this work is not aimed at analyzing curves, but three-dimensional surfaces, in the following B-splines are extended to surfaces. Since a surface is a two dimensional object, control points form a control net $\mathbf{B}_{i,j}(x, y, z)$, where $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$. Additionally two knot vectors are constructed for the two directions ξ and η in parametric space as they are

$\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$ and $H = \{\eta_1, \eta_2, ..., \eta_{m+q+1}\}$. A tensor product of these curves in two orthogonal directions ends up as B-spline surface defined as

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(\xi) M_{j,q}(\xi) \mathbf{B}_{i,j}. \tag{2.4}$$

It is important to notice that as for the B-spline curve, p is the polynomial order in ξ-direction and n the number of basis functions that comprise the B-spline in the same direction. This applies in the same way for q as polynomial order in η-direction and the number of basis functions m. Also, $N_{i,p}(\xi)$ and $M_{j,q}(\eta)$ act as the basis functions. When it comes to structural analysis, the elements commonly used in finite element analysis are now the knot spans.

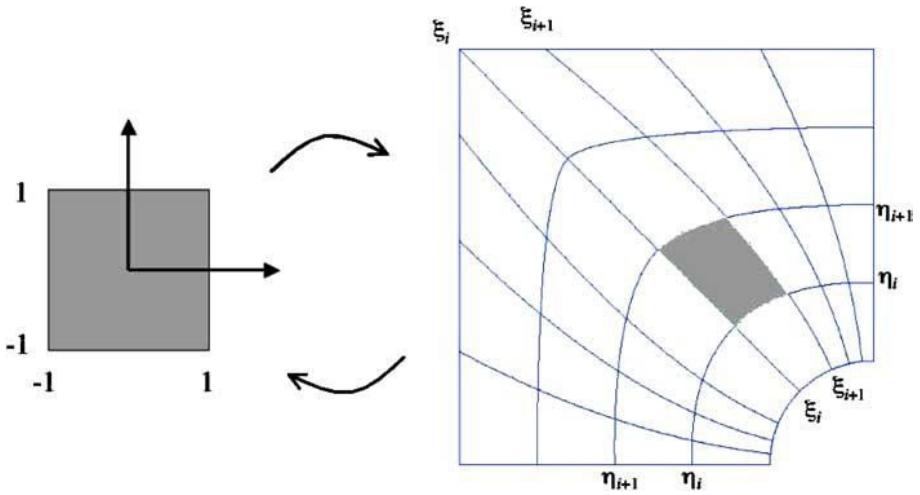


Figure 2.5: Example of how knot spans act as elements [9].

Figure 2.5 shows how a bi-unit parent element is shaped in its physical space by the knot spans $[\xi_i, \xi_{i+1}] \times [\eta_j, \eta_{j+1}]$. In this way elements on the patch can be reparametrized to run the Gauss-Legendre quadrature for the numerical integration. The precise procedure is explained in Section 2.3.

2.1.5. Rational B-splines

Using B-splines for the geometric description of an entity in $\mathbb{R}^d$ is advantageous for many applications, but when it comes to the exact description of, for example, circles and all conical sections, B-splines of the same dimension can no longer represent them. Therefore, a projective transformation of B-spline entities in $\mathbb{R}^{d+1}$ is constructed. This means in the case of a circle in $\mathbb{R}^2$, that the circle is described by a quadratic B-spline curve in $\mathbb{R}^3$ which yields a rational polynomial of the form $C_R(\xi) = f(\xi)/g(\xi)$. Consequently, the functions of $f(\xi)$ and $g(\xi)$ themselves consist of piecewise polynomials. Additionally, weighting factors w are introduced which project the projective control points defined in $\mathbb{R}^3$ into control points in $\mathbb{R}^3$. In general, this means that the new rational basis function is defined as

$$R_i^p(\xi) = \frac{N_{i,p}(\xi) w_i}{\sum_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi) w_{\hat{i}}}. \tag{2.5}$$

It is important to notice, that $\hat{i}$ has not the same meaning as i, which is the knot index, but means just the summation index. With the Non-Uniform Rational B-Spline basis function in hand, the

rational curve can be defined similarly as the B-spline curve by taking a linear combination of NURBS basis functions, which yields

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} R_i^p(\xi)\mathbf{B}_i. \tag{2.6}$$

Rational surface basis functions are determined the same way by taking the tensor product of two NURBS basis functions (2.7a) in orthogonal direction and the NURBS surface (2.7b),

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m} N_{\hat{i},p}(\xi)M_{\hat{j},q}(\eta)} \tag{2.7a}$$

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\mathbf{B}_{i,j}. \tag{2.7b}$$

If the weights of the control points of NURBS are all equal, NURBS are equal to B-splines, which makes B-splines a special case of NURBS. Therefore, all the properties of B-splines are also properties that can be stated for NURBS. So as for the B-spline basis function, the NURBS basis functions form a partition of unity, have the same support and continuity. Furthermore, the basis functions both are non-negative on the whole domain.

2.1.6. Derivatives of B-splines and NURBS

To use NURBS in an analysis framework it is not just important to determine points on a surface but also the changes at these points. There are many different ways to determine derivatives of NURBS and B-splines reported in the literature as [20] or [21]. In this work two approaches were used, the first approach employs analytical expressions and the second approach finds the derivative via finite differences.

Analytical derivation

The analytical solution considers that B-splines are defined recursively in the *Cox-de Boer recursive formula* and thus, the derivatives of B-spline functions can be determined as lower order basis functions, defined as

$$\frac{d}{d\xi}N_{i,p}(\xi) = \frac{p}{\xi_{i+p}-\xi_i}N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1}-\xi_{i+1}}N_{i+1,p-1}(\xi). \tag{2.8}$$

For a more generalized form, where the k-th derivative can be determined both sides are differentiated as

$$\frac{d^k}{d^k\xi}N_{i,p}(\xi) = \frac{p}{\xi_{i+p}-\xi_i}\left(\frac{d^{k-1}}{d^{k-1}\xi}N_{i,p-1}(\xi)\right) - \frac{p}{\xi_{i+p+1}-\xi_{i+1}}\left(\frac{d^{k-1}}{d^{k-1}\xi}N_{i+1,p-1}(\xi)\right). \tag{2.9}$$

A similar way applies for the first derivative of NURBS basis functions as

$$R_i'^p(\xi) = \frac{W_i N_i'^p(\xi)}{\sum_{i=1}^{n} w_i N_i^p(\xi)} - \frac{w_i N_i^p(\xi)\sum_{i=1}^{n} w_i N_i'^p(\xi)}{(\sum_{i=1}^{n} w_i N_i^p(\xi))^2}. \tag{2.10}$$

However, this only applies for B-spline and NURBS curves. For surfaces the partial derivatives of the NURBS basis functions $R_{ij}^{p,q}(\xi,\eta)$ need to be determined by using the quotient rule. However,

for the sake of compactness some auxiliary equations consisting of B-spline basis functions, their derivatives and weights are defined in Table 2.1. Note that, for notational simplicity, the explicit dependence of some quantities on ξ and eta is omitted.

Table 2.1: List of auxiliary equations to simplify the derivatives.

$A = \sum_{i=1}^{n}\sum_{j=1}^{m} N_i^p(\xi)M_j^q(\eta)w_{ij}$	$G = N_i^p(\xi)M_j^p(\eta)w_{i,j}$
$B = \sum_{i=1}^{n}\sum_{j=1}^{m} N_i'^p(\xi)M_j^q(\eta)w_{ij}$	$H = N_i'^p(\xi)M_j^p(\eta)w_{ij}$
$C = \sum_{i=1}^{n}\sum_{j=1}^{m} N_i^p(\xi)M_j'^q(\eta)w_{i,j}$	$I = N_i^p(\xi)M_j'^q(\eta)w_{i,j}$
$D = \sum_{i=1}^{n}\sum_{j=1}^{m} N_i'^p(\xi)M_j'^q(\eta)w_{i,j}$	$K = N_i'^p(\xi)M_j'^q(\eta)w_{i,j}$
$E = \sum_{i=1}^{n}\sum_{j=1}^{m} N_i''^p(\xi)M_j^q(\eta)w_{ij}$	$H = N_i''^p(\xi)M_j^p(\eta)w_{ij}$
$F = \sum_{i=1}^{n}\sum_{j=1}^{m} N_i^p(\xi)M_j''^q(\eta)w_{i,j}$	$L = N_i^p(\xi)M_j''^q(\eta)w_{i,j}$

Using these auxiliary equations, the first derivative with respect to ξ and η yields

$$\frac{\partial R_{i,j}^{p,q}(\xi,\eta)}{\partial \xi} = \frac{H}{A} - \frac{GB}{A^2} \tag{2.11a}$$

$$\frac{\partial R_{i,j}^{p,q}(\xi,\eta)}{\partial \eta} = \frac{I}{A} - \frac{GC}{A^2}. \tag{2.11b}$$

The second partial derivatives of the basis functions are derived in a similar ways as

$$\frac{\partial^2 R_{i,j}^{p,q}(\xi,\eta)}{\partial \xi^2} = \frac{KA - HB}{A^2} - \frac{(HB + GE)A^2 - 2GAB^2}{A^4} \tag{2.12a}$$

$$\frac{\partial^2 R_{i,j}^{p,q}(\xi,\eta)}{\partial \eta^2} = \frac{LA - IC}{A^2} - \frac{(IC + GF)A^2 - 2GAC^2}{A^4} \tag{2.12b}$$

$$\frac{\partial^2 R_{i,j}^{p,q}(\xi,\eta)}{\partial \xi \partial \eta} = \frac{JA - HC}{A^2} - \frac{(IB + GD)A^2 - 2GBAC}{A^4}. \tag{2.12c}$$

Thus, for the geometric description of the surface, all properties can be derived which are necessary in the further course. In order to be able to check these computations, however, a further method is presented, which is based on a completely different approach.

Finite Difference Method

The second method to determine the derivative of a function works numerically and is called the Finite Difference Method (FDM). The basic idea is to approximate the derivative of an arbitrary, continuous function by a difference quotient. Thereby a secant of points of equidistant orientation is formed around the point $\bar{x}$ at which the derivative is determined. If the distance between the points is minimized, the derivative at the investigated point can be determined more and more precisely, but never exactly. Depending on how the points around the examined point $\bar{x}$ are distributed, it is called a forward, backward or central difference. Figure 2.6 shows the three approaches for an example function. In this work only the central difference was used. Thus, the important equations of this approach are derived. However, with these expressions in hand, the other approaches can be easily determined. Since this method can be used for all kinds of derivatives that are upcoming in this thesis, it is described in a more general way.

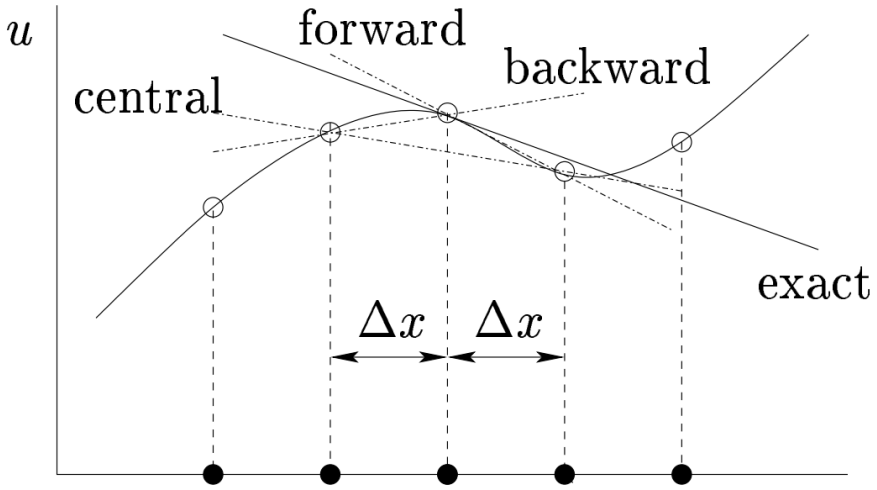


Figure 2.6: Different approaches of the finite difference method.

Let us assume that we want to find the derivative of the function $u(x)$ at point $\bar{x}$, then we determine the values of the function $u(x)$ at the points $\bar{x} - \Delta x$ and $\bar{x} + \Delta x$, where Δx is the differential increment and the points $\bar{x} \pm \Delta x$ are grid points. By dividing the difference of the two points by their distance, which is Δx twice and decreasing the differential increment towards zero yields the exact derivative at the point of $\bar{x}$. For numerical approximations Δx cannot be chosen to be equal to zero, but to be sufficiently small to determine the derivative precisely but not too small to evoke numerical instabilities. In this work Δx was chosen to be $\Delta x \in [10^{-6}, 10^{-4}]$. This yields the approximation of the first derivative as

$$\frac{\partial u}{\partial x}(\bar{x}) = \lim_{\Delta x \to 0} \frac{u(\bar{x} + \Delta x) - u(\bar{x} - \Delta x)}{2\Delta x} \approx \frac{u(\bar{x} + \Delta x) - u(\bar{x} - \Delta x)}{2\Delta x}. \tag{2.13}$$

With the first derivative in hand, the second derivative can be determined in the same way by calculation the differential quotient of the derivatives $u'(\bar{x} - \Delta x)$ and $u'(\bar{x} + \Delta x)$ as

$$\frac{\partial^2 u}{\partial x^2}(\bar{x}) = \lim_{\Delta x \to 0} \frac{u(\bar{x} + \Delta x) - 2u(\bar{x}) + u(\bar{x} - \Delta x)}{(\Delta x)^2} \approx \frac{u(\bar{x} + \Delta x) - 2u(\bar{x}) + u(\bar{x} - \Delta x)}{(\Delta x)^2}. \tag{2.14}$$

It is important to notice that for the derivation of the second derivative in (2.14) the backward and forward derivatives are considered for the derivatives at the grid points. When it comes to mixed derivatives where the derivatives are conducted in two orthogonal directions, it applies that

$$\frac{\partial^2 u}{\partial x \partial y} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y}\right) = \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}\right) \tag{2.15}$$

and, therefore,

$$\frac{\partial^2 u}{\partial x \partial y}(\bar{x}, \bar{y}) \approx \frac{u(\bar{x} + \Delta x, \bar{y} + \Delta y) - u(\bar{x} + \Delta x, \bar{y} - \Delta y) - u(\bar{x} - \Delta x, \bar{y} + \Delta y) + u(\bar{x} - \Delta x, \bar{y} - \Delta y)}{4\Delta x \Delta y}. \tag{2.16}$$

Similar to the equations above, Δy is the differential increment and the evaluated point is now in the 2D-coordinate system at point $(\bar{x}, \bar{y})$.

These equations can be used to determine the derivative, if the function is continuous on the domain and level. Which means that if a function is only C^1-continuous, it is not possible to derive the second derivative of the function.

2.2. Refinement techniques

One of the most important aspects of using B-splines and NURBS as basis functions for structural analysis is the many ways to enrich the basis with leaving the geometry and the parametrization unchanged. In the following, three different possibilities are shown with which the initial B-spline or NURBS can be refined. Not only the element size can be influenced, but also the order of the basis functions and consequently their continuity. The methods of h-, p- and k-refinement are presented and shown on the basis of the exemplary B-spline from the previous passages. Also the striking differences to the commonly used FEA are pointed out.

2.2.1. h-refinement: knot insertion

The first strategy is the h-refinement, where knots can be inserted without changing the geometry or parametric space, therefore, it is also called knot insertion. So for the insertion of a single knot into an initial knot vector $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$ and an initial amount n of control points $B = \{B_1, B_2, ..., B_n\}$ the new knot is defined at $\bar{\xi} \in [\xi_k, \xi_{k+1}]$, where k is the last knot on the initial knot vector smaller than $\bar{\xi}$. Then, the new $n+1$ control points $\bar{B} = \{\bar{B_1}, \bar{B_2}, ..., \bar{B_{n+1}}\}$ can be calculated by

$$\bar{B_i} = \alpha_i B_i + (1 - \alpha_i) B_{i-1} \tag{2.17}$$

where α_i is defined as

$$\alpha_i = \begin{cases} 1 & 1 \leq i \leq k-p, \\ \frac{\bar{\xi} - \xi_i}{\xi_{i+p} - xi_1} & k-p+1 \leq i \leq k, \\ 0 & k+1 \leq i \leq n+p+2. \end{cases} \tag{2.18}$$

This knot insertion can also be done for multiple knots simultaneously. More detailed application of this strategy can be found in [20] and [2]. An exemplary knot insertion for multiple knots is done to the example of Section 2.1.2 and shown in Figure 2.7.

The example shows that the curve is geometrically and parametrically identical. The knot insertion neither elevated the order of the basis function, nor did it smooth the C^0-continuity at $\xi = 4$. However, the domain of each interval of the basis functions is shortened. This is analogous to the classical h-refinement in finite element analysis. Furthermore, the new control points are closer to the dedicated element. This is a reasonable phenomena, since for example high-order Beziér curves have control points which are quite far away since every basis function is defined over the whole domain.

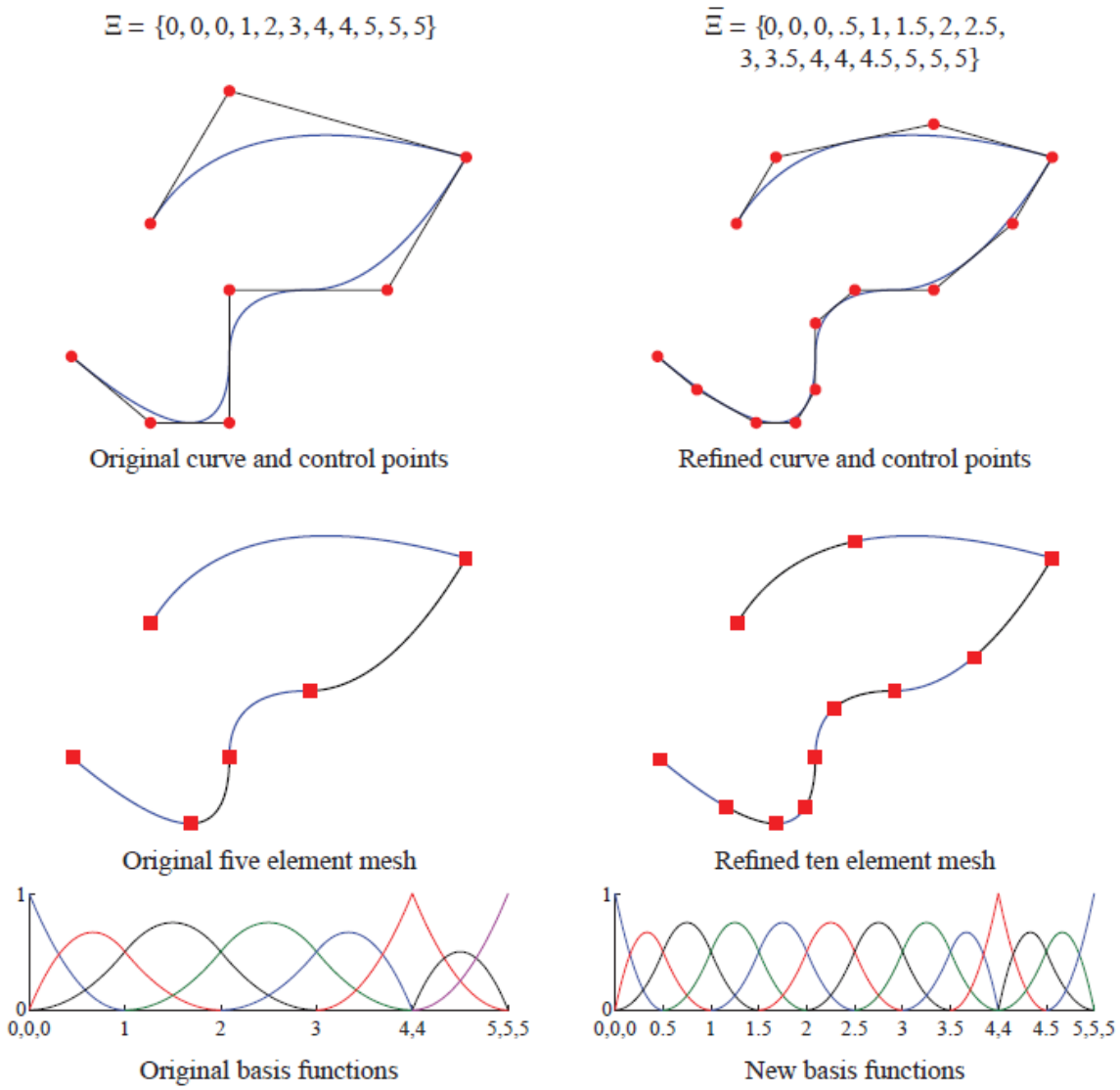


Figure 2.7: Example of h-refinement (knot insertion) [2].

2.2.2. p-refinement: order elevation

Another refinement technique is to elevate the order of the basis functions without increasing the number of elements. As for all the refinement startegies, the model is not changed geometrically. The important property of the basis functions is their continuity, where a B-spline or NURBS basis function has $(p-k)$-continuous derivatives which need to be preserved. So if the order of the basis functions is increased it is also important to preserve the discontinuities in the derivatives existing in the original curve. Therefore, the multiplicity of each knot is increased by one as well. To perform an order elevation three steps need to be done:

1. First, all existing knots are duplicated until their multiplicity k is equal to the polynomial order of the basis functions. This results in the curve being divided into segments, which are defined on the respective segment like Beziér curves.
2. Second, the order of the polynomials on each segment is elevated which results in higher order Beziér curves.
3. Third, the knots that exceed the original continuity are removed and the segments are combined into an order-elevated B-spline or NURBS.

With regard to our example from Section 2.1.2, an order elevation is applied in Figure 2.8. Thus, it can be seen that the order of the basis functions has increased and yet continuity has been preserved at the specific points and over the entire interval. Especially at point $\xi = 4$ the removal of the exceeding knots leads to the same C^0-continuity as before the refinement. This removal becomes especially clear when looking at the knot vector. Furthermore, the number of control points is increased, but not the number of elements, since the elements now consist of a higher order.

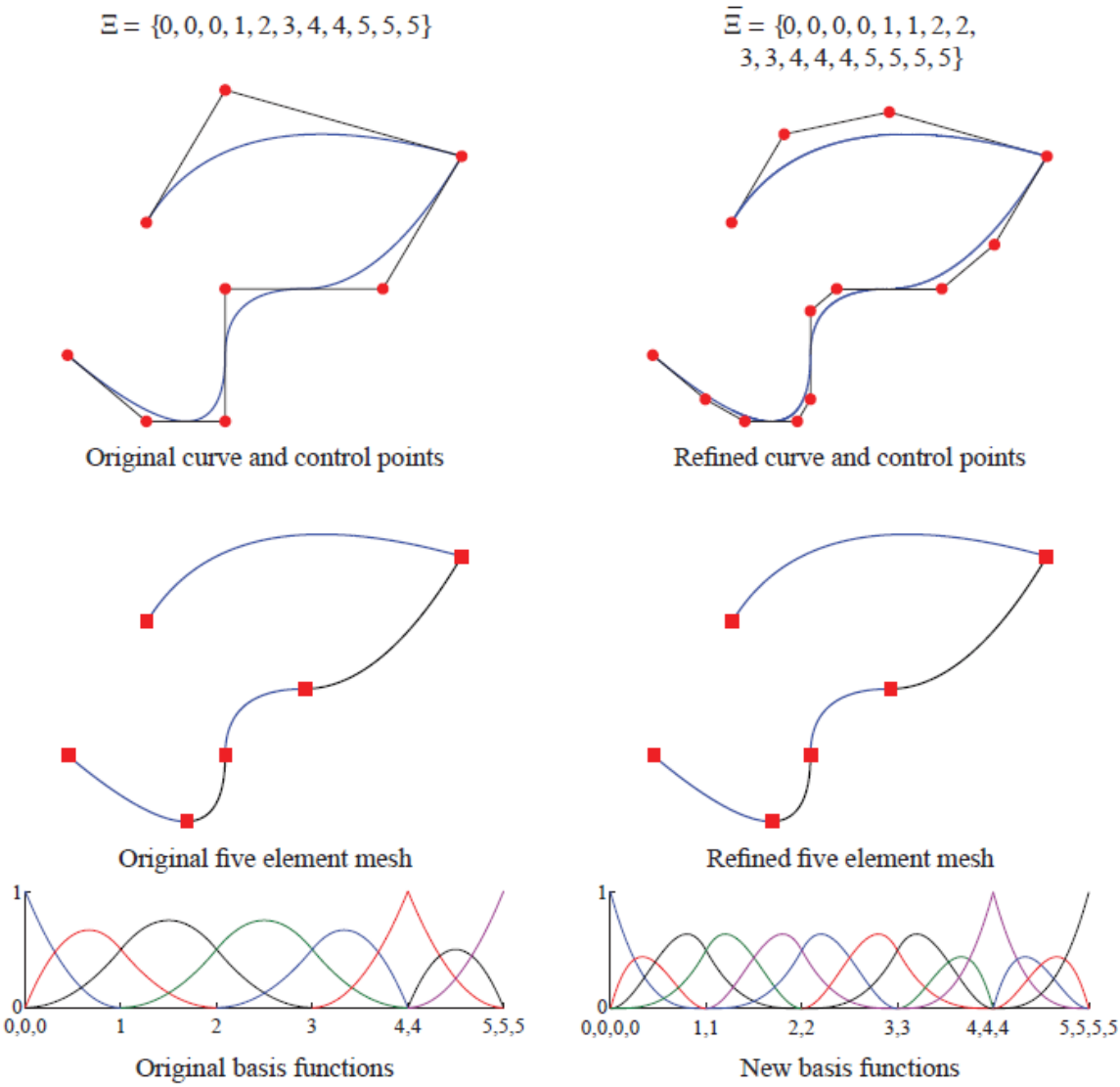


Figure 2.8: Example of p-refinement (order elevation) [2].

The order elevation has many parallels with the classical p-refinement in finite element analysis as the polynomial order of the basis functions is increased. However, it is important that the p-refinement starts with basis functions that are C^0-continuous everywhere, while the order elevation of B-splines and NURBS is compatible with any combination of continuities defined on the non-refined initial mesh.

2.2.3. k-refinement

The third refinement strategy mentioned in this work is called the k-refinement and has no comparable counterpart in finite element analysis, which makes it specific for isogeometric analysis. This technique takes into account that the order elevation and the knot insertion do not interact. The basic idea is to combine both in a way that first an order elevation is applied and then the knot insertion follows. This has the benefit that not only the mesh is refined but also the continuity is increased. This is different from the commonly used knot insertion before the order elevation. However, this refinement technique is not used in this work. For further literature and detailed information [2] is recommended.

2.3. NURBS in an analysis framework

Now that the geometric and mathematical background has been introduced, the analysis framework based on NURBS can be described, which is known as isogeometric analysis (IGA) and was developed by Hughes [9] and co-workers. As explained in the introduction of the chapter, this has the advantage that the CAD model, created in the construction process which is usually made of NURBS and B-splines, can be directly inserted into the analysis framework by using the same basis function for design and analysis. The fundamental idea of IGA is to subtitute the Lagrange polynomials used in finite element analysis by NURBS as basis functions for the geometry and the solution field. Consequently the solution is calculated at the control points and the meshing are applied in the parametric space.

2.3.1. Interaction between parameter space and physical space

Before presenting the overall framework, the observations described in this chapter are briefly summarized and combined with the description of the geometry in an algorithm for structural analyses.

Figure 2.9 shows the interaction of an isogeometric shell patch with its parental element of the calculation framework. It starts with the index space, which is a mesh made up by the indices $\xi_1, \xi_2, \xi_3, ...$ and $\eta_1, \eta_2, \eta_3, ...$ These indices can be defined by the knot vectors Ξ and H. These knot vectors then transform the index space into a parameter space. Using these two knot vectors and the weights of the control points also yields the corresponding NURBS basis functions $R_{ij}^{p,q}$ of the patch. Having determined the NURBS basis functions the surface in physical space at an arbitrary point can be computed by the sum over all the control points times the NURBS basis functions. In the figure, the control points act as a scaffolding for the surface, however, the surface does not necessarily interpolate the control points. With the surface in parameter space and physical space in hand, by applying a reparametrization the integration on the parental element is performed. This is the very crucial step to determine the stiffness matrix and the load vector and explained in detail in Section 2.3.2. Thus, the geometric description is completed and the analysis specific principles can be explained.

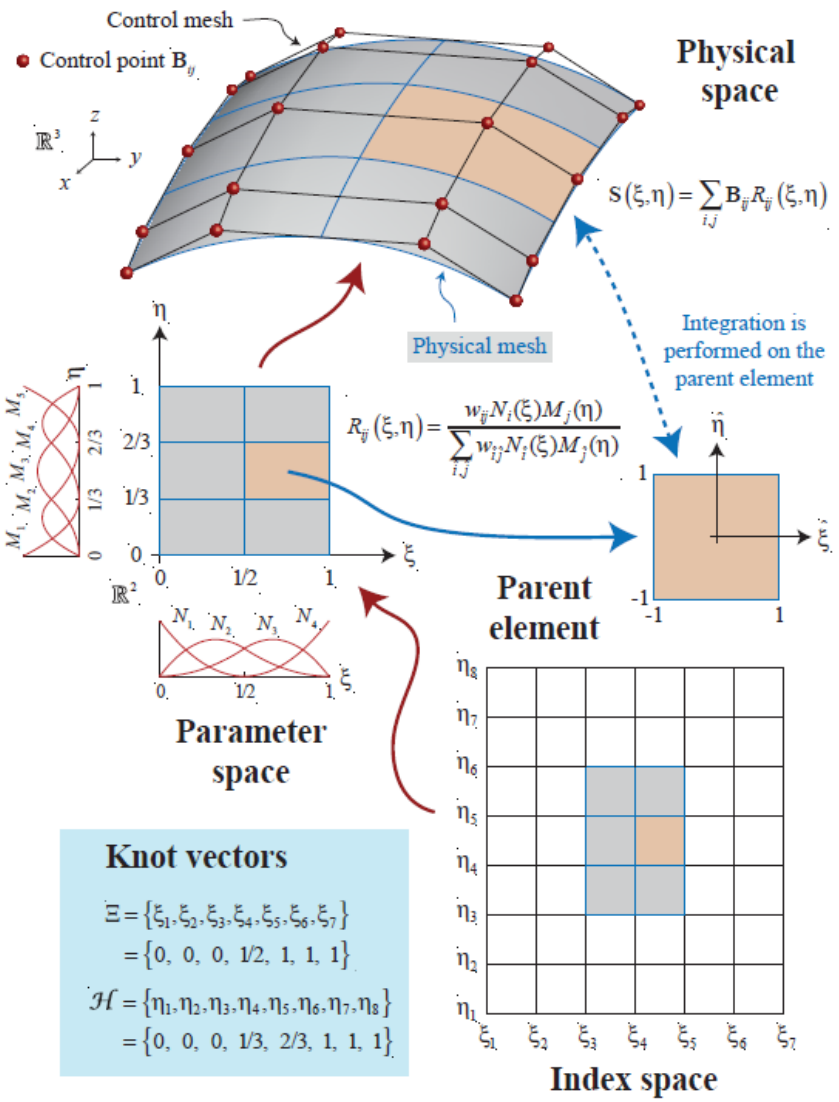


Figure 2.9: Sketch how the interaction of physical and parameter space is realized for NURBS [2].

2.3.2. Integration on the parental element

Since an analytical integration on the element is hardly possible for computational methods, several approaches were derived to solve integrals numerically. In this work, the Gauss-Legendre quadrature is used, which is a special case of the Gauss quadrature. It is a numerical integration method for the numerical solution of an integral on the interval of $[-1, 1]$. For this purpose, the integral of a function to be determined is converted into a sum of function values multiplied by so-called quadrature weights as

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = \sum_{i=1}^{n_{gp}} w_i\, f(x_i)\,, \tag{2.19}$$

where n_{gp} is the number of Gauss points to be evaluated, w_i are the quadrature weights and x_i the location of the evaluated Gauss point. The number and location of Gauss points such as the corresponding quadrature weights are predefined and can be found in plenty of books and therefore not listed again. However, it is important to know that for polynomials the numerical approach is exact for $n_{gp} = \lceil (p+q+1)/0.5 \rceil$. This means B-splines are integrated exactly if this rule is applied. For NURBS, by contrast, the integration is always approximate since NURBS

are rational by definition and therefore not polynomials. However, this method is still sufficiently accurate for rational functions as shown in [9]. To increase the accuracy of the calculations done in this work, the number of Gauss points per direction is increased by three to

$$n_{gp} = \lceil \frac{p+q+1}{2} + 3 \rceil. \tag{2.20}$$

Since the Gauss-Legendre quadrature is defined only on the interval [-1, 1], but the knot vectors of the NURBS usually start with $\xi_1 = 0$, these must be rewritten in a process called reparametrization. Thereby, the knot vectors of an element are adjusted in such a way that the basis functions to be evaluated are expressed on the integration interval.

To visualize this, an example of basis functions of a NURBS curve is chosen with two elements on the initial knot vector $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$, with the basis functions shown in Figure 2.10, where element 1 is defined on the interval of $[0, 0.5]$ and element 2 on $[0.5, 1]$.

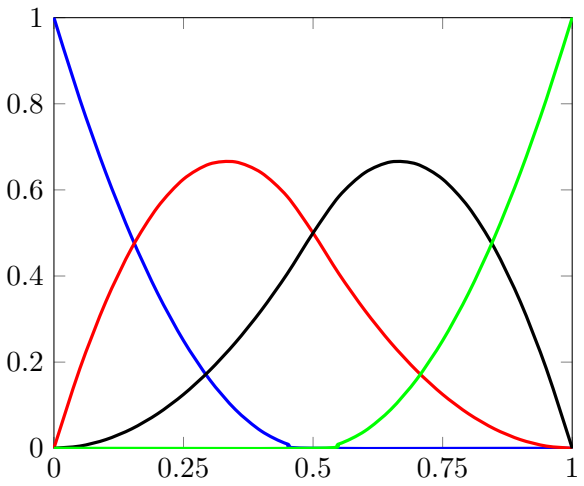


Figure 2.10: Basis functions of a NURBS curve with knot vector $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$.

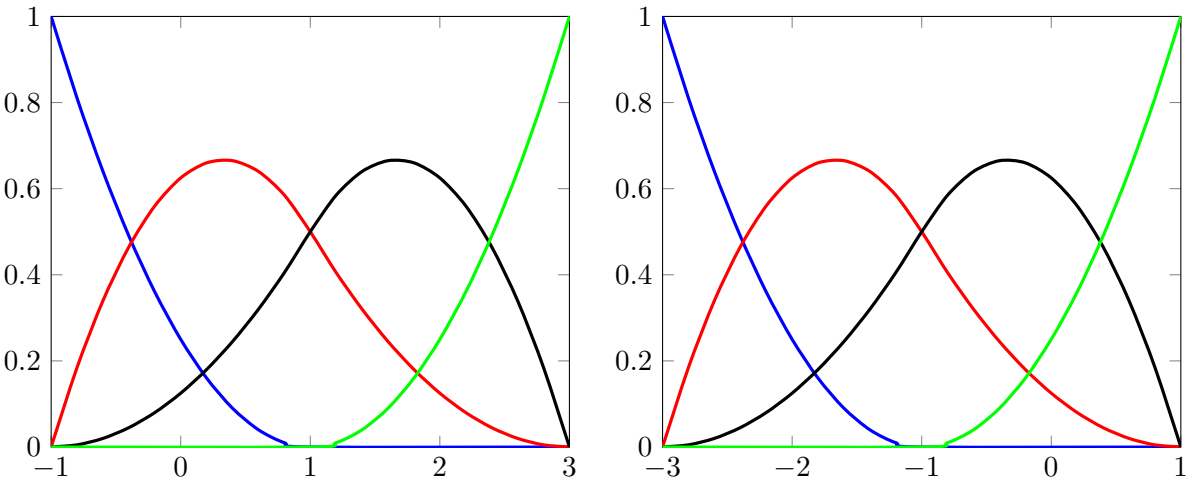


Figure 2.11: Reparametrized basis functions of element 1 (left) and element 2 (right) with knot vectors Ξ_1 =[-1,-1,-1,1,3,3,3] and Ξ_2 =[-3,-3,-3,-1,1,1,1].

To have the elements on their domain of [-1,1] the knot vector needs to be adjusted for each

element by

$$\xi_i^j = \frac{2}{l_{au}}\xi_i - 2 \cdot j + 1, \tag{2.21}$$

where ξ_i is the i-th entry on the general knot vector Ξ, ξ_i^j the modified i-th entry on the knot vector Ξ_j of the j-th element and l_{au} is the element length in parametric space. This yields the element knot vectors as Ξ_1 =[-1, -1, -1, 1, 3, 3, 3] and Ξ_2 =[-3,-3,-3,-1, 1, 1, 1]. The corresponding elementwise basis functions are shown in Figure 2.11.

When comparing Figure 2.10 with Figure 2.11, it is obvious that the basis functions have not changed and only the horizontal axis scaling has been adjusted. This ensures that the reparametrization does not cause any change of the basis functions, which is desired and can be applied similarly for surfaces, where the reparametrization is not only applied in ξ-direction but also in η-direction. Thus, the integration can be performed by means of Gauss-Legendre quadrature without changing the geometry.

2.3.3. The analysis framework

Finally, the analysis framework can be presented. The framework used in this work is a single patch loop, which means that all the elements are on one patch and therefore a continuous mesh and surface is required. An exemplary flowchart how such an architecture of an analysis framework looks like is shown in Figure 2.12.

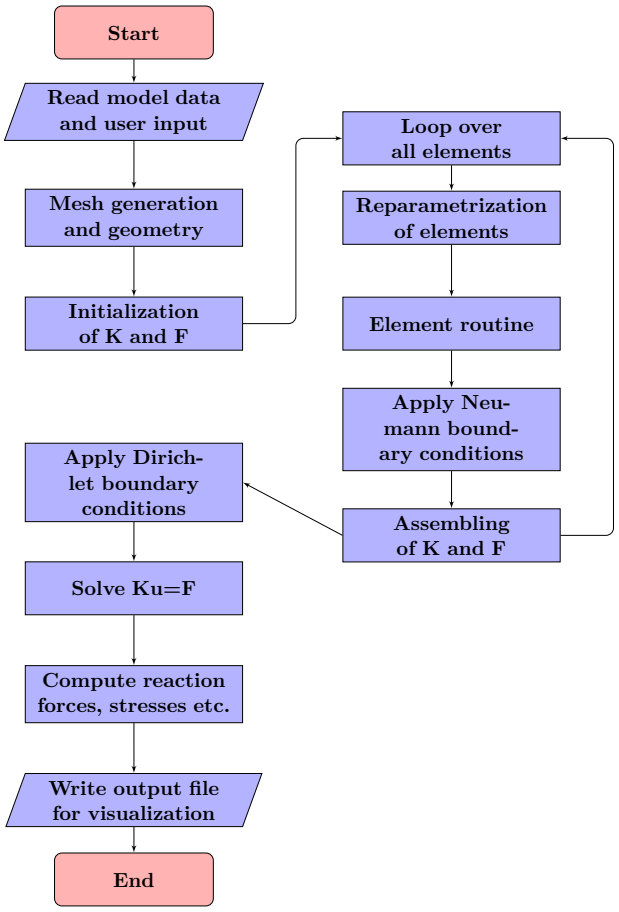


Figure 2.12: Framework using NURBS as basis functions.

It starts reading the model data and all the necessary user data as material parameters, geometry

and boundary conditions, which is in general, the definition of the boundary value problem. Afterwards, the geometry and corresponding mesh is generated and refined. Furthermore, the global stiffness matrix and the load vectors are initialized and set to zero. With all this basic information in hand, the loop over all the elements on the patch can be started. At first, the elements are adjusted for the Gauss-Legendre quadrature and therefore reparametrized. Then, the element routine computes the element stiffness matrix. At this point, element routine can be considered as some kind of black box, since this is the part, where the scaled boundary method is conducted and explained as Chapter 3. So it only matters that we have a routine that can obtain the necessary values, which is the element stiffness matrix. Subsequently, the element load vector is determined by evaluating the Neumann boundary conditions. Both, the element load vector and the element stiffness matrix are then assembled in the global stiffness matrix. Having derived and assembled all the elements, the loop is terminated and the next step is to adjust the model to the structural supports as the Dirichlet boundary conditions are applied. Now the displacements $\mathbf{u}$ can be computed by solving $\mathbf{Ku} = \mathbf{F}$. Moreover, some additional information can be gained by computing reaction forces, stresses and all necessary values. Finally the information is written down in an output file which simplifies the visualization and evaluation of the results.

Thus, in this chapter, an analysis framework has been established to compute static problems by applying the isogeometric analysis. In the following chapter, the just introduced black box of the element routine will be considered in more detail, and the scaled boundary method, which is relevant for this work, will be derived and described.

3. Scaled boundary method for shell structures

In recent years the scaled boundary method has emerged as a powerful and competitive numerical approach for the discretization of entities in boundary representation. It is based on the idea to only describe the boundary of an entity and then scaling the boundary with help of a scaling parameter. The boundary of the domain can be represented in many different ways, and so does the solution along the radial direction. Most commonly the boundary domain is discretized by finite elements and the solution along the radial direction is determined using an analytical approach and therefore named Scaled Boundary Finite Element Method (SBFEM). The SBFEM was coined in 1997 by Song and Wolf [26]. Since then, the method has been applied to many different problems such as elastostatics, crack propagation and electromagnetics or isogeometric analysis [25]. Also, scaled boundary formulations were derived for plates [16], cylindrical shells [13] and arbitrary curved shells [14], which are of particular relevance to this thesis.

As explained in the previous chapters, the discretization is conducted using isogeometric analysis combined with an analytical solution in thickness direction, which is a so called "semi-analytical" approach due to the combination of numerical and analytical solution approach and the overall method is named Isogeometric Scaled Boundary Analysis (SBIGA). For this purpose, the formulation from [14], which describes a SBFEM shell for static problems, was extended with the IGA to an SBIGA shell. At the beginning of the chapter the novel scaling strategy will be explained and the governing equations for the shell are presented. It follows the mapping of the scaled boundary shell and the interpolation of the displacement and solution field. Afterwards the isogeometric scaled boundary formulation based on the principle of virtual work is derived. Finally the analytical solution in thickness direction yields the stiffness matrix to compute the solution.

3.1. Scaling strategy for the shell

The fundamental concept of the scaled boundary shell is that it is discretized on the bottom surface and the thickness of the shell is scaled in the direction of the normal vector $\mathbf{n}$ which is always orthogonal to the shell surface and in the direction of the axis γ of the local coordinate system. Therefore, the normal starts on the bottom surface of the shell which is denoted as point A and follows along γ to point B, as shown in Figure 3.1.

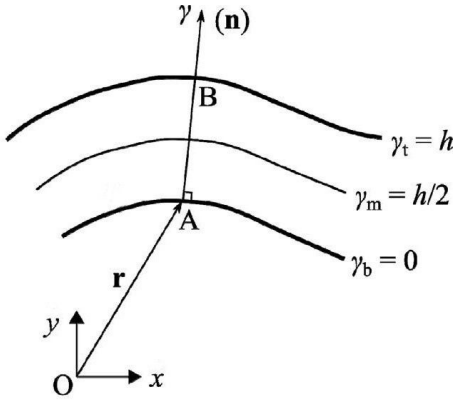


Figure 3.1: The normal scaling strategy for the shell scaled from point A onto point B along γ [14].

As demonstrated, point A on the bottom surface is described by the position vector $\mathbf{r}$ in the

global carthesian coordinate system of x, y and z and the corresponding orthonormal basis vectors $\mathbf{e}_i \quad (i = 1, 2, 3)$ as

$$\mathbf{r} = x\,\mathbf{e}_1 + y\,\mathbf{e}_2 + z\,\mathbf{e}_3 = [x \quad y \quad z]^{\mathrm{T}}. \tag{3.1}$$

Consequently, the location of point B is the sum of the position vector of A and the normal vector at the specific point times the thickness h. Therefore, any point on the line between A and B is described by the position vector $\hat{\mathbf{r}}$ use of the position vector to A $\mathbf{r}$ as

$$\hat{\mathbf{r}} = \mathbf{r} + \gamma\mathbf{n} = [\hat{x} \quad \hat{y} \quad \hat{z}]^{\mathrm{T}}, \qquad \gamma \in [0, h]. \tag{3.2}$$

Apparently, the scaling center is not fixed for all points in an element as in conventional scaled boundary solids, but changes within the shell. Thus, any point on the shell can be mapped with the normal scaling strategy, but every point on the bottom surface also acts as a scaling center.

3.2. Governing equations for the shell

Within the regime the shell is treated as a 3D problem with displacements $\mathbf{u}$ which are defined as $\mathbf{u} = \mathbf{u}(x, y, z) = [u_x \quad u_y \quad u_z]^{\mathrm{T}}$ at an arbitrary point. Assuming linearized strains leads to the strain defined from the strain-displacement relation for the strain tensor $\boldsymbol{\epsilon}$ as

$$\boldsymbol{\epsilon} = [\epsilon_{xx}\ \epsilon_{yy}\ \epsilon_{zz}\ \epsilon_{yz}\ \epsilon_{zx}\ \epsilon_{xy}]^T = \mathbf{L}\mathbf{u}, \tag{3.3}$$

where the differential operator $\mathbf{L}$ is defined as

$$\mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix}. \tag{3.4}$$

Furthermore, the material is assumed to be linear elastic and isotropic for which the elasticity matrix $\mathbb{C}$ is defined as

$$\mathbb{C} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \tag{3.5}$$

with the Lamé constants λ and μ expressed from Young's modulus E and Poisson's ratio ν as

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \qquad \text{and} \qquad \mu = \frac{E}{2(1+\nu)}. \tag{3.6}$$

Consequently, according to Hooke's law the Cauchy stress tensor $\boldsymbol{\sigma}$ can be defined as

$$\boldsymbol{\sigma}(x,y,z) = [\sigma_{xx}\ \sigma_{yy}\ \sigma_{zz}\ \sigma_{yz}\ \sigma_{zx}\ \sigma_{xy}]^{\mathrm{T}} = \mathbb{C}\boldsymbol{\epsilon}. \tag{3.7}$$

Thus, all governing equations of the shell needed for the further derivation are defined and the transformation from the global to the local coordinate system follows.

3.3. Scaled boundary transformation of shell geometry

As in the finite element method, the physical coordinates defined in the global axes x, y and z must be transformed into the local coordinate system defined in ξ and η. The goal is to rearrange the differential operator $\mathbf{L}$ derived in equation (3.4) from the Cartesian coordinate system of x, y and z to the local coordinate system of γ, ξ and η. This can be done by the Jacobian matrix, its determinant and the inverse Jacobian matrix. The Jacobian matrix $\hat{\mathbf{J}}_s$ is defined as

$$\hat{\mathbf{J}}_s(\gamma,\xi,\eta) = \begin{bmatrix} \hat{x}_{,\gamma} & \hat{y}_{,\gamma} & \hat{z}_{,\gamma} \\ \hat{x}_{,\xi} & \hat{y}_{,\xi} & \hat{z}_{,\xi} \\ \hat{x}_{,\eta} & \hat{y}_{,\eta} & \hat{z}_{,\eta} \end{bmatrix} = [\hat{\mathbf{r}}_{,\gamma} \quad \hat{\mathbf{r}}_{,\xi} \quad \hat{\mathbf{r}}_{,\eta}]^{\mathrm{T}}. \tag{3.8}$$

The entries of the Jacobian matrix consist of the partial derivatives with respect to the local coordinate system. Using equation (3.2) this can be rewritten as

$$\hat{\mathbf{r}}_{,\gamma} = \mathbf{n}, \qquad \hat{\mathbf{r}}_{,\xi} = \mathbf{r}_{,\xi} + \gamma \mathbf{n}_{,\xi}, \qquad \hat{\mathbf{r}}_{,\eta} = \mathbf{r}_{,\eta} + \gamma \mathbf{n}_{,\eta}. \tag{3.9}$$

Since the surface is described only as a 2-dimensional body the unit normal vector $\mathbf{n}$ needs to be constructed. Therefore, the tangent vectors in the direction of the local axes of ξ and η from (3.9) are used to calculate an orthonormal vector, where

$$\mathbf{n} = \frac{\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|}. \tag{3.10}$$

Furthermore, the derivative of the unit normal vector needs to be calculated, which also defines the scaling and thickness direction. This can be either done by FDM or by the analytical approach (see Section 2.1.6). Both methods are effective in their way and are used in this work. For the analytical solution the partial derivative of the unit normal vector is determined with respect to the local axis ξ and η, which leads to

$$\mathbf{n}_{,\xi} = \frac{\frac{\partial \mathbf{r}_{,\xi}}{\partial \xi} \times \mathbf{r}_{,\eta} + \mathbf{r}_{,\xi} \times \frac{\partial \mathbf{r}_{,\xi}}{\partial \xi}}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|} - \frac{(\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta})[(\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}) \cdot (\frac{\partial \mathbf{r}_{,\xi}}{\partial \xi} \times \mathbf{r}_{,\eta} + \frac{\mathbf{r}_{,\eta}}{\partial \xi} \times \mathbf{r}_{,\xi})]}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|^3} \tag{3.11a}$$

$$\mathbf{n}_{,\eta} = \frac{\frac{\partial \mathbf{r}_{,\xi}}{\partial \eta} \times \mathbf{r}_{,\eta} + \mathbf{r}_{,\xi} \times \frac{\partial \mathbf{r}_{,\xi}}{\partial \eta}}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|} - \frac{(\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta})[(\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}) \cdot (\frac{\partial \mathbf{r}_{,\xi}}{\partial \eta} \times \mathbf{r}_{,\eta} + \frac{\mathbf{r}_{,\eta}}{\partial \eta} \times \mathbf{r}_{,\xi})]}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|^3} \tag{3.11b}$$

For the sake of brevity, the detailed analytical derivation of the unit normal vector is given in Appendix A. At this point, all the variables and derivatives are known or can be determined.

Substituting the Jacobi matrix of equation (3.8) with equation (3.9) yields

$$\hat{\mathbf{J}}_s(\gamma,\xi,\eta) = \begin{bmatrix} n_x & n_y & n_z \\ x_{,\xi}+\gamma n_{x,\xi} & y_{,\xi}+\gamma n_{y,\xi} & z_{,\xi}+\gamma n_{z,\xi} \\ x_{,\eta}+\gamma n_{x,\eta} & y_{,\eta}+\gamma n_{y,\eta} & z_{,\eta}+\gamma n_{z,\eta} \end{bmatrix}. \tag{3.12}$$

and the inverse of the Jacobian matrix as

$$\hat{\mathbf{J}}_s^{-1} = \frac{1}{\hat{J}_s}\Big(\mathbf{A}+\bar{\mathbf{B}}\gamma+\mathbf{C}\gamma^2\Big). \tag{3.13}$$

Therefore, the denominator $\hat{J}_s$ can be expressed by the partial derivatives of $\hat{\mathbf{r}}$ as

$$\hat{J}_s = \hat{\mathbf{r}}_{,\gamma}\cdot(\hat{\mathbf{r}}_{,\xi}\times\hat{\mathbf{r}}_{,\eta}) = J_0+J_1\gamma+J_2\gamma^2. \tag{3.14}$$

The parameters J_0 and J_1 are

$$J_0 = n_xA_{11}+n_yA_{21}+n_zA_{31} \tag{3.15a}$$

$$J_1 = n_x\bar{B}_{11}+n_y\bar{B}_{21}+n_z\bar{B}_{31}, \tag{3.15b}$$

where the components A_{ij} and $\bar{B}_{ij}$ are denoted in (3.18). As the last entries $\mathbf{C}$ and J_2 of the terms are of higher order, they will be neglected and only linear terms will be considered. Thus, the denominator of the inverse of the Jacobian matrix of equation (3.14) can be approximated as

$$\hat{J}_s \approx J_0+J_1\gamma \equiv J_s. \tag{3.16}$$

Furthermore, the inverse of the Jacobian matrix is approximated as

$$\hat{\mathbf{J}}_s^{-1} \approx \frac{1}{J_0}\Big(\mathbf{A}+\Big(\bar{\mathbf{B}}-\frac{J_1}{J_0}\mathbf{A}\Big)\gamma\Big) \equiv \frac{1}{J_0}(\mathbf{A}+\mathbf{B}\gamma), \tag{3.17}$$

where the matrices $\mathbf{A}$, $\bar{\mathbf{B}}$ and $\mathbf{B}$ are

$$\mathbf{A} = \begin{bmatrix} y_{,\xi}z_{,\eta}-z_{,\xi}y_{,\eta} & y_{,\eta}n_z-z_{,\eta}n_y & n_yz_{,\xi}-n_zy_{,\xi} \\ z_{,\xi}x_{,\eta}-x_{,\xi}z_{,\eta} & z_{,\eta}n_x-x_{,\eta}n_z & n_zx_{,\xi}-n_xz_{,\xi} \\ x_{,\xi}y_{,\eta}-y_{,\xi}x_{,\eta} & x_{,\eta}n_y-y_{,\eta}n_x & n_xy_{,\xi}-n_yx_{,\xi} \end{bmatrix} \tag{3.18a}$$

$$\bar{\mathbf{B}} = \begin{bmatrix} y_{,\xi}n_{z,\eta}+n_{y,\xi}z_{,\eta}-z_{,\xi}n_{y,\eta}-n_{z,\xi}y_{,\eta} & n_{y,\eta}n_z-n_{z,\eta}n_y & n_yn_{z,\xi}-n_zn_{y,\xi} \\ z_{,\xi}n_{x,\eta}+n_{z,\xi}x_{,\eta}-x_{,\xi}n_{z,\eta}-n_{x,\xi}z_{,\eta} & n_{z,\eta}n_x-n_{x,\eta}n_z & n_zn_{x,\xi}-n_xn_{x,\xi} \\ x_{,\xi}n_{y,\eta}+n_{x,\xi}y_{,\eta}-y_{,\xi}n_{x,\eta}-n_{y,\xi}x_{,\eta} & n_{x,\eta}n_y-n_{y,\eta}n_x & n_xn_{y,\xi}-n_yn_{y,\xi} \end{bmatrix}. \tag{3.18b}$$

and

$$\mathbf{B} = \bar{\mathbf{B}}-\frac{J_1}{J_0}\mathbf{A}. \tag{3.18c}$$

With the inverse of the Jacobian matrix in hand, the derivatives of $\hat{x}$, $\hat{y}$ and $\hat{z}$ can be transformed to γ, ξ and η as

$$\begin{Bmatrix} \frac{\partial}{\partial z} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix} = \hat{\mathbf{J}}_s^{-1} \begin{Bmatrix} \frac{\partial}{\partial \gamma} \\ \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{Bmatrix} \approx \frac{1}{J_0}(\mathbf{A} + \mathbf{B}\gamma) \begin{Bmatrix} \frac{\partial}{\partial \gamma} \\ \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{Bmatrix}. \tag{3.19}$$

This transformation substituted into the differential operator $\mathbf{L}$ of equation (3.4) yields

$$\mathbf{L} = \left(\mathbf{b}_0^1 \frac{\partial}{\partial \gamma} + \mathbf{b}_0^2 \frac{\partial}{\partial \xi} + \mathbf{b}_0^3 \frac{\partial}{\partial \eta}\right) + \gamma\left(\mathbf{b}_1^1 \frac{\partial}{\partial \gamma} + \mathbf{b}_1^2 \frac{\partial}{\partial \xi} + \mathbf{b}_1^3 \frac{\partial}{\partial \eta}\right) \tag{3.20}$$

with

$$\mathbf{b}_0^i = \frac{1}{J_0} \begin{bmatrix} A_{1i} & 0 & 0 \\ 0 & A_{2i} & 0 \\ 0 & 0 & A_{3i} \\ 0 & A_{3i} & A_{2i} \\ A_{3i} & 0 & A_{1i} \\ A_{2i} & A_{1i} & 0 \end{bmatrix} (i = 1, 2, 3), \quad \mathbf{b}_1^i = \frac{1}{J_0} \begin{bmatrix} B_{1i} & 0 & 0 \\ 0 & B_{2i} & 0 \\ 0 & 0 & B_{3i} \\ 0 & B_{3i} & B_{2i} \\ B_{3i} & 0 & B_{1i} \\ B_{2i} & B_{1i} & 0 \end{bmatrix} (i = 1, 2, 3). \tag{3.21}$$

The entries of the matrices A_{ij} and B_{ij} are the entries of the matrices defined in equation (3.18).

3.4. Interpolation for the scaled boundary shell

For the interpolation of the geometry and the solution field of the shell the connection of isogeometric analysis and scaled boundary method is proceeded. An example of a geometry of a shell as single element of $4 \times 4 = 16$ nodes is shown in Figure 3.2.

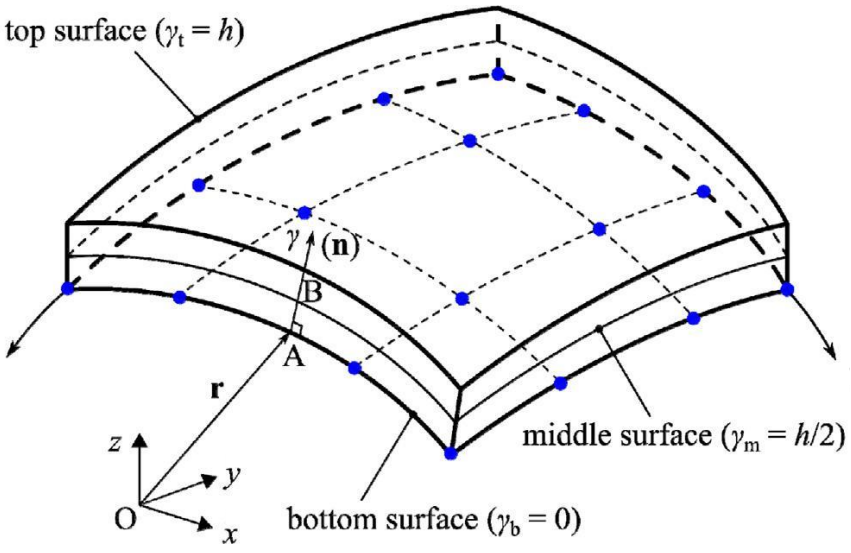


Figure 3.2: Geometry of a third-order scaled boundary shell element [14].

The position vector $\mathbf{r}$ derived in (3.1) is reformulated in terms of isogeometric analysis. There-

fore, the interpolated position vector is rewritten as

$$\mathbf{r}(\gamma,\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{R}_{i,j}^{p,q}(\xi,\eta)\mathbf{B}_{i,j} \tag{3.22}$$

and any point in the shell can be described by the equation

$$\hat{\mathbf{r}}(\gamma,\xi,\eta) = \gamma\mathbf{n} + \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{R}_{i,j}^{p,q}(\xi,\eta)\mathbf{B}_{i,j}, \tag{3.23}$$

where the second term defines the reference plane at the bottom of the shell and first term scales at any point of the shell in the thickness direction of γ. This yields that the basis functions for the structural analysis are composed of the NURBS basis functions as

$$\mathbf{N}_s(\xi,\eta) = \begin{bmatrix} R_{1,1}^{p,q} & 0 & 0 & ... & R_{n,m}^{p,q} & 0 & 0 \\ 0 & R_{1,1}^{p,q} & 0 & ... & 0 & R_{n,m}^{p,q} & 0 \\ 0 & 0 & R_{1,1}^{p,q} & ... & 0 & 0 & R_{n,m}^{p,q} \end{bmatrix}. \tag{3.24}$$

But not only the geometry is interpolated in the framework of isogeometric analysis but also the solution field, wherefore solution fields of higher continuity can be obtained. Here, the solution field is discretized for one element as

$$\mathbf{u}(\gamma,\xi,\eta) = \begin{Bmatrix} \mathbf{u}_x(\gamma,\xi,\eta) \\ \mathbf{u}_y(\gamma,\xi,\eta) \\ \mathbf{u}_z(\gamma,\xi,\eta) \end{Bmatrix} = \begin{bmatrix} \mathbf{N}_s(\xi,\eta) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_s(\xi,\eta) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{N}_s(\xi,\eta) \end{bmatrix} \begin{Bmatrix} \mathbf{u}_x(\gamma) \\ \mathbf{u}_y(\gamma) \\ \mathbf{u}_z(\gamma) \end{Bmatrix} \equiv \bar{\mathbf{N}}_s(\xi,\eta)\mathbf{u}(\gamma). \tag{3.25}$$

The same applies for the virtual solution $\delta\mathbf{u} = \mathbf{N}(\xi,\eta)\delta\mathbf{u}(\gamma)$. Thus, the strains are calculated by inserting equation (3.25) and equation (3.20) into the strain-displacement relation of equation (3.3) to

$$\boldsymbol{\epsilon} = \left(\mathbf{B}_1^1\gamma + \mathbf{B}_0^1\right)\mathbf{u}(\gamma)_{,\gamma} + \left(\mathbf{B}_1^2\gamma + \mathbf{B}_0^2\right)\mathbf{u}(\gamma), \tag{3.26}$$

where

$$\mathbf{B}_0^1 = \mathbf{b}_0^1\bar{\mathbf{N}}_s, \qquad \mathbf{B}_0^2 = \mathbf{b}_0^2(\bar{\mathbf{N}}_s)_{,\xi} + \mathbf{b}_0^3(\bar{\mathbf{N}}_s)_{,\eta} \tag{3.27}$$

$$\mathbf{B}_1^1 = \mathbf{b}_1^1\bar{\mathbf{N}}_s, \qquad \mathbf{B}_1^2 = \mathbf{b}_1^2(\bar{\mathbf{N}}_s)_{,\xi} + \mathbf{b}_1^3(\bar{\mathbf{N}}_s)_{,\eta}. \tag{3.28}$$

This yields the stress tensor as

$$\boldsymbol{\sigma} = \mathbb{C}\left[\left(\mathbf{B}_1^1\gamma + \mathbf{B}_0^1\right)\mathbf{u}(\gamma)_{,\gamma} + \left(\mathbf{B}_1^2\gamma + \mathbf{B}_0^2\right)\mathbf{u}(\gamma)\right]. \tag{3.29}$$

The following coefficient matrices are defined for the later reference, that are needed in the derivation of the scaled boundary equation for the shell using the principle of virtual work

$$E_i^{00} = \int_S \left(\mathbf{B}_0^1\right)^{\mathrm{T}}\mathbb{C}\mathbf{B}_0^1 J_i d\xi d\eta \quad (i=0,1) \tag{3.30a}$$

$$E_i^{01} = \int_S \left(\mathbf{B}_0^2\right)^{\mathrm{T}}\mathbb{C}\mathbf{B}_0^1 J_i d\xi d\eta \quad (i=0,1) \tag{3.30b}$$

$$E_i^{02} = \int_S (\mathbf{B}_0^2)^{\mathrm{T}} \mathbb{C} \mathbf{B}_0^2 J_i d\xi d\eta \quad (i = 0, 1) \tag{3.30c}$$

$$E_0^{10} = \int_S \left[(\mathbf{B}_1^1)^{\mathrm{T}} \mathbb{C} \mathbf{B}_0^1 + (\mathbf{B}_0^1)^{\mathrm{T}} \mathbb{C} \mathbf{B}_1^1 \right] J_0 d\xi d\eta \tag{3.30d}$$

$$E_0^{11} = \int_S \left[(\mathbf{B}_1^2)^{\mathrm{T}} \mathbb{C} \mathbf{B}_0^1 + (\mathbf{B}_0^2)^{\mathrm{T}} \mathbb{C} \mathbf{B}_1^1 \right] J_0 d\xi d\eta \tag{3.30e}$$

$$E_0^{12} = \int_S \left[(\mathbf{B}_1^2)^{\mathrm{T}} \mathbb{C} \mathbf{B}_0^2 + (\mathbf{B}_0^2)^{\mathrm{T}} \mathbb{C} \mathbf{B}_1^2 \right] J_0 d\xi d\eta \tag{3.30f}$$

and

$$\mathbf{E}_{01}^0 = \mathbf{E}_0^{10} + \mathbf{E}_1^{00} \tag{3.31a}$$

$$\mathbf{E}_{01}^1 = \mathbf{E}_0^{11} + \mathbf{E}_1^{01} \tag{3.31b}$$

$$\mathbf{E}_{01}^2 = \mathbf{E}_0^{12} + \mathbf{E}_1^{02}. \tag{3.31c}$$

The coefficient matrices can be split into $\mathbf{E}_0^{00}$, $\mathbf{E}_0^{01}$ and $\mathbf{E}_0^{01}$ which are dependent on J_0, therefore, they are always nonzero and derived in the same way as the coefficient matrices in [18]. The coefficient matrices $\mathbf{E}_{01}^0$, $\mathbf{E}_{01}^1$ and $\mathbf{E}_{01}^2$ are dependent on J_1 and represent the stiffness component from curvatures. Therefore, they are zero for plates.

3.5. Isogeometric scaled boundary formulation

The isogeometric scaled boundary formulation is derived on the basis of the principles of virtual work and virtual strains for elastostatics [3]. In this principle the shell body is subjected to a virtual displacement field $\delta\mathbf{u}$ which leads to the corresponding virtual strains

$$\delta\epsilon = \mathbf{L}\delta\mathbf{u}. \tag{3.32}$$

Furthermore, it is based on the equation that the virtual strain energy δU is equal to the external virtual work δW denoted as

$$\delta U - \delta W = 0. \tag{3.33}$$

Both parts of the equation will now be modified and adjusted to the conditions of the shell problem. Starting with the external virtual work, which is influenced by the boundary concentrated forces $\mathbf{T}$ and distributed tractions $\boldsymbol{\tau}$. Therefore, the external virtual work can be rewritten as

$$\delta W = \delta\mathbf{u}^{\mathrm{T}}\mathbf{T} + \int_S \delta\mathbf{u}^{\mathrm{T}}\boldsymbol{\tau} dS. \tag{3.34}$$

It is important to notice, that the formulation does not consider tractions on the side edges. However, the shell kinematics in the thickness direction enforces a linear distribution of the tractions over the side edges and will be artificially considered in the derivation of the stiffness matrix. The external virtual work can then be expressed in manners of the interpolated virtual displacement field derived in Section 3.4 as

$$\delta W = \delta\mathbf{u}(\gamma)^{\mathrm{T}} \left(\mathbf{N}_s^{-\mathrm{T}}\mathbf{T} + \int_S \mathbf{N}_s^{-\mathrm{T}}\boldsymbol{\tau}(J_0 + \gamma J_1) d\xi d\eta \right), \tag{3.35}$$

where the infinitesimal surface element S used for the integral is approximated as

$$dS \approx J_s d\xi d\eta = (J_0 + \gamma J_1) d\xi d\eta. \tag{3.36}$$

Under the assumption that loads only nodes at the top and bottom of the surface can be subjected to external loads, equation (3.35) can be modified to

$$\delta W = \delta \mathbf{u}(\gamma_b)^{\mathrm{T}} \mathbf{f}_b + \delta \mathbf{u}(\gamma_t)^{\mathrm{T}} \mathbf{f}_t, \tag{3.37}$$

where the nodal forces are denoted as $\mathbf{f}_b$ and $\mathbf{f}_t$ on top and bottom that are equivalent to the tractions and concentrated forces on surfaces of the shell. These nodal forces consist of the concentrated loads and the distributed loads as

$$\mathbf{f}_b = \mathbf{N}_s^{-\mathrm{T}} \mathbf{T}_b + \int_S \mathbf{N}_s^{-\mathrm{T}} \boldsymbol{\tau}_b J_0 d\xi d\eta \tag{3.38a}$$

$$\mathbf{f}_t = \mathbf{N}_s^{-\mathrm{T}} \mathbf{T}_t + \int_S \mathbf{N}_s^{-\mathrm{T}} \boldsymbol{\tau}_t (J_0 + h J_1) d\xi d\eta. \tag{3.38b}$$

Having derived the final state of the virtual external work, the counterpart of the equation is determined. The virtual strain energy is given as the integral over all the virtual strains multiplied by the stresses in the shell body by

$$\delta U = \int_V \delta \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{\sigma} dV. \tag{3.39}$$

Let the internal nodal forces at an arbitrary surface with constant γ be defined as

$$\mathbf{q}(\gamma) = \int_S \left[\left(\mathbf{B}_1^1\right)^{\mathrm{T}} \gamma + \left(\mathbf{B}_0^1\right)^{\mathrm{T}} \right] \boldsymbol{\sigma} (J_0 + \gamma J_1) d\xi d\eta. \tag{3.40}$$

Substituting $\mathbf{q}$ in the virtual strain energy from equation (3.39) yields

$$\delta U = \int_{\gamma_b}^{\gamma_t} \delta \mathbf{u}(\gamma)_{,\gamma}^{\mathrm{T}} \mathbf{q}(\gamma) d\gamma + \int_{\gamma_b}^{\gamma_t} \delta \mathbf{u}(\gamma)^{\mathrm{T}} \left(\int_S \left[\left(\mathbf{B}_1^2\right)^{\mathrm{T}} \gamma + \left(\mathbf{B}_0^2\right)^{\mathrm{T}} \right] \boldsymbol{\sigma} (J_0 + \gamma J_1) d\xi d\eta \right) d\gamma. \tag{3.41}$$

Here the infinitesimal volume dV for any γ is approximated that

$$dV = \mathbf{r}_{,\gamma} \cdot (\hat{\mathbf{r}}_{,\xi} \times \hat{\mathbf{r}}_{,\eta}) d\gamma d\xi d\eta = \hat{J}_s d\gamma d\xi d\eta \approx J_s d\gamma d\xi d\eta = (J_0 + J_1 \gamma) d\gamma d\xi d\eta \tag{3.42}$$

In a similar procedure as for the external virtual work, the integration by terms is applied and yields the virtual strain energy as

$$\begin{aligned} \delta U = & \left[\delta \mathbf{u}(\gamma)^{\mathrm{T}} \mathbf{q}(\gamma) \right] \Big|_{\gamma_b}^{\gamma_t} \\ & + \int_{\gamma_b}^{\gamma_t} \delta \mathbf{u}(\gamma)^{\mathrm{T}} \left[-\mathbf{q}(\gamma)_{,\gamma} + \mathbf{E}_{01}^1 \gamma \mathbf{u}(\gamma)_{,\gamma} + \mathbf{E}_0^{01} \mathbf{u}(\gamma)_{,\gamma} + \mathbf{E}_{01}^2 \gamma \mathbf{u}(\gamma)_{,\gamma} + \mathbf{E}_0^{02} \mathbf{u}(\gamma) \right] d\gamma. \end{aligned} \tag{3.43}$$

By evaluating the first term, the final state of the virtual strain energy yields

$$\begin{aligned} \delta U = & \delta \mathbf{u}(\gamma_t)^{\mathrm{T}} \mathbf{q}(\gamma_t) - \delta \mathbf{u}(\gamma_b)^{\mathrm{T}} \mathbf{q}(\gamma_b) \\ & + \int_{\gamma_b}^{\gamma_t} \delta \mathbf{u}(\gamma)^{\mathrm{T}} \left[-\mathbf{q}(\gamma)_{,\gamma} + \mathbf{E}_{01}^1 \gamma \mathbf{u}(\gamma)_{,\gamma} + \mathbf{E}_0^{01} \mathbf{u}(\gamma)_{,\gamma} + \mathbf{E}_{01}^2 \gamma \mathbf{u}(\gamma)_{,\gamma} + \mathbf{E}_0^{02} \mathbf{u}(\gamma) \right] d\gamma. \end{aligned} \tag{3.44}$$

Having derived the external virtual work in equation (3.37) and the virtual strain energy in equation (3.44) both, the principle of virtual work from equation (3.33) can be reformulated as

$$\begin{aligned}&\delta\mathbf{u}(\gamma_t)^{\mathrm{T}}\left[\mathbf{q}(\gamma_t)-\mathbf{f}_t\right]-\delta\mathbf{u}(\gamma_b)^{\mathrm{T}}\left[\mathbf{q}(\gamma_b)-\mathbf{f}_b\right]\\&\quad+\int_{\gamma_b}^{\gamma_t}\delta\mathbf{u}(\gamma)^{\mathrm{T}}\left[-\mathbf{q}(\gamma)_{,\gamma}+\mathbf{E}_{01}^{1}\gamma\mathbf{u}(\gamma)_{,\gamma}+\mathbf{E}_{0}^{01}\mathbf{u}(\gamma)_{,\gamma}+\mathbf{E}_{01}^{2}\gamma\mathbf{u}(\gamma)+\mathbf{E}_{0}^{02}\mathbf{u}(\gamma)\right]d\gamma=0.\end{aligned}\tag{3.45}$$

This equation has to hold for any $\delta\mathbf{u}(\gamma)$. It follows, that three conditions need to be satisfied:

$$\mathbf{q}(\gamma_b)+\mathbf{f}_b=\mathbf{0}\tag{3.46a}$$

$$\mathbf{q}(\gamma_t)-\mathbf{f}_t=\mathbf{0}\tag{3.46b}$$

$$-\mathbf{q}(\gamma)_{,\gamma}+\mathbf{E}_{01}^{1}\gamma\mathbf{u}(\gamma)_{,\gamma}+\mathbf{E}_{0}^{01}\mathbf{u}(\gamma)_{,\gamma}+\mathbf{E}_{01}^{2}\gamma\mathbf{u}(\gamma)+\mathbf{E}_{0}^{02}\mathbf{u}(\gamma)=\mathbf{0}.\tag{3.46c}$$

Furthermore, the internal nodal forces $\mathbf{q}$ of equation (3.40) can be rearranged by substituting the stress vector $\boldsymbol{\sigma}$ by equation (3.29) and using the interpolation of the strains. This yields

$$\mathbf{q}(\gamma)=\mathbf{E}_{01}^{0}\gamma\mathbf{u}(\gamma)_{,\gamma}+\mathbf{E}_{0}^{00}\mathbf{u}(\gamma)_{,\gamma}+\left(\mathbf{E}_{01}^{1}\right)^{\mathrm{T}}\gamma\mathbf{u}(\gamma)+\left(\mathbf{E}_{0}^{01}\right)^{\mathrm{T}}\mathbf{u}(\gamma)\tag{3.47}$$

and the derivative

$$\mathbf{q}(\gamma)_{,\gamma}=\mathbf{E}_{01}^{0}\gamma\mathbf{u}(\gamma)_{,\gamma\gamma}+\mathbf{E}_{0}^{00}\mathbf{u}(\gamma)_{,\gamma\gamma}+\left(\mathbf{E}_{01}^{1}\right)^{\mathrm{T}}\gamma\mathbf{u}(\gamma)_{,\gamma}+\left[\mathbf{E}_{01}^{0}+\left(\mathbf{E}_{0}^{01}\right)^{\mathrm{T}}\right]\mathbf{u}(\gamma)_{,\gamma}+\left(\mathbf{E}_{01}^{1}\right)^{\mathrm{T}}\mathbf{u}(\gamma).\tag{3.48}$$

Consequently, inserting (3.48) into the third condition of the virtual strains (3.46c), the isogeometric scaled boundary equation for the shell is derived as

$$\begin{aligned}&\mathbf{E}_{01}^{0}\gamma\mathbf{u}(\gamma)_{,\gamma\gamma}+\mathbf{E}_{0}^{00}\mathbf{u}(\gamma)_{,\gamma\gamma}+\left[\left(\mathbf{E}_{01}^{1}\right)^{\mathrm{T}}-\mathbf{E}_{01}^{1}\right]\gamma\mathbf{u}(\gamma)_{,\gamma}\\&\quad+\left[\left(\mathbf{E}_{0}^{01}\right)^{\mathrm{T}}-\mathbf{E}_{0}^{01}+\mathbf{E}_{01}^{0}\right]\mathbf{u}(\gamma)_{,\gamma}+\mathbf{E}_{01}^{2}\gamma\mathbf{u}(\gamma)\left[-\mathbf{E}_{0}^{02}+\left(\mathbf{E}_{01}^{1}\right)^{\mathrm{T}}\right]\mathbf{u}(\gamma)=\mathbf{0}.\end{aligned}\tag{3.49}$$

3.6. Analytical solution in thickness direction

After the derivation of the isogeometric scaled boundary equation, a consideration in thickness direction needs to be done to come up with the derivation of the stiffness matrix. Several methods can be used to derive the fineal stiffness matrix for example the semi-analytical method, the isogeometric collocation method or the weak form of equilibrium equations, which was studied for plates in [29]. Since this work follows the description of [14], the derivation follows along the analytical solution of [18] with some modifications to consider the curvature parts of the shell. At first, a variable $\mathbf{X}$ is introduced as

$$\mathbf{X}(\gamma)=\begin{Bmatrix}\mathbf{u}(\gamma)\\\mathbf{q}(\gamma)\end{Bmatrix}.\tag{3.50}$$

Using this variable the equations (3.46c) and a slightly modified equation (3.40) can be expressed as

$$\mathbf{X}(\gamma)_{,\gamma}=-(\mathbf{Z}_0+\mathbf{Z}_1\gamma)\mathbf{X}(\gamma).\tag{3.51}$$

The coefficients matrices $\mathbf{Z}_0$ and $\mathbf{Z}_1$ are

$$\mathbf{Z}_0 = \begin{bmatrix} \left(\mathbf{E}_0^{00}\right)^{-1} \left(\mathbf{E}_0^{01}\right)^{\mathrm{T}} & -\left(\mathbf{E}_0^{00}\right)^{-1} \\ -\mathbf{E}_0^{02} + \mathbf{E}_0^{01} \left(\mathbf{E}_0^{00}\right)^{-1} \left(\mathbf{E}_0^{01}\right)^{\mathrm{T}} & -\mathbf{E}_0^{01} \left(\mathbf{E}_0^{00}\right)^{-1} \end{bmatrix} \tag{3.52a}$$

$$\mathbf{Z}_1 = \begin{bmatrix} \left(\left(\mathbf{E}_0^{00}\right)^{-1} \left[\left(\mathbf{E}_{01}^{1}\right)^{\mathrm{T}} - \mathbf{E}_{01}^{0} \left(\mathbf{E}_0^{00}\right)^{-1} \left(\mathbf{E}_0^{01}\right)^{\mathrm{T}}\right]\right) & \left(\mathbf{E}_0^{00}\right)^{-1} \mathbf{E}_{01}^{0} \left(\mathbf{E}_0^{00}\right)^{-1} \\ -\mathbf{E}_{01}^{2} + \mathbf{E}_0^{01} \left(\mathbf{E}_0^{00}\right)^{-1} \left(\mathbf{E}_{01}^{1}\right)^{\mathrm{T}} - \left[\mathbf{E}_0^{01} \left(\mathbf{E}_0^{00}\right)^{-1} \mathbf{E}_{01}^{0} - \mathbf{E}_{01}^{1}\right] \left(\mathbf{E}_0^{00}\right)^{-1} \left(\mathbf{E}_0^{01}\right)^{\mathrm{T}} & \left[\mathbf{E}_0^{01} \left(\mathbf{E}_0^{00}\right)^{-1} \mathbf{E}_{01}^{0} - \mathbf{E}_{01}^{1}\right] \left(\mathbf{E}_0^{00}\right)^{-1} \end{bmatrix}. \tag{3.52b}$$

This yields the general solution of the differential equation as

$$\mathbf{X}(\gamma) = \exp\left[-(\mathbf{Z}_0\gamma + \mathbf{Z}_1\gamma^2/2)\right] \mathbf{c}, \tag{3.53}$$

where $\mathbf{c}$ is the integration constant in the dimension of a vector. Applying the boundary conditions of equation (3.46a) and (3.46b) yields the general solution as

$$\begin{Bmatrix} \mathbf{u}_t \\ \mathbf{f}_t \end{Bmatrix} = \exp(-\mathbf{Z}h) \begin{Bmatrix} \mathbf{u}_b \\ \mathbf{f}_b \end{Bmatrix} \tag{3.54}$$

and the $\mathbf{Z}$-matrix as

$$\mathbf{Z} = \mathbf{Z}_0 + \frac{h}{2}\mathbf{Z}_1. \tag{3.55}$$

Using the Padé expansion and transforming the equation from translational degrees of freedom at the top and bottom surface only to translational and rotational degrees of freedom at the bottom surface, the matrix exponential (3.54) can be expressed as

$$h \begin{bmatrix} \mathbf{I} + \frac{h}{2}\mathbf{E}_{01}^{0}\left(\mathbf{E}_0^{00}\right)^{-1} & h^2\mathbf{E}_0^{00}\mathbf{V}_{12} \\ \frac{h}{2}\mathbf{E}_{01}^{1}\left(\mathbf{E}_0^{00}\right)^{-1} & \mathbf{I} + h^2\left(\mathbf{E}_0^{01}\mathbf{V}_{12} - \mathbf{V}_{22}\right) \end{bmatrix} \times \begin{bmatrix} \mathbf{E}_0^{00} + h^2\mathbf{E}_0^{00}\mathbf{V}_{11} & \mathbf{E}_0^{00}\mathbf{Z}_{11} \\ \mathbf{E}_0^{01} + h^2\left(\mathbf{E}_0^{01}\mathbf{V}_{11} - \mathbf{V}_{21}\right) & \mathbf{E}_0^{01}\mathbf{Z}_{11} - \mathbf{Z}_{21} \end{bmatrix} \begin{Bmatrix} \theta \\ \mathbf{u} \end{Bmatrix} = \begin{Bmatrix} \mathbf{m} \\ \mathbf{f} \end{Bmatrix} \tag{3.56}$$

where $\mathbf{V}$ can be obtained by $\mathbf{Z}$ as

$$\mathbf{V} = \frac{1}{12}\mathbf{Z}^2 = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{21} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix}. \tag{3.57}$$

Consequently, substituting $\mathbf{V}$ in equation (3.56) by equation (3.55) and (3.57) yields

$$\mathbf{K} \begin{Bmatrix} \theta \\ \mathbf{u} \end{Bmatrix} = \begin{Bmatrix} \mathbf{m} \\ \mathbf{f} \end{Bmatrix} \tag{3.58}$$

where the stiffness matrix $\mathbf{K}$ is

$$\mathbf{K} = h \begin{bmatrix} \mathbf{I} + \frac{h}{2}\mathbf{E}_{01}^{0}(\mathbf{E}_{0}^{00})^{-1} & \frac{h^2}{12}\left[\mathbf{E}_{0}^{01} - (\mathbf{E}_{0}^{01})^{\mathrm{T}}\right](\mathbf{E}_{0}^{00})^{-1} \\ \frac{h}{2}\mathbf{E}_{01}^{1}(\mathbf{E}_{0}^{00})^{-1} & \mathbf{I} - \frac{h^2}{12}\mathbf{E}_{0}^{02}(\mathbf{E}_{0}^{00})^{-1} \end{bmatrix}$$
$$\times \begin{bmatrix} \mathbf{E}_{0}^{00} + \frac{h^2}{12}\left(\left[(\mathbf{E}_{0}^{01})^{\mathrm{T}} - \mathbf{E}_{0}^{01}\right](\mathbf{E}_{0}^{00})^{-1}(\mathbf{E}_{0}^{01})^{\mathrm{T}} + \mathbf{E}_{0}^{02}\right) & (\mathbf{E}_{0}^{01})^{\mathrm{T}} + \frac{h}{2}\left[(\mathbf{E}_{01}^{1})^{\mathrm{T}} - \mathbf{E}_{01}^{0}(\mathbf{E}_{0}^{00})^{-1}(\mathbf{E}_{0}^{01})^{\mathrm{T}}\right] \\ \mathbf{E}_{0}^{01} + \frac{h^2}{12}\mathbf{E}_{0}^{02}(\mathbf{E}_{0}^{00})^{-1}(\mathbf{E}_{0}^{01})^{\mathrm{T}} & \mathbf{E}_{0}^{02} + \frac{h}{2}\left[\mathbf{E}_{01}^{2} - \mathbf{E}_{01}^{1}(\mathbf{E}_{0}^{00})^{-1}(\mathbf{E}_{0}^{01})^{\mathrm{T}}\right]. \end{bmatrix} \tag{3.59}$$

Equation (3.58) can be rearranged from translational and rotational nodes at the bottom surface of the shell to only translational nodes at top and bottom as

$$\mathbf{T}^{-\mathrm{T}}\mathbf{K}\mathbf{T}^{-\mathbf{1}} \begin{Bmatrix} \mathbf{u}_b \\ \mathbf{u}_t \end{Bmatrix} = \mathbf{K}_s \begin{Bmatrix} \mathbf{u}_b \\ \mathbf{u}_t \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_b \\ \mathbf{f}_t \end{Bmatrix} \tag{3.60}$$

where the transformation matrices $\mathbf{T}$ and $\mathbf{T}^{-\mathrm{T}}$ are

$$\mathbf{T} = \begin{bmatrix} -\frac{h}{2}\mathbf{I} & \mathbf{I} \\ \frac{h}{2}\mathbf{I} & \mathbf{I} \end{bmatrix} \qquad \text{and} \qquad \mathbf{T}^{-\mathrm{T}} = \begin{bmatrix} -\frac{1}{h}\mathbf{I} & \frac{1}{2}\mathbf{I} \\ \frac{1}{h}\mathbf{I} & \frac{1}{2}\mathbf{I} \end{bmatrix}. \tag{3.61}$$

Further, the formulation for translational degrees of freedom is applied to the top and bottom side of the shell. Thus, in this chapter an isogeometric scaled boundary method for shells could be derived, which combines the properties of the scaled boundary method with the properties of the isogeometric analysis. In the following, the equations derived here theoretically are first transferred into a numerical implementation in order to then perform the obligatory benchmarking (see Chapter 5).

4. Numerical implementation

In this chapter the numerical implementation of the 3D isogeometric scaled boundary method for shells is summarized and the most important parts of the algorithm is explained. Same as conventional simulation software, it consists of three parts: preprocessing, processing and postprocessing. The parts are explained below with their functions and calculations and their order is shown in the flow chart below.

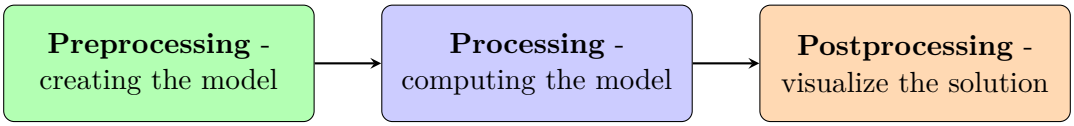


Figure 4.1: The fundamental parts of a simulation for numerical analysis.

Recapitulating the isogeometric framework from Section 2.3.3, the processes can be categorized into the specific areas as shown in Figure 4.2, where the green squares define the preprocessing, the blue squares are the processing and the orange squares are the postprocessing operations.

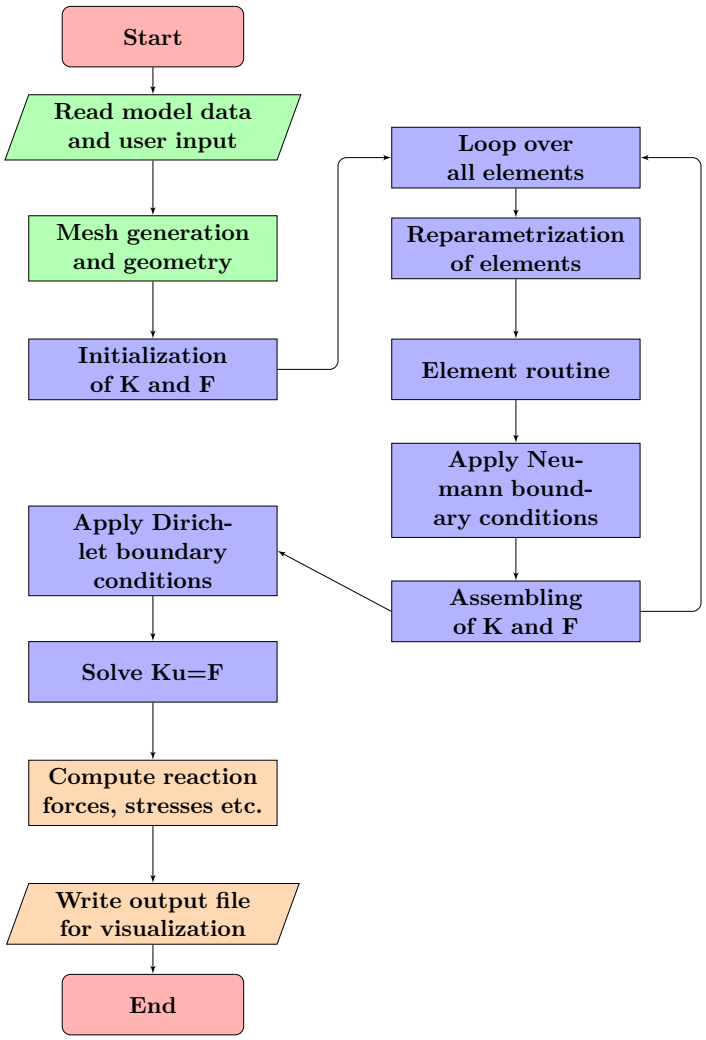


Figure 4.2: Framework using NURBS as basis functions.

As shown, the preprocessing is the part where the model is created, modified and changed by using the input data and processing the mesh data. Afterwards, in the processing the

model properties are collected and the virtual model is calculated. Therefore, the corresponding stiffness matrix is determined using the element routine and the boundary conditions are applied. At last, in the preprocessing the solution is evaluated and edited for a proper visualization of the results.

In this work all calculations were done using the numeric computing environment MATLAB 2017b of the company MathWorks. It provides an ease of use especially for users who do not have a background in computer science and it offers a variety of predefined functions, which can be used for a simplified coding [19].

4.1. Preprocessing

In the preprocessing all the required information for the computation of the problem are set by the user and auxiliary functions to define the model. The information includes the geometry, initial mesh, mesh refinement, material parameters, boundary conditions and level of resolution of displacements for the postprocessing. After initialization a function is called which is annotated as Algorithm 1. Here the CAD-specific properties are created like the open knot vectors, coordinates of the control points and weights.

Algorithm 1 Preprocessing/Mesh generation

Input:

Length L
Width B or radius r
Thickness t
CAD-model CAD
refinement flag $netzv$ and number of elements per parametric direction $anzpu$, $anzpv$

Output:

Coordinates of control points **KP**
Weights of control points **W**
Weighted coordinates of control points **KPW**
Open knot vectors per parametric direction **U** and **V**
Order of basis functions per parametric direction p and q
Total number of nodes $nnode$
Number of degrees of freedom $ndof$
Number of elements nel

Choose coordinates of control points with weight for initial mesh according to the desired CAD model determined by the input CAD
Create the open knot vectors **U** and **V** for the initial mesh in tangential directions
Evaluate the refinement flag $netzv$
if $netzv$ =1 **then**
    Generate the new knot vectors **U** and **V**
    Determine the new number of nodes per direction n and m
    Proceed a refinement of the control points **KP** and weights **W**
    Compute the new weighted control points **KPW**
end
else
    no meshrefinement will be conducted
end
Determine the total number of nodes in 3D $nnode$, the number of degrees of freedom in 3D $ndof$ and the number of elements nel
Return: **KP**, **W**, **KPW**, **U**, **V**, p, q, $nnode$, $ndof$ and nel

So far, models with their initial meshes of a plate, a quarter ring, an eighth of a cylinder, a Scordelis-Lo roof and a hemisphere with and without a hole stored (for control points see Appendix B). The initial mesh can be choosen of different orders so that afterwards an h-refinement can be conducted to refine the mesh. The mesh refinement can be activated by setting the refinement flag *netzv* to one. This is an *h*-refinement which is explained in Section 2.2.1. After the set up of the geometry, the corresponding values are defined, which are the necessary values for the further computation as the total number of nodes *nnode*, of degrees of freedom *ndof* and of elements *nel*. It is noteworthy that the thickness of the shell is not considered in this algorithm since only the bottom surface of the shell is prescribed.

4.2. Processing

In the processing the problem created by the user and set up in the preprocessing is computed by evaluating the geometry and the boundary conditions. At first, the global stiffness matrix $\mathbf{K}$ and the global load vector $\mathbf{F}$ are initialized. Then a a loop over all the elements defined on the knot vectors is performed. Thereby, the reparametrization of the evaluated element is proceeded and afterwards the element routine is conducted to compute the element stiffness matrix $\mathbf{K}_e$, followed by the application of the loads to determine the element load vector $\mathbf{F}_e$ and the assembling in the global stiffness matrix and the global load vector.

Algorithm 2 Element routine

Input:
- Material parameters E and μ
- Control points $\mathbf{KP}$ and weights $\mathbf{W}$
- Knot vectors $\mathbf{U}$ and $\mathbf{V}$
- Numbers of element in index space eli, elj
- Thickness t

Output:
- Element stiffness matrix $\mathbf{K}_{s,e}$
- Internal load vector $\mathbf{F}_e$

1 Read out material parameters E and μ to build elasticity matrix $\mathbf{D}$
2 Initialization of the coefficient matrices $\mathbf{E}_i^{00}$ to $\mathbf{E}_0^{12}$
3 Initialization of $\mathbf{F}_e$ and $\mathbf{K}_e$
4 Determination of number of Gauss points, corresponding location and weights in local direction of ξ and η
5 **begin** Loop over all Gauss points
6 Computation of N, dN, ddN
7 Computation of $\mathbf{r}_{,\xi}$, $\mathbf{r}_{,\eta}$ and $\mathbf{n}$
8 Computation of $\mathbf{n}_\xi$ and $\mathbf{n}_\eta$
9 Computation of Jacobian matrices $\mathbf{A}$ and $\mathbf{B}$
10 Computation of determinants of Jacobian matrices J_0 and J_1
11 Computation of $\mathbf{b}_0^i$ and $\mathbf{b}_1^i$ with $i = 1, 2, 3$
12 **begin** Loop over all element nodes
13 Construction of $\mathbf{B}_0^1$, $\mathbf{B}_0^2$, $\mathbf{B}_1^1$ and $\mathbf{B}_1^2$
14 Calculation of coefficient matrices $\mathbf{E}_i^{00}$ to $\mathbf{E}_0^{12}$
15 **end**
16 **end**
17 Sum up $\mathbf{E}_{01}^0 = \mathbf{E}_0^{10} + \mathbf{E}_1^{00}$, $\mathbf{E}_{01}^1 = \mathbf{E}_0^{11} + \mathbf{E}_1^{01}$ and $\mathbf{E}_{01}^2 = \mathbf{E}_0^{12} + \mathbf{E}_1^{02}$
18 **Call** algorithm for analytical solution in thickness direction (Algorithm 3)
19 Return: $\mathbf{K}_{s,e}$ and $\mathbf{F}_e$

Having ended the loop over all the elements supports are defined by the application of the Dirichlet boundary conditions and the solution of the problem, which is the last step of the processing.

The numerical implementation is mostly focused on the element routine which is described in Algorithm 2. This process computes the coefficient matrices $\mathbf{E}$ for an element, which are then taken to calculate the element stiffness matrix. It starts by the construction of the elasticity matrix and the initialization of all the necessary matrices. Then, the number of Gauss points and the location with weights is determined to start the Gauss point loop for the numerical computation of the integrals for the computation of the coefficient matrices. In this loop the basis functions and its derivatives are determined and also the derivatives of the position vector and the normal vector. Having these in hand, the Jacobian matrices $\mathbf{A}$ and $\mathbf{B}$ with the determinants can be computed and consequently the auxiliary matrices $\mathbf{b}_i$, too. A new interior loop over the element nodes follows, where the B-matrices and the coefficient matrices are determined for each node and are assembled. This is done for all the Gauss points and every node per Gauss point. When the loops ended the coefficient matrices are summed up so that six coefficient matrices $\mathbf{E}_0^{00}$, $\mathbf{E}_0^{01}$, $\mathbf{E}_0^{02}$, $\mathbf{E}_{01}^{0}$, $\mathbf{E}_{01}^{1}$ and $\mathbf{E}_{01}^{2}$ remain. With these, Algorithm 3 can be called to determine the element stiffness matrix and the internal load vector.

Algorithm 3 Analytical solution in thickness direction

Input:

Coefficient matrices $\mathbf{E}_0^{00}$, $\mathbf{E}_0^{01}$, $\mathbf{E}_0^{02}$, $\mathbf{E}_{01}^{0}$, $\mathbf{E}_{01}^{1}$ and $\mathbf{E}_{01}^{2}$
Thickness t
Number of nodes *enode*
Number of degrees of freedom per side *enodep*

Output:

Element stiffness matrix $\mathbf{K}_{s,e}$
Internal load vector $\mathbf{F}_e$

1 Set up of inverse of transformation matrix $\mathbf{T}^{-1}$
2 Set up of inverse-transposed of transformation matrix $\mathbf{T}^{-T}$
3 Calculate element stiffness matrix $\mathbf{K}_e$
4 Transform the element stiffness matrix into translational degrees of freedom only by $\mathbf{K}_{s,e} = \mathbf{T}^{-T}\mathbf{K}_e\mathbf{T}^{-1}$
5 Rearrange the stiffness matrix to the common denotation as $\mathbf{u} = [u_{1,b} \quad u_{1,t} \quad u_{2,b} \quad u_{2,t} \quad ...]^T$
6 Return: $\mathbf{K}_{s,e}$ and $\mathbf{F}_e$

Algorithm 3 conducts the analytical solution in thickness direction, where at first the inverse of the transformation matrix $\mathbf{T}^{\mathrm{T}}$ and the inverse-transposed of the transformation matrix $\mathbf{T}^{-\mathrm{T}}$ are constructed. Then the element stiffness matrix $\mathbf{K}_e$ is determined, which is up to this point considering translational and rotational degrees of freedom on the bottom surface. By multiplication of the transformation matrix, the degrees of freedom can be transformed to translational degrees of freedom only with nodes on bottom and top surface. The new element stiffness matrix is denoted as $\mathbf{K}_{s,e}$. It is important to rearrange the rows and columns of the stiffness matrix according to the standard arrangement so that the degrees of freedom are not separated in bottom nodes first and and afterwards top nodes. The algorithm returns the element stiffness matrix and the internal load vector, which is zero if body forces are neglected.

4.3. Postprocessing

Once the displacements of the nodes have been calculated, preprocessing can begin. In this process, the calculated data is extended by information that is helpful for comparability or clarity, such as support forces, stresses or similar. Afterwards the data is visualized in a further process. For this purpose, an output file is generated in which the information of the calculation is sorted and formatted. In this work ParaView is used to visualize the deformation of the structures. It is an open-source application for interactive, scientific visualization. Algorithm 4 shows how the output file is created in the code.

Algorithm 4 Visualization of the solution in Paraview

Input:

Knot vectors **U** and **V**
Weighted control points **KPW**
Number of elements per parametric direction *anzpu* and *anzpv*
Number of resolution points *resu*, *resv* and *resw*
Solution **d**

Output:

vtk-file for ParaView

1 Determine the number of export points
2 Determine the number of export cells
3 Calculate the step sizes *stepu*, *stepv* and *stepw*
4 Compute the coordinates **Coor** for each resolution point
5 Compute the connectivity matrix of the elements **Conn**
6 Compute the solution field in matrix format without weights **solmat**
7 Compute the solution field in matrix format with weights **solmatw**
8 Conduct the interpolation of the solution **solexp**
9 Create filename and open vtk-file
10 Write coordinates **Coor**, connectivity **Conn** and solution **solexp** in vtk-file
11 Close vtk-file

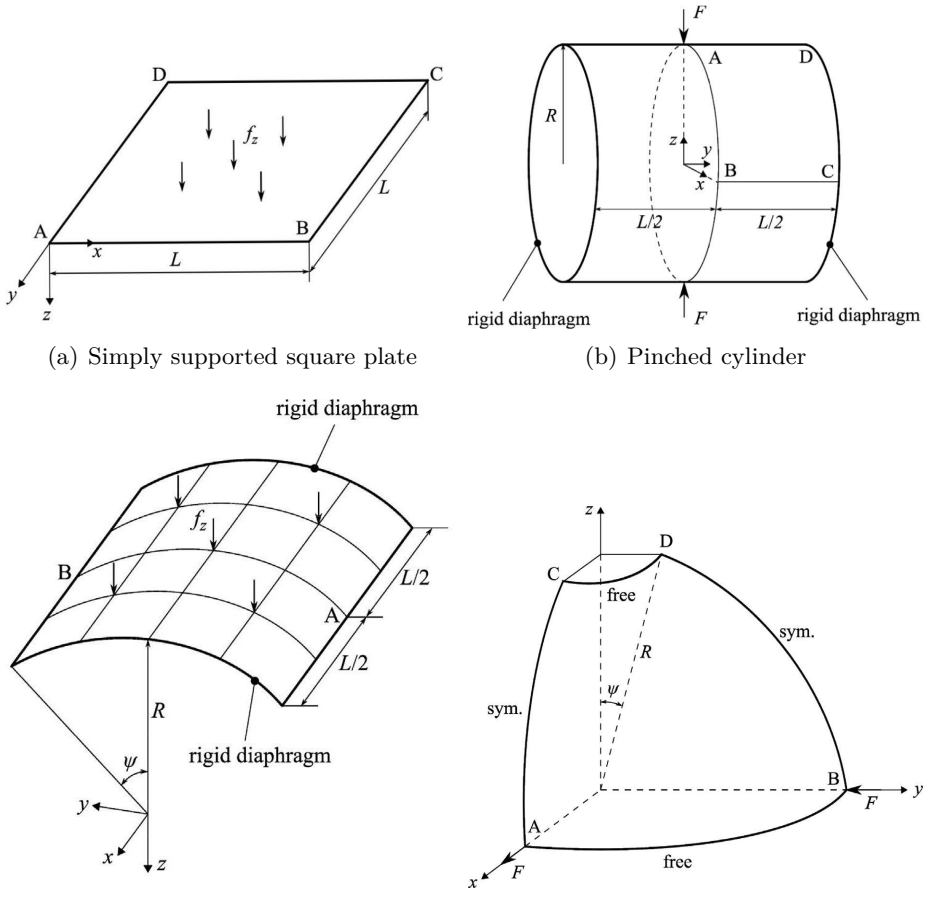
Algorithm 4 shows how the data for output file of ParaView is generated and saved. At first, the number of export points *np* and the number of export cells *nc* is determined, which defines later the set of coordinates and solution points. Then the step sizes in each parametric direction are calculated by dividing the difference of the end point and the starting point of the knot vector by the number of resolution points, where *resu* and *resv* are the number of resolution points in ξ and η and *resw* is the number of resolution points in thickness direction along γ. This provides the possibility to determine the coordinates **Coor** on the shell at all resolution points. In addition, the connectivity matrix **Conn** can be set up, which determines the meshing of the coordinates among each other. Next, the solution field without weights **solmat** and the solution field with weights **solmatw** are structured in a matrix format to have a similar structure as the weighted control points. To derive the displacements at each resolution point the interpolation of the solution field can be conducted which is a linear combination of the value of the basis functions at the resolution point and the corresponding deformation. Having derived all the necessary data, the vtk-file can be opened with a predefined filename most likely containing the order and the number of elements. Then, the specific data for ParaView is written in the file, which has a strict format to be readable. At first the coordinates **Coor** are saved followed by the connectivity **Conn**. Then, the number of nodes according to the number of cells *nc* and the cell type are written in the file and the displacement at the resolution points are saved. Finally

the vtk-file can be saved and closed. Thus the postprocessing is finished and the vtk-file can be evaluated in ParaView.

As a result, the numerical implementation of the isogeometric scaled boundary shell formulation was presented in this chapter, which calculates the static load-bearing behavior of thin shells. For this purpose, the IGA was combined with the element formulation of the scaled boundary method. In the following chapter, the performance of this formulation is examined and evaluated with respect to the accuracy and the calculation costs.

5. Benchmarking

In this chapter the presented shell formulation is tested on its reliability. Therefore, a series of benchmark tests is performed and the solution of the calculation at specific points is compared to reference solutions of several previous results from the literature as for example [1], [7], [4], [15] or [14]. A reasonable shell element must have the power to handle inextensional bending modes of deformation, rigid body motion without straining and complex membrane states of stress [1]. Furthermore, the obstacle course is reasonably short, not to evaluate the same behavior for several problems. In this work four test problems are presented and evaluated, that are shown in Figure 5.1.



(a) Simply supported square plate (b) Pinched cylinder

(c) Scordelis-Lo roof (d) Pinched hemisphere with hole

Figure 5.1: The evaluated benchmark problems for the SBIGA shell [14].

First problem is the simply supported square plate, which follows the previous work of [29] and [18], respectively, where an SBIGA plate with different formulation in thickness direction was derived and was the preliminary work for the SBIGA shell. It is especially important to evaluate the plates tendency to transverse shear locking. The second problem of the pinched cylinder is a more advanced model to check for the behavior of the shell element when it comes to inextensional bending modes and complex membrane states. Furthermore, the geometry can be easily investigated due to the constant curvature. The Scordelis-Lo roof is the third problem. In this test, almost exclusively membrane modes are evoked and decoupled from other states, contrary to the second problem. In the fourth model, the hemisphere with hole, the inextensional bending modes are evaluated, which form the main deformation in the test. And

although the inextensional bending modes are not as severe as in the pinched cylinder, the test is helpful in also testing the rigid body rotation as the load is applied in the normal direction to the shell. With these four problems in hand, the SBIGA shell is evaluated for its ability to reliable determine displacements of shells.

For all of the problems a convergence study is applied, and compared to the reference solution of the problem. For the sake of comparability, the accuracy of the results is determined by dividing the computed displacement u_{comp} by the reference displacement u_{ref} as

$$\text{accuracy} = \frac{u_{comp}}{u_{ref}}, \tag{5.1}$$

where a perfect accuracy is achieved for a value of 1. Furthermore, the relative error is determined as

$$\text{relative error} = \frac{|u_{comp} - u_{ref}|}{|u_{ref}|}. \tag{5.2}$$

Consequently, for a perfectly computed solution the relative error is 0.

5.1. Simply supported square plate

The first benchmark is a simply supported plate as described in [4] and depicted in Figure 5.2. The length of both sides is $L = 10.0$ m and varies in its thickness h. It consists of linear elastic material with Young's modulus $E = 1000.0$ Pa and Poisson's ratio $\nu = 0.3$. The model is evaluated by the increase of the polynomial order for all thicknesses and afterwards a mesh refinement for a patch using bi-quadratic elements ($p = q = 2$) is applied on a plate of thickness $h = 0.01$ m.

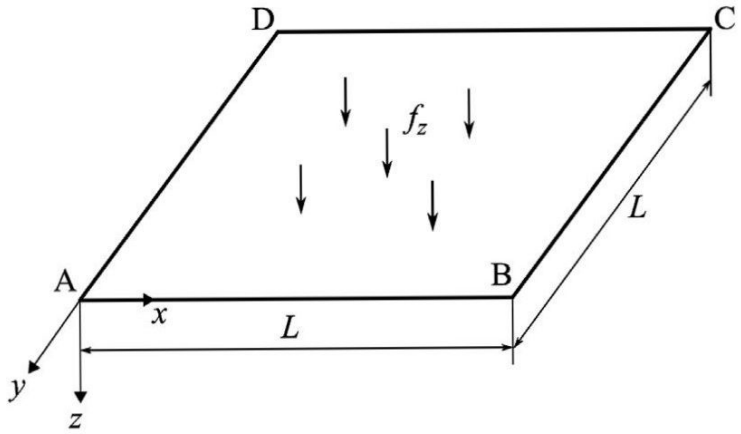


Figure 5.2: Sketch of the problem setup for the simply supported square plate [14].

The plate is subjected to a pressure load $f_z = 1.0 \cdot h^3$ which varies in relation to the thickness. This assumption is quite useful for the comparison of the results of various values of the thickness t considering the analytical Kirchhoff series solution based on the first two terms for the center deflection is $u_z = 0.442892$ m according to [27]. Equation (5.3) shows the solution of the first two terms of the analytical Kirchhoff series for the displacement in the middle of a simply supported square plate w_{max} where the second term already has such a low impact, that the other parts

of the series are neglected. Hence, with the flexural rigidity of the plate as D with $D = \frac{Eh^3}{12(1-\nu^2)}$ the displacement yields

$$w_{max} = \frac{5}{384}\frac{f_z L^4}{D} - \frac{4 f_z L^4}{\pi^5 D}(0.68562 - 0.00025 + ...) = 0.00406\frac{f_z L^4}{D}. \tag{5.3}$$

The plate is simulated with several thicknesses $h = [0.5, 0.2, 0.1, 0.001]$ m. Since the boundary conditions both Neumann b.c. and Dirichlet b.c. are applied symmetrically the model can be divided into a quarter of this plate with a side length of $L/2 = 5$ m. Therefore, the Dirichlet boundary conditions need to be readjusted to fulfill the symmetry conditions. The new supports are imposed as: $u_z = 0$ along edge AB and edge AD, $u_z = u_x = \theta_y = 0$ along edge BC and $u_z = u_y = \theta_x = 0$ along edge CD.

First, one element with various thicknesses is evaluated. For each thickness the order of the basis functions in both directions are increased from $p = q = 2$ (quadratic basis functions) to $p = q = 8$. The results of the order elevation are shown in Table 5.1.

Table 5.1: Deflection at the middle point of the simply supported plate in [m].

$p = q$	$h = 0.5$	$h = 0.2$	$h = 0.1$	$h = 0.01$	$h = 0.001$
2	0.4062	0.3910	0.3886	0.3878	0.3990
3	0.4596	0.4516	0.4505	0.4502	0.4444
4	0.4583	0.4459	0.4440	0.4434	0.4419
5	0.4619	0.4469	0.4445	0.4436	0.4464
6	0.4638	0.4478	0.4447	0.4436	0.4648
7	0.4658	0.4486	0.4450	0.4436	0.4398
8	0.4667	0.4494	0.4453	0.4436	0.4443
reference	0.4429	0.4429	0.4429	0.4429	0.4429

It shows, that the SBIGA shell can determine the deflection of the plate for all thicknesses in an adequate precision, even though for thicker plates of $h = 0.5$ m and $h = 0.2$ m the results are a little bit less accurate than for thin plates. On the overall, for basis functions of order $p = q = 3$ (cubic) or higher, the results are within a relative error of 4% for every thickness evaluated.

Additionally, the plate of thickness $h = 0.01$ m is evaluated on results for a mesh refinement by using bi-quadratic elements. The plate is evaluated with a refinement of [5, 10, 15, 20, 25, 30, 35, 40] elements per side and shown in Figure 5.3. It is important to notice that the degrees of freedom increase quadratically by increasing the number of elements per side. Therefore, the diagram is displayed in the logarithmic scale along the x-axis. However, the number of degrees of freedom is not calculated for patches in IGA as in FEM, because the elements overlap. The degrees of freedom DOF can be calculated as $DOF = (n_{el,\xi} + p) \cdot (n_{el,\eta} + q) \cdot 6$, where $n_{el,\xi}$ is the number of elements in ξ-direction and $n_{el,\eta}$ the number of elements in η-direction.

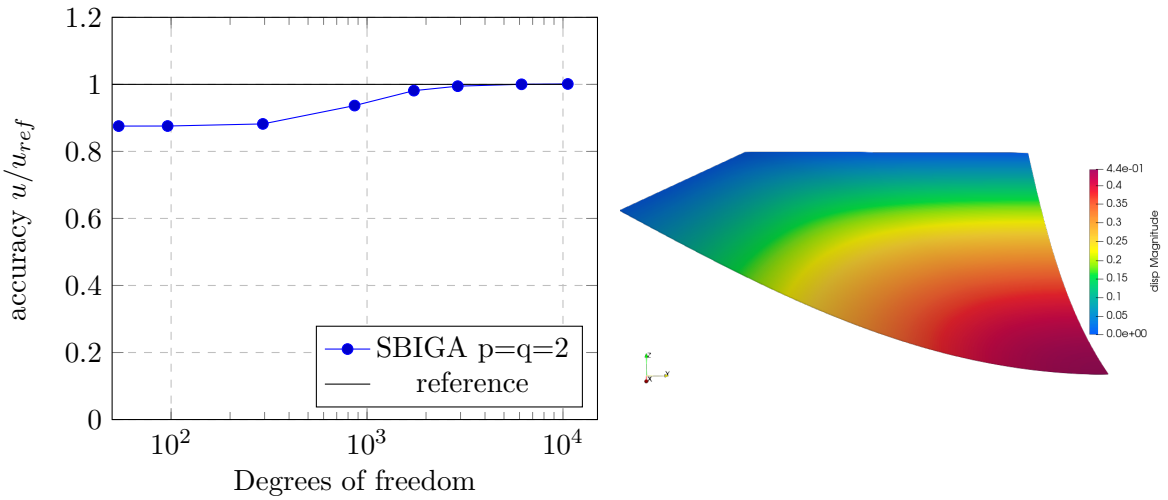


Figure 5.3: Convergence of the vertical displacement in the middle of the simply supported square plate for $h = 0.01$ m (left) and the deformed structure for 40 elements per side where the deformation is scaled by 4 (right).

The convergence diagram shows that even for rough meshes the SBIGA shell can properly determine the displacement at the middle of the plate with an accuracy of approximately 90%. For finer meshes the computed displacements converge towards the reference solution and for a mesh with 15 elements per side, the computed solution has already an accuracy of at least 98 %. Also the model of the deformed structure reveals that the maximal displacement is at the edge, where the symmetry axis intersect, which is the middle of the simply supported plate.

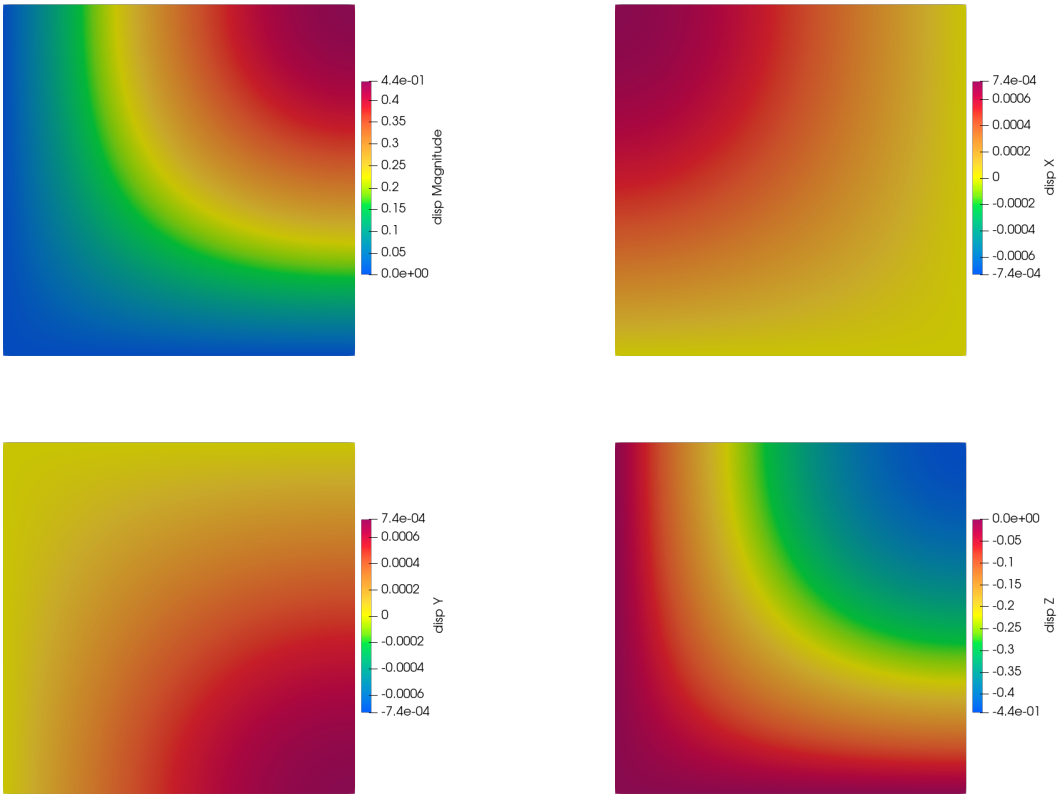


Figure 5.4: Results of the computed displacements for the simply supported square plate using 40 elements per side and a thickness of $h = 0.01$ m separated in magnitude (upper left), displacement in x (upper right), displacement in y (lower left) and displacement in z (lower right).

The displacement plots are separated according to the global axis and shown in Figure 5.4. The plots visualize that there are no displacements, where the boundary conditions are applied as they are zero in z-direction along the left and lower side, in x-direction on the right side due to symmetry conditions and in y direction on the upper side due to the same reason.

It is shown that SBIGA can calculate the simply supported plate very precisely and has already sufficient accuracy for bi-cubic basis functions with one element. But not only for higher order basis functions, but also for quadratic elements with a refined mesh, the displacements can be determined accurately.

5.2. Pinched cylinder

The second benchmark is a cylinder that is pinched at the top and bottom by two opposite and equal single loads at its mid span. The cylinder is supported by a rigid diaphragm at the ends. The dimensions of the cylinder are length $L = 600$ m, radius $R = 300$ m and $h = 3$ m with the elastic material properties of Young's modulus $E = 3 \cdot 10^6$ Pa and $\nu = 0.3$. The corresponding load is $F = 1$ N. The reference solution is the vertical displacement at point A (the point of the upper single load) and is determined as $u_{ref} = -1.8248 \cdot 10^{-5}$ m [1]. This model is the first model considering curvature in the shell and the most sophisticated since it does not only evoke inextensional bending modes but also membrane stresses. It is of special interest since the curvature κ can be evaluated as it is constant with $\kappa = -1/R$ in the x-z-plane.

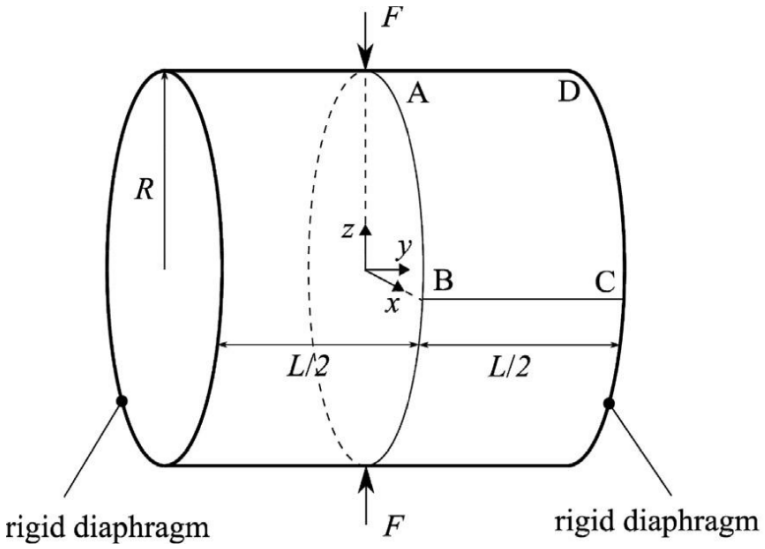


Figure 5.5: Sketch of the problem setup for the pinched cylinder [14].

Since the problem setup is double symmetric, the model is simplified as an eighth of the cylinder where the length is decreased to the half and only the upper cylinder in between the points A, B, C and D is discretized. The boundaries then need to be adjusted as $u_{y,b} = u_{y,t} = 0$ along the edge of AB, $u_{z,b} = u_{z,t} = 0$ along the edge of BC, $u_{z,b} = u_{z,t} = 0$, $u_{x,b} = u_{x,t} = u_{y,b} = u_{y,t} = 0$ along the edge of CD and $u_{x,b} = u_{x,t} = 0$ along the edge of AD. Consequently, the single load at point A is adjusted to F= 1/4 N and for a better numerical stability, the load is applied to the top and bottom node in A by the value of $F/2$ each. The initial mesh of the cylinder is of quadratic basis functions with $p = q = 2$. The control points and weights are denoted in the Appendix B. The model is evaluated by conducting a mesh refinement and afterwards an order elevation.

At first as explained above, the curvature of the model can be evaluated since for a circle the curvature κ at any point is $\kappa = -1/R$ and yields the possibility to check the geometric description of the cylinder. Hence, it was analyzed for its curvature in the x-z-plane along ξ. The curvature of the cylinder can be determined by

$$\kappa(\xi) = \frac{x_{,\xi}\, z_{,\xi\xi} - x_{,\xi\xi}\, z_{,\xi}}{((x_{,\xi})^2 + (z_{,\xi})^2)^{3/2}}. \tag{5.4}$$

The curvature was checked for each Gauss point on every element for the cylinder and the calculation showed that by the analytical solution of the derivatives as well as by the method of finite differences the curvature was always calculated to $\kappa = -1/R$. The same applied for the derivative of the normal vector which was compared for both methods and yielded the same results. For this reason, in the further course, the analytical approach was always used for the computation of the derivatives, both for the derivatives of the basis functions and the normal vector.

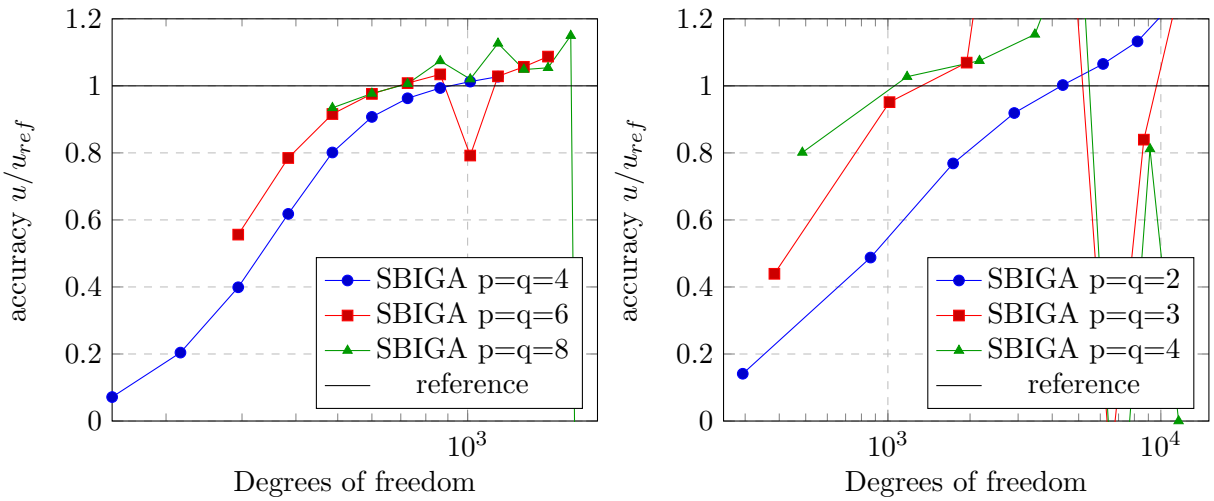


Figure 5.6: Convergence study by mesh refinement of the vertical displacement at point A (point of load application) of the pinched cylinder for 1 to 10 elements and orders of $p = q = 4$, $p = q = 6$ and $p = q = 8$ (left) and for 5 to 40 elements and orders of $p = q = 2$, $p = q = 3$ and $p = q = 4$ (right).

Figure 5.6 shows the diagram of the convergence study of the SBIGA for the pinched cylinder. The mesh refinement is conducted in two different ways. For higher orders the mesh refinement is conducted from 1 to 10 elements each direction of orders $p = q = 4$ ($n_{gp} = 8$), $p = q = 6$ ($n_{gp} = 10$) and $p = q = 8$ ($n_{gp} = 12$). It can be seen that for higher orders the solution can be predicted quite accurately even for a few elements where for example for bi-octic elements already the initial mesh reaches an accuracy of 94%. It is noticeable, however, that the solution for more bi-octic elements no longer shows a straight line, but increases and decreases abruptly and ends up on a opposite deflection for 10 elements per side. Also for the bi-sextic elements the solution deviates strongly for 7 elements per side. Moreover, the curve seems to converge in between 4 and 8 elements per side towards a slightly higher value than the reference solution, but rises again with more elements. For bi-quartic elements no abrupt changes appear and the convergence takes longer to reach the reference solution. For 8 elements per side the computed solutions has a relative error of 0.6% and for 10 elements per side it is 2.7%.

On the right side of the figure the convergence study for [5, 10, 15, 20, 25, 30, 35, 40] elements per side and orders of $p = q = 2$ ($n_{gp} = 6$), $p = q = 3$ ($n_{gp} = 7$) and $p = q = 4$ ($n_{gp} = 8$) is

shown. It is clear that no curve converges. In addition, it is recognizable that similar to the results of higher order for fine meshes, the results sometimes deviate strongly even stronger than for higher orders.

Table 5.2: Properties and solution of the order elevation for the pinched cylinder.

$p = q$	DOFs	number of Gauss points	calculation time	solution $[10^{-5}\,m]$
2	180	6	11 s	-0.0151
3	252	7	27 s	-0.0512
4	336	8	78 s	-1.0944
5	432	9	183 s	-1.5868
6	540	10	397 s	-1.7633
7	660	11	803 s	-1.8275
8	792	12	1634 s	-1.8542

The order elevation is conducted by using 4×3 elements to discretize the cylinder and increasing the order of the elements from $p = q = 2$ to $p = q = 8$ continuously. Table 5.2 shows the important properties of the calculation. The number of Gauss points is always three points more than necessary per direction on an element for the exact solution in common FEA and sufficiently accurate for the solution of the problem. The calculation time increases non-linearly and a calculation takes 27 minutes at the highest order. The solutions for $p = q = 7$ and $p = q = 8$ are precise with a relative error of 0.15 % and 1.61 %, respectively. Unfavorably, the relative error is higher for $p = q = 8$.

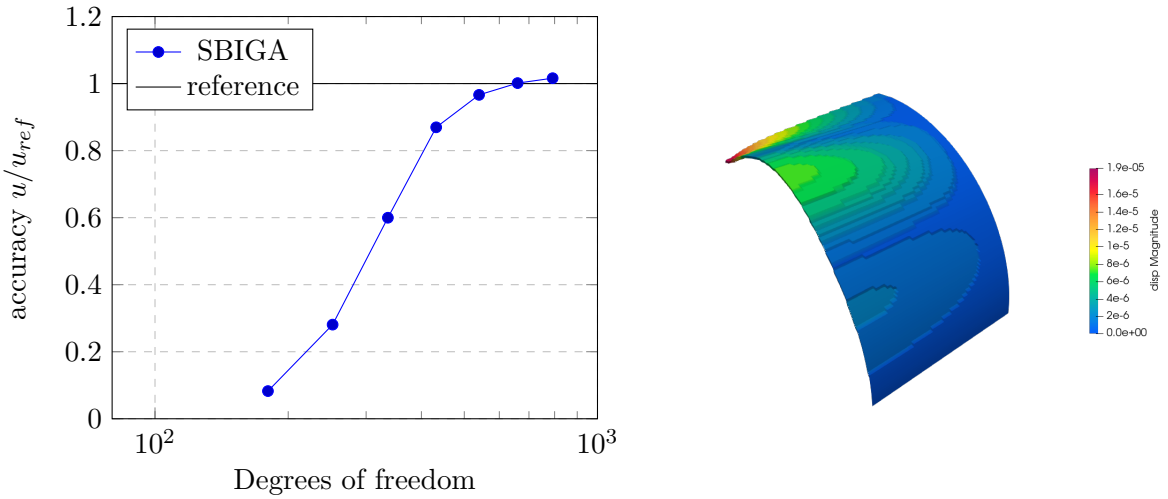


Figure 5.7: Convergence study by order elevation of the vertical displacement at point A (point of load application) of the pinched cylinder (left) and the deformed structure for 4×3 elements per side and basis functions of order $p = q = 8$, where the deformation is scaled by $2 \cdot 10^6$ (right).

The convergence study in Figure 5.7 shows that the SBIGA converges towards the reference solution and slightly overestimates the displacement at the load application. However, a good

convergence can be observed. When it comes the displacements of the cylinder, the deformed structure shown in the figure on the right has the typical buckling around the point of the load application which decreases towards the rigid diaphragm that can be observed also in other results [7].

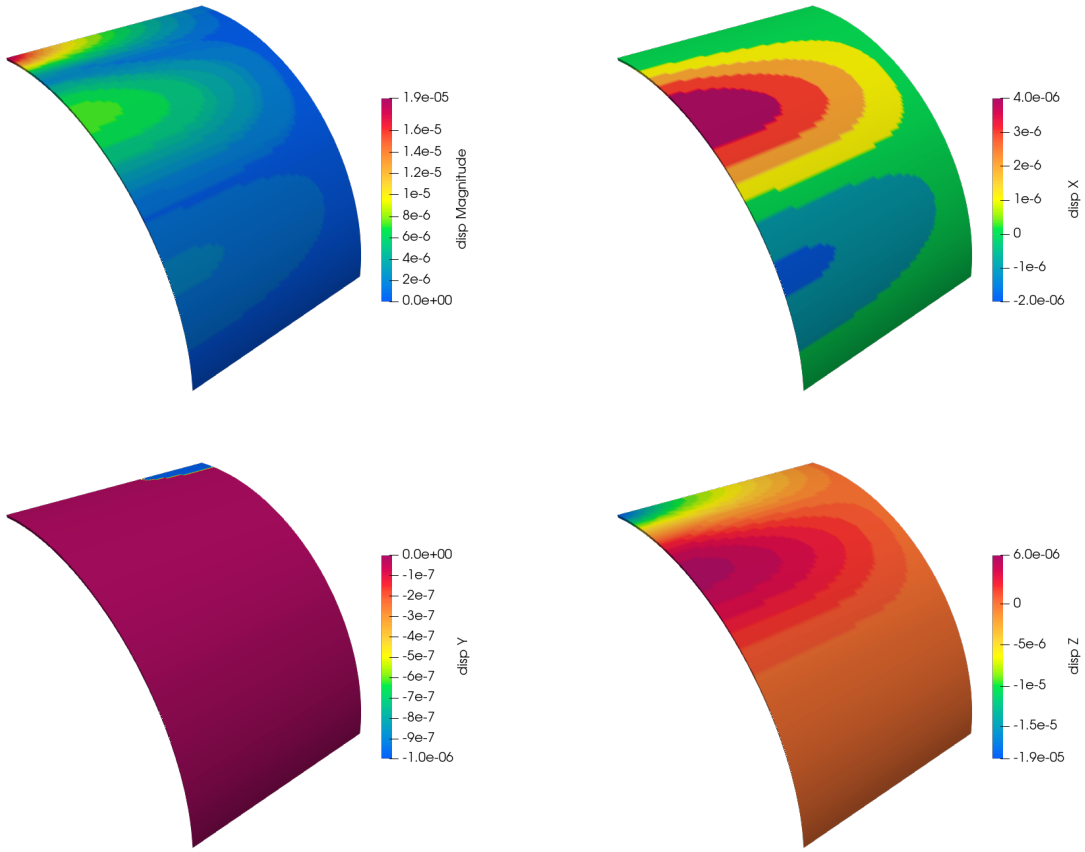


Figure 5.8: Results of the computed displacements for the pinched cylinder for 4×3 elements per side of order $p = q = 8$ separated in magnitude (upper left), displacement in x (upper right), displacement in y (lower left) and displacement in z (lower right).

Figure 5.8 shows the displacement plots for the calculation of the patch with 4×3 elements per side of an order $p = q = 8$. The plot of magnitude shows that the largest displacements occur at the point of load application. In addition, a bulge can be seen slightly away from the load application. From the plots of displacement in the x-direction and z-direction, it can be seen that the displacement at the load application is purely vertical displacement (which is consistent with the boundary conditions), while the buckling is dominated by horizontal displacement in the x-direction. The displacements in y-direction are negligible.

Finally, the convergence studies of the pinched cylinder show that the SBIGA cannot calculate the displacements of the model with sufficient accuracy for any mesh and order. The observations on the cylinder suggest that an order increase is much more likely to improve the solution than a mesh refinement. Since the load application is likely to evoke singularities an overshooting of the deflection does not necessarily mean that the formulation is erroneous, however, the relative error is quite high to only suggest this to be due to singularities. Furthermore it could be shown via the curvature that the geometric model of the cylinder including first and second derivatives can be determined correctly and the isogeometric framework is providing an accurate geometric description.

5.3. Scordelis-Lo roof

Next, the Scordelis-Lo roof [22] is investigated. The shell is a cylinder cutout of 80°. It is vertically loaded by a dead load and the curved edges are supported by rigid diaphragms same as the pinched cylinder in Section 5.2. A sketch of the problem setup is shown in Figure 5.9. The corresponding parameters of the model are length $L = 50$ m, radius $R = 25$ m, thickness $h = 0.25$ m angle of cylinder cutout $\psi = 40°$, Young's modulus $E = 4.32 \cdot 10^8$ Pa, Poisson's ration $\nu = 0$ and surface load $f_z = 90$ Pa. The investigated displacement is the vertical displacement at the middle of the free edge at point A with a reference solution of $u_{ref} = 0.03024$ m [15]. The problem mainly evokes membrane stresses, which are to be investigated.

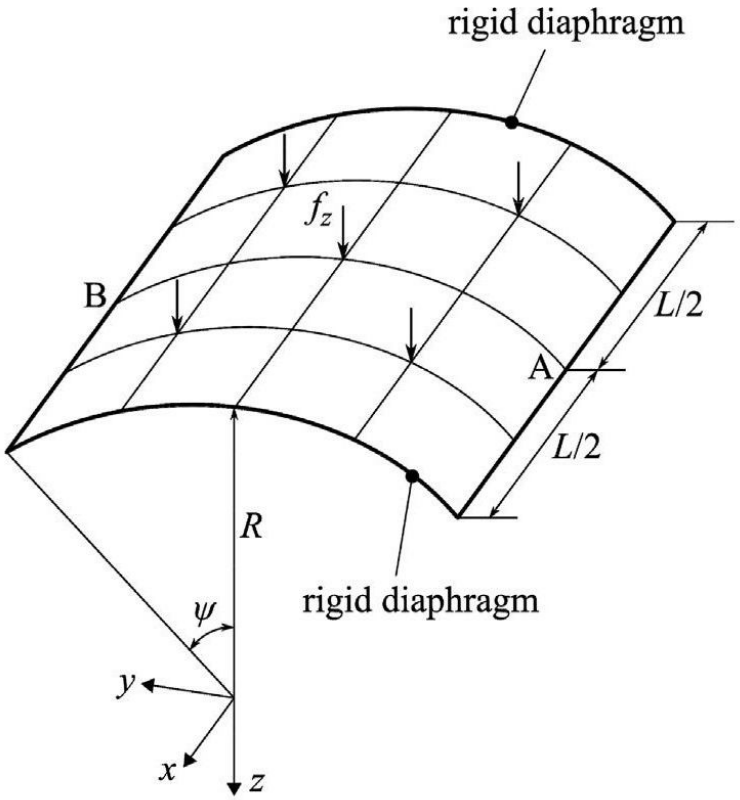


Figure 5.9: Sketch of the problem setup for the Sordelis-Lo roof [14].

Since the shell is symmetric, only a half of the shell is modeled. Therefore, it is cut at $L/2$ from point A to point B. The boundary conditions then are $u_{y,b} = u_{z,b} = 0$ at the rigid diaphragm and $u_{z,b} = u_{z,t} = 0$ at the edge of AB. The straight edges are free. The initial mesh of the roof is modeled using quadratic basis functions, $p = q = 2$. The control points are reported in Appendix B. Two convergence studies are conducted by a mesh refinement first and afterwards an order elevation.

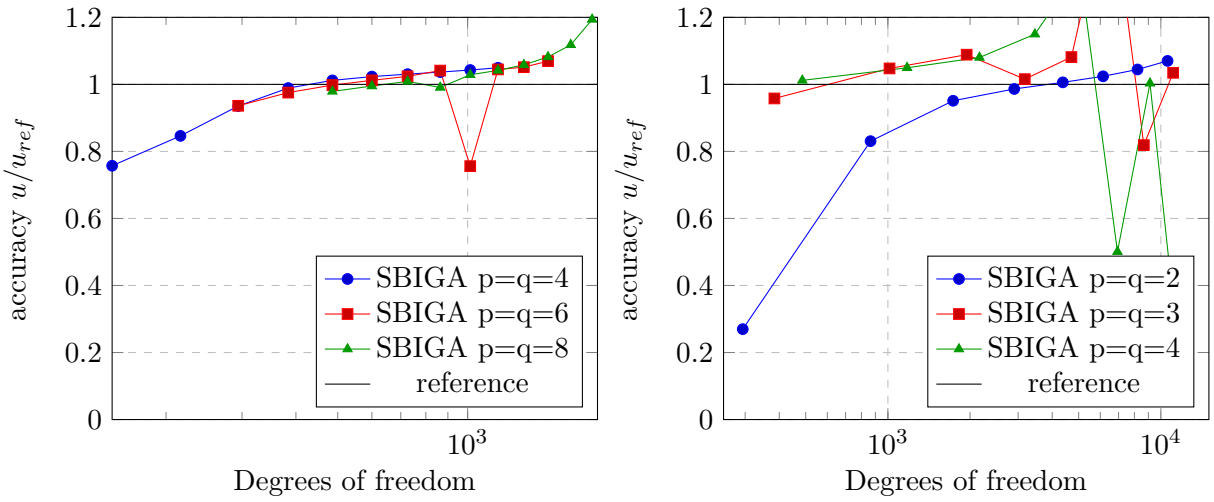


Figure 5.10: Convergence study by mesh refinement of the vertical displacement in the middle of the straight edge at point A of the Scordelis-Lo roof for 1 to 10 elements and orders of $p = q = 4$, $p = q = 6$ and $p = q = 8$ (left) and for 5 to 40 elements and orders of $p = q = 2$, $p = q = 3$ and $p = q = 4$ (right).

The convergence study by mesh refinement for the Scordelis-Lo roof is shown in 5.10. On the left the mesh refinement is conducted for 1 to 10 elements with orders of $p = q = 4$, $p = q = 6$ and $p = q = 8$. All curves show a similar convergence behavior with a slight overshooting of the computed solution. As also with the pinched cylinder, abrupt changes are recognizable with the convergence for $p = q = 6$ and $p = q = 8$, whereby particularly with $p = q = 6$ the change is very strong. For finer meshes of more than 7 elements the solution of the bi-octic elements increases strongly.

On the right the convergence study for a mesh refinement with [5, 10, 15, 20, 25, 30, 35, 40] elements per side and orders of $p = q = 2$, $p = q = 3$ and $p = q = 4$ is presented. It can be observed, that no curve converges towards any limit. Furthermore, the curves of order $p = q = 3$ and $p = q = 4$ heavily change for fine meshes and there is no common prediction of the solution visible.

Table 5.3: Properties and solution of the order elevation for the Scordelis-Lo roof.

$p = q$	DOFs	number of Gauss points	calculation time	solution [m]
2	96	6	8 s	-0.0220
3	150	7	14 s	-0.2012
4	216	8	31 s	-0.2559
5	294	9	67 s	-0.2879
6	384	10	143 s	-0.2950
7	486	11	282 s	-0.2991
8	600	12	548 s	-0.3009

Next, an order elevation is conducted using 2×2 elements for the discretization and increasing the order from $p = q = 2$ to $p = q = 8$. Table 5.3 lists the properties related to the calculations. The

number of Gauss points per direction are increased from 6 to 12 points which also increases the calculation time. While a bi-quadratic element with 6 Gauss points takes 8 seconds to calculate, an element of order $p = q = 8$ with 12 Gauss points takes 9 minutes. The computed solutions are tending towards the reference solution where the displacement of an order of $p = q = 8$ yields $u_{comp} = -0.03009\,m$ which is a relative error of 0.46% and therefore very accurate.

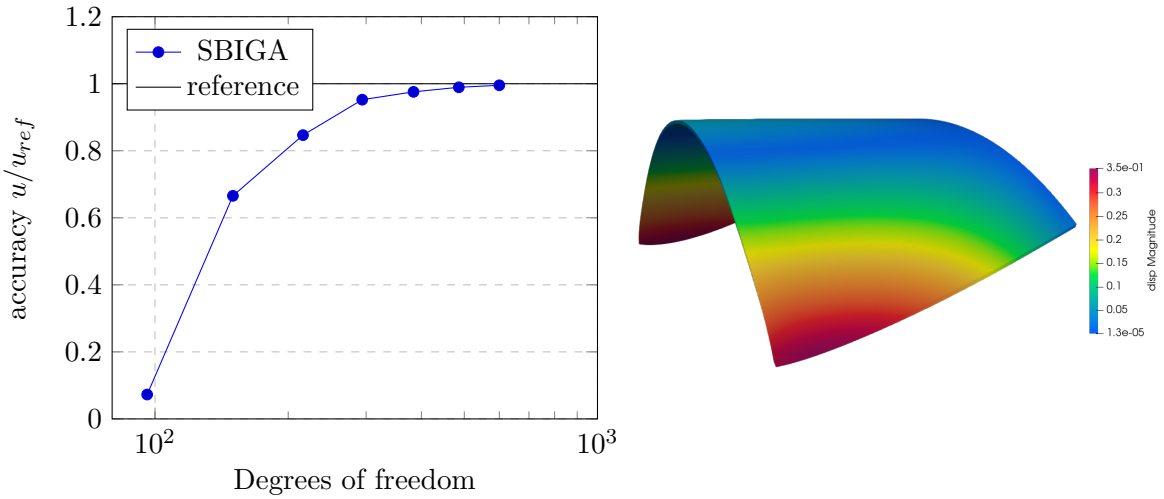


Figure 5.11: Convergence study by order elevation of the vertical displacement in the middle of the straight edge at point A (left) and the deformed structure for 2×2 elements of order $p = q = 8$ where the deformation is scaled by 20 (right).

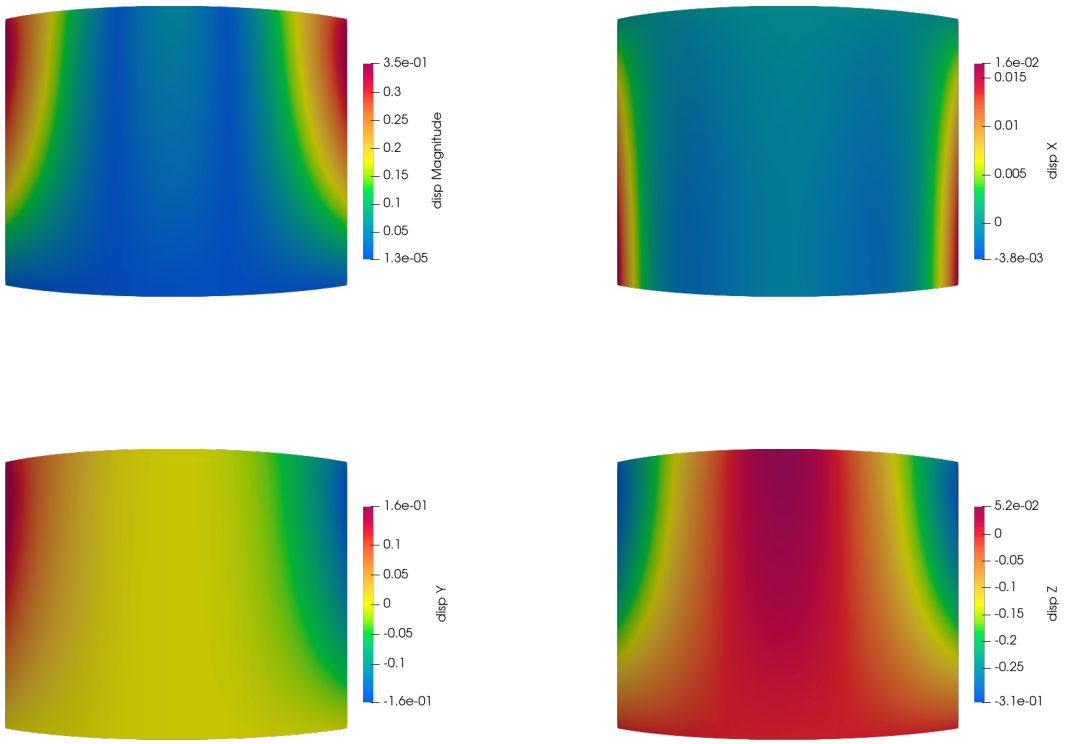


Figure 5.12: Results of the computed displacements from top view for the Scordelis-Lo roof using 2×2 elements of order $p = q = 8$ separated in magnitude (upper left), displacement in x (upper right), displacement in y (lower left) and displacement in z (lower right).

Figure 5.11 shows the convergence study for the order elevation of the model with 2×2 elements where the order is increased from $p = q = 2$ to $p = q = 8$ continuously. On the right, the deformed model at 2×2 elements and order $p = q = 8$ is shown with a scale of 20. It shows the half of the Scordelis-Lo roof (symmetry), where the surface load causes the free, straight sides to yield slightly inward and thus move downward. It also shows the membrane dominated behavior.

The plots of the displacements in Figure 5.12 show separated displacement components. It is visible that the above described boundary conditions are fulfilled, where the displacement in x is zero on the upper curved edge and the displacement in y and z is set to zero for the lower curved edge.

Similarly to the problem of the pinched cylinder, the results show that for a small number of elements with a sufficient high order of basis functions the solution can be computed to high accuracy. However, for very fine meshes, the stiffness of the roof is underestimated and the deformations are overestimated. It is especially noticeable that the solution does not converge. In contrary, for sufficiently rough meshes the solution converges in the vicinity of the reference solution. It is much more advantageous to increase the order instead of refining the mesh.

5.4. Pinched hemisphere with hole

The last problem is the pinched hemisphere with hole described in [1]. The hemisphere has a cutout of a hole on top in an angle of $\psi = 18°$. The other parameters are radius $R = 10$ m, Young's modulus $E = 6.825 \cdot 10^7$ Pa and $\nu = 0.3$. The model is evaluated for the thickness and corresponding force amplitudes of $h = 0.04$ m with $F = 1.0$ N and $h = 0.004$ m with $F = 1.0 \cdot 10^{-3}$ N. For both versions the radial displacement at point A is evaluated with a reference solution of $u_{ref} = 0.0930$ m [23]. It is a bending dominated problem including rigid body rotation due to the load application.

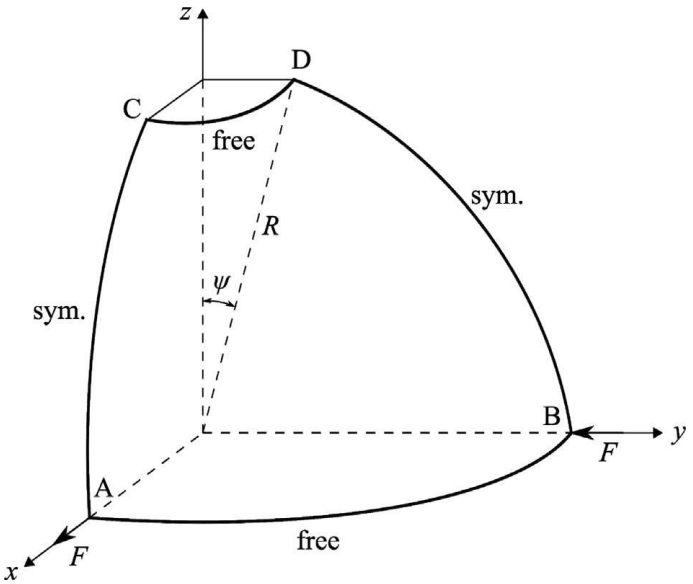


Figure 5.13: Sketch of the problem setup for the pinched hemisphere with hole [14].

Due to symmetry, only one quarter of the hemisphere is modeled and the boundary conditions are adjusted. The new Dirichlet boundary conditions as shown in Figure 5.13 are $u_{y,b} = u_{y,t} = 0$ along the edge of AC and $u_{x,b} = u_{x,t} = 0$ along the edge of BD. The edges AB and CD are

free. To prevent movements in z-directions, additionally point C is supported in z-direction. The initial mesh uses quadratic basis functions, $p = q = 2$ and the control points are denoted in the Appendix B. The model is evaluated by conducting a mesh refinement for both thicknesses with corresponding load amplitudes. Additionally an order elevation is proceeded for a constant mesh.

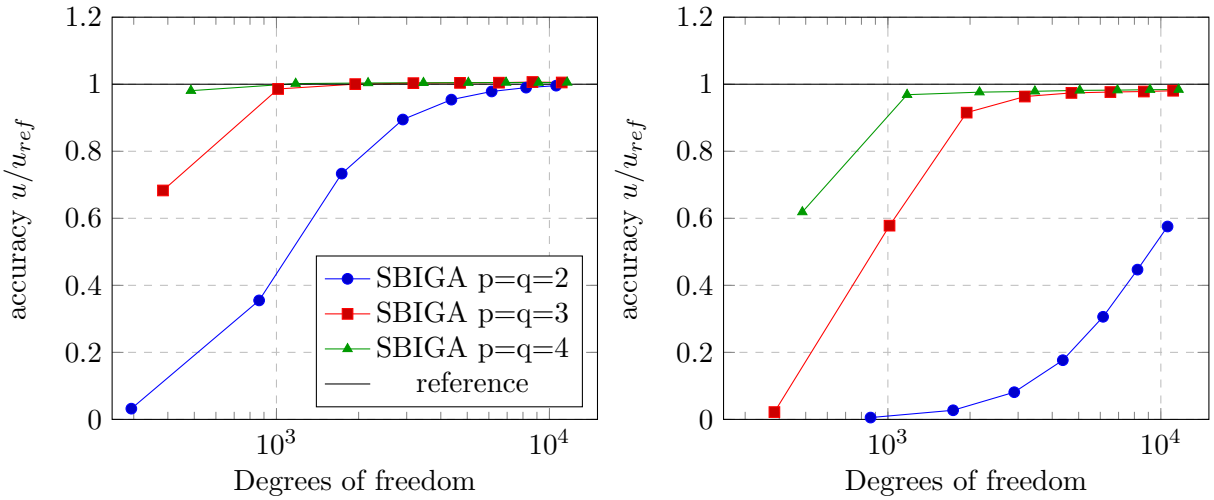


Figure 5.14: Convergence study by mesh refinement of the radial displacement at point A for $h = 0.04$ m (left) and $h = 0.004$ m (right) for elements of bi-quadratic, bi-cubic and bi-quartic order, where the legend applies to both diagrams.

The convergence study of the mesh refinement for [5, 10, 15, 20, 25, 30, 35, 40] elements per side is conducted with the orders of $p = q = 2$, $p = q = 3$ and $p = q = 4$. On the left side the convergence of the thick shell ($h = 0.04$ m) shows a very accurate and smooth convergence behavior for all orders. Especially for the bi-quartic order the solution is precise for all considered spatial discretizations with a relative error of 2%. Even for fine meshes the results are accurate and do not change. So is the accuracy of the computed displacement for the bi-quadratic elements with 40 elements per side 99.6%.

When it comes to the convergence study of the shell of $h = 0.004$ m the solution does not converge as rapidly as the thick shell and does not converge for bi-quadratic elements even for 40 elements per side. However, for higher orders the solution converges and slightly underestimates the displacements.

Table 5.4: Properties and solution of the order elevation for the pinched hemisphere with hole.

$p = q$	DOFs	number of Gauss points	calculation time	$h = 0.04$ [m]	$h = 0.004$ [m]
2	150	6	10 s	0.0004	0.0000
3	216	7	26 s	0.0104	0.0001
4	294	8	59 s	0.0764	0.0045
5	384	9	156 s	0.0907	0.0501
6	486	10	319 s	0.0922	0.0868
7	600	11	681 s	0.0929	0.0901
8	216	12	1222 s	0.0932	0.0905

The order elevation for the pinched hemisphere with hole is conducted with a mesh of 3×3 elements and the order is elevated from $p = q = 2$ to $p = q = 8$. The important properties of the order elevation are listed in Table 5.4 with degrees of freedom, number of Gauss points, calculation time and the computed results of both models of $h = 0.04$ m and $h = 0.004$ m. The calculation time was similar for both calculations. Therefore, the calculation time of the thin shell was chosen. The results show that for the order of $p = q = 8$ both models can be calculated accurately with a relative error of 0.22% for the thick shell and a slight underestimation of the displacement for the thin shell of 2.69%.

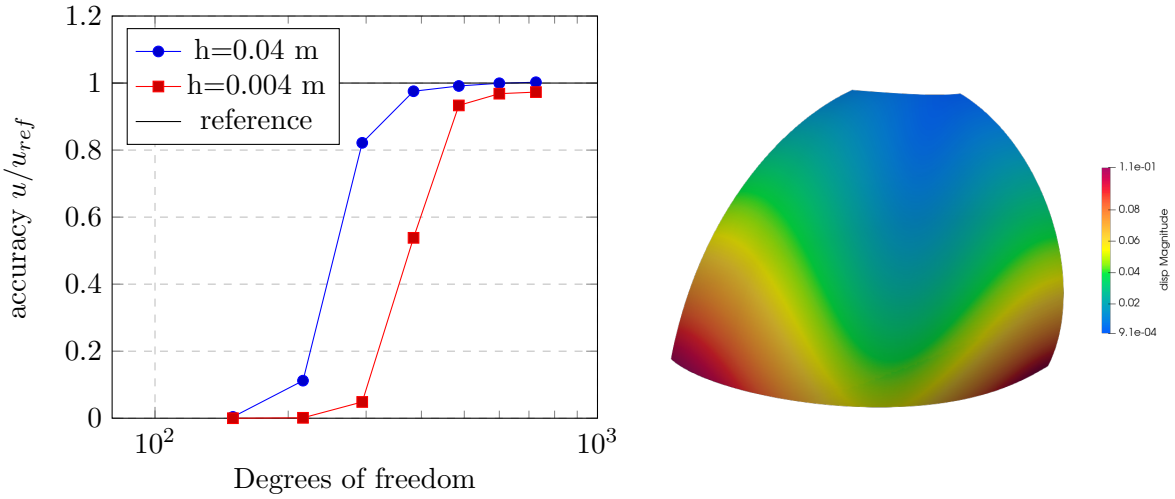


Figure 5.15: Convergence study by order elevation of the radial displacement at point A for $h = 0.04$ m and $h = 0.004$ m (left) and the deformed structure of thickness $h = 0.04$ m for 3×3 elements of order $p = q = 8$ where the deformation is scaled by 20 (right).

Having a look on the convergence study for the order elevation in Figure 5.15 shows that the thick shell converges very smooth towards the reference solution and already for an order $p = q = 5$ the solution has an accuracy of over 97%. The thin shell also converges towards the reference solution even though it slightly underestimates the displacements. Furthermore, the solution does not converge as rapidly as the thick shell. These results also match the SBFEM results of [14].The deformed structure shows both rigid body rotation for the whole structure and bending modes especially at the bottom corners of the hemisphere.

The displacement plots of Figure 5.16 display for the magnitude of the displacement, that the SBIGA shell can handle the loads applied in normal direction which is visible at the bottom corners. Furthermore, the plots of the displacement in x-direction and y-direction show that the loads are applied anti-symmetrically since the resulting displacements are working in opposite directions. Finally the plot of the displacement in z-direction has a perfectly symmetric deformation which legitimizes the application of a single support in z-direction.

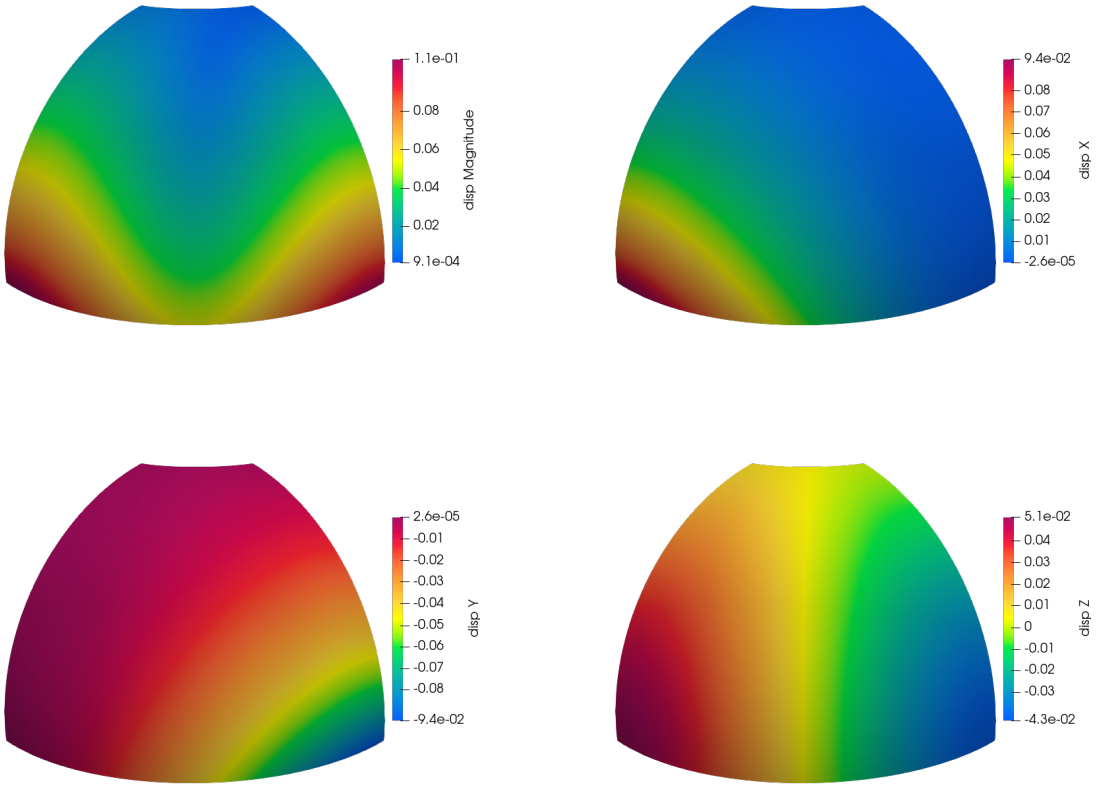


Figure 5.16: Results of the computed displacements for the pinched hemisphere with hole of thickness $h = 0.04$ m using 3×3 elements and order of $p = q = 8$ separated in magnitude (upper left), displacement in x (upper right), displacement in y (lower left) and displacement in z (lower right).

Finally, for the problem of the pinched hemisphere with hole, it can be observed that, unlike the other problems with curved shells, for the pinched hemisphere also a mesh refinement leads to an accurate calculation of the solution. Since the results do not converge fast for bi-quadratic elements (especially for the thin shell) it is recommended to use higher order basis functions.

6. Summary & Outlook

6.1. Summary

This thesis has aimed at deriving and implementing an isogeometric scaled boundary formulation. For this purpose, the SBFEM shell formulation was combined with the IGA. First the mathematical background of the IGA was explained and its advantages compared to the conventional FEM were described. Afterwards the element description with the SBM was executed and integrated into the IGA so that an SBIGA formulation was derived. The formulation was then implemented numerically in MATLAB and the key processes and algorithms were visualized.

Afterwards, the formulation was evaluated by benchmark tests and checked for its reliability. For this purpose, four problems were chosen with the simply supprted square plate, the pinched cylinder, the Scordelis-Lo roof and the pinched hemisphere with hole. For the simply supported square plate the computed solution was very accurately determining the displacements in the middle of the plate and also an order elevation yielded accurate solutions. This shows that the solution for flat structures is reliable. For the pinched cylinder the calculations for the mesh refinement do not converge towards any limit. Possibly the singularity of the load might evoke higher displacements at the point of the load application. However, this needs to be evaluated further to identify the problem. When performing an order elevation for coarse meshes the computed solution is quite accurate and converges towards the reference solution. In addition, it could be shown that the isogeometric framework works and is suitable to be coupled with the SBM using the example of the pinched cylinder. A similar behavior of the convergence study is observed for the Scordelis-Lo roof which is subjected to a surface load distributed over the whole shell structure. For the mesh refinement the computed solution does not converge towards any limit in particular for low order elements and fine meshes. For higher orders and mesh refinement of more coarse meshes the computed solution converges at first to then increase substantially, which seems to be accompanied by numerical instabilities. For an order elevation, again, the calculation yield proper results and converges very smoothly. For the pinched hemisphere with hole, the results for the order elevation and the mesh refinement both converge perfectly. It shows that for slender shells it takes more degrees of freedom to properly determine the displacements, however for higher orders or finer meshes the results are calculated correctly. Only the mesh refinement for bi-quadratic elements does not converge. However, for an even finer mesh the solution is expected to get closer to the reference solution.

Concerning the problems of the mesh refinement, several checks were made. The IGA framework was checked, where the calculation of the curvature showed that the basis functions and their derivatives were correctly determined. In addition, the analytical and numerical calculation reached almost the same values (taking into account that the numerical solution can be determined only approximately). In addition, the derivative of the normal vector was determined numerically and analytically, which also gave consistent results. By consultation with Jianghuai Li (Ph.D.), the author of [14], also no problems in the description of the element routine for the SBM could be identified.

Finally, in this work an isogeometric scaled boundary formulation could be derived, which was also tested in benchmarks for its reliability. Its efficiency and capabilities could be shown, although problems are still visible. Thus, the problems of convergence studies for very fine meshes have to be verified and evaluated. It is remarkable that these deviations did not occur for the plate, which is not curved, and for the hemisphere, which is curved in all planes, and caused convergence problems only for the examples with a single curvature. Nevertheless, in

this work a good basis for further investigations and implementations was prepared, which has a wide potential.

6.2. Outlook

As explained above, the isogeometric scaled boundary formulation and the algorithm provided with this work yields a wide range of further investigations which will be sketched out herein.

At first, for a further use of this formulation it is necessary to check on the problems that occur for the mesh refinement. Several things were already checked and seem to work properly. However, some further checks can be done as for example the analytical solution in thickness direction. These reviews should focus particularly on the examples of the pinched cylinder and the Scordelis-Lo roof, as these examples have given rise to problems.

Similar to the thesis of [29], different approaches for the solution in thickness direction can be derived. So the analytical solution can be substituted by a semi-analytical method, a collocation method or the weak formulation in thickness direction. This would also help to check the solution reported in this thesis.

For a more accurate and faster performance of the simulation, it is possible to change the numerical integration over the element. So far the Gauss-Legendre quadrature rule was used and the number of points was increased by three. Since NURBS are rational the the Gauss-Legendre quadrature is not exact but is sufficiently precise for an increased number of points. However, for a more efficient quadrature the calculation costs might be significantly decreased. A method presented in [10] is the so called half-point rule which is explicitly derived for NURBS in isogeometric analysis.

Furthermore, the material model, which was assumed to be linear elastic can be extended to an electroactive polymer (EAP). These are polymers that perform a change of size or shape when subjected to an electric field and are used for the design of artificial muscles in robotics or for gripping devices [17]. These electroactive grippers are closed in the undeformed state and open up under voltage. Figure 6.1 shows a gripper for all states of a working cycle.

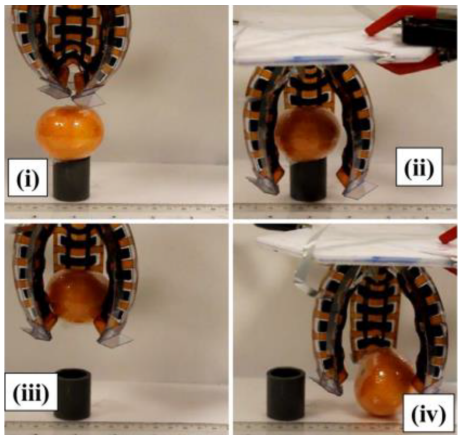


Figure 6.1: Working principle of an electroactive gripper [28].

REFERENCES

[1] Belytschko, T., Stolarski, H., Liu, W. K., Carpenter, N., and Ong, J. S. Stress projection for membrane and shear locking in shell finite elements. *Computer Methods in Applied Mechanics and Engineering 51*, 1-3 (1985), 221–258.

[2] Cottrell, J. A., Hughes, T. J., and Bazilevs, Y. *Isogeometric Analysis: Toward Integration of CAD and FEA*. 2009.

[3] Deeks, A. J., and Wolf, J. P. A virtual work derivation of the scaled boundary finite-element method for elastostatics. *Computational Mechanics 28*, 6 (2002), 489–504.

[4] Echter, R., Oesterle, B., and Bischoff, M. A hierarchic family of isogeometric shell finite elements. *Computer Methods in Applied Mechanics and Engineering 254* (2013), 170–180.

[5] Hardwick, M. F., Clay, R. L., Boggs, P. T., Walsh, E. J., Larzelere, A. R., and Altshuler, A. DART System analysis. Tech. rep., Sandia National Laboratories, Albuquerque, 2005.

[6] Heinze GmbH (30.04.2021). https://www.heinze.de/architekturobjekt/kuwait-international-airport-terminal-2/12810995/.

[7] Hu, Q., Xia, Y., Natarajan, S., Zilian, A., Hu, P., and Bordas, S. P. Isogeometric analysis of thin Reissner–Mindlin shells: locking phenomena and B-bar method. *Computational Mechanics 65*, 5 (2020).

[8] Hughes, T. J. Isogeometric Analysis: introduction and overview.

[9] Hughes, T. J., Cottrell, J. A., and Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering 194*, 39-41 (2005), 4135–4195.

[10] Hughes, T. J., Reali, A., and Sangalli, G. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering 199*, 5-8 (2010).

[11] Kiendl, J., Bletzinger, K. U., Linhard, J., and Wüchner, R. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering 198*, 49-52 (2009), 3902–3914.

[12] Kiendl, J. M. Isogeometric Analysis and Shape Optimal Design of Shell Structures. 140.

[13] Li, J., Shi, Z., and Liu, L. A scaled boundary finite element method for static and dynamic analyses of cylindrical shells. *Engineering Analysis with Boundary Elements 98*, August 2018 (2019), 217–231.

[14] Li, J., Shi, Z., Liu, L., and Song, C. An efficient scaled boundary finite element method for transient vibro-acoustic analysis of plates and shells. *Computers and Structures 231* (2020), 106211.

[15] Macneal, R. H., and Harder, R. L. A proposed standard set of problems to test finite element accuracy. *Finite Elements in Analysis and Design 1*, 1 (1985).

[16] Man, H., Song, C., Gao, W., and Tin-Loi, F. A unified 3D-based technique for plate bending analysis using scaled boundary finite element method. *International Journal for Numerical Methods in Engineering 91*, 5 (2012).

[17] Man, H., Song, C., Gao, W., and Tin-Loi, F. Semi-analytical analysis for piezoelectric plate using the scaled boundary finite-element method. *Computers and Structures 137* (2014), 47–62.

[18] Man, H., Song, C., Xiang, T., Gao, W., and Tin-Loi, F. High-order plate bending analysis based on the scaled boundary finite element method. *International Journal for Numerical Methods in Engineering 95*, 4 (2013).

[19] Moore, H. About MATLAB®. In *MATLAB® for Engineers*, no. 3. 2011.

[20] Piegl, L., and Tiller, W. The NURBS book, 2nd edition, 1997.

[21] Rogers, D. F. *An Introduction to NURBS: With Historical Perspective*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[22] Scordelis, A. C., and Lo, K. S. Computer Analysis of Cylindrical Shells. *ACI Journal Proceedings 61*, 5 (1964).

[23] Simo, J. C., Fox, D. D., and Rifai, M. S. On a stress resultant geometrically exact shell model. Part II: The linear theory; Computational aspects. *Computer Methods in Applied Mechanics and Engineering 73*, 1 (1989), 53–92.

[24] SMART Engineering GmbH (30.04.2021). https://www.smart-fem.de/nx-nastran/.

[25] Song, C. *The scaled boundary finite element method : introduction to theory and implementation*. 2018.

[26] Song, C., and Wolf, J. P. The scaled boundary finite-element method - Alias consistent infinitesimal finite-element cell method - For elastodynamics. *Computer Methods in Applied Mechanics and Engineering 147*, 3-4 (1997), 329–355.

[27] Timoshenko, S. P. Theory of Plates and Shells, 1941.

[28] Youn, J. H., Jeong, S. M., Hwang, G., Kim, H., Hyeon, K., Park, J., and Kyung, K. U. Dielectric elastomer actuator for soft robotics applications and challenges. *Applied Sciences (Switzerland) 10*, 2 (2020).

[29] Zhang, D. *Numerical formulations for the thickness direction of Isogeometric Scaled-Boundary plates and shells*. Master thesis, 2019.

A. Appendix A

The derivative of the unit normal vector is about to be derived, since its derivative needs to be determined for the Jacobian matrix of curved elements. In the following, the equation used in the paper is derived in detail and the important steps are explained for the partial derivative with respect to ξ. The unit normal vector is derived by the cross product of the tangent vectors on the local axes ξ and η [12]. Therefore, to determine its derivative, it is useful to start with

$$\frac{\partial \mathbf{n}}{\partial \xi} = \frac{\partial}{\partial \xi} \frac{\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|}. \tag{A.1}$$

Applying the quotient rule on the equation leads to

$$\begin{aligned} \frac{\partial \mathbf{n}}{\partial \xi} &= \frac{\frac{\partial}{\partial \xi}\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta} \cdot \|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\| - \mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta} \cdot \frac{\partial}{\partial \xi}\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|^2} \\ &= \frac{\frac{\partial}{\partial \xi}(\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta})}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|} - \frac{(\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}) \cdot \frac{\partial}{\partial \xi}\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|^2}. \end{aligned} \tag{A.2}$$

At this point, there are basically two terms that can be considered separately for the derivation. So for the first term the product rule will be applied and for second term the norm needs to be derived. In general, the partial derivative of a norm of a function is defined as

$$\frac{\partial \|f(x)\|}{\partial x} = \frac{f(x)}{\|f(x)\|} \cdot \frac{\partial f(x)}{\partial x} \tag{A.3}$$

So then, applying the product rule on the first term of equation (A.2) and the derivative of the norm from equation (A.3) into the second term, the equation yields

$$\mathbf{n}_{,\xi} = \frac{\frac{\partial \mathbf{r}_{,\xi}}{\partial \xi} \times \mathbf{r}_{,\eta} + \mathbf{r}_{,\xi} \times \frac{\partial \mathbf{r}_{,\xi}}{\partial \xi}}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|} - \frac{(\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta})[(\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}) \cdot (\frac{\partial \mathbf{r}_{,\xi}}{\partial \xi} \times \mathbf{r}_{,\eta} + \frac{\mathbf{r}_{,\eta}}{\partial \xi} \times \mathbf{r}_{,\xi})]}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|^3}. \tag{A.4}$$

Consequently, the same procedure can be done for the derivative of $\mathbf{n}$ with respect to η as

$$\mathbf{n}_{,\eta} = \frac{\frac{\partial \mathbf{r}_{,\xi}}{\partial \eta} \times \mathbf{r}_{,\eta} + \mathbf{r}_{,\xi} \times \frac{\partial \mathbf{r}_{,\xi}}{\partial \eta}}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|} - \frac{(\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta})[(\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}) \cdot (\frac{\partial \mathbf{r}_{,\xi}}{\partial \eta} \times \mathbf{r}_{,\eta} + \frac{\mathbf{r}_{,\eta}}{\partial \eta} \times \mathbf{r}_{,\xi})]}{\|\mathbf{r}_{,\xi} \times \mathbf{r}_{,\eta}\|^3}. \tag{A.5}$$

With these equations in hand the derivative of the unit normal vector can be determined, since the second derivatives of the tangent vectors with respect to the local axes can be calculated as described in Section 2.1.6 and thus, all variables are known.

B. Appendix B

This appendix lists the necessary control points to represent the geometries from the benchmarking using the IGA. The control points are chosen so that all lengths and radii are equal to one. Thus, the models can be freely dimensioned by the input and can be described flexibly.

The cylinder (see Section 5.2) consists basically of the combination of a quarter circle and a straight line. Consequently, most of the control points are interpolatory except for (1,2), (2,2) and (3,2). The initial mesh is of bi-quadratic order and has 3×3 control points, which are denoted in Table B.1.

Table B.1: Control points and weights for the initial mesh of the cylinder.

(i,j)	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
x	0	1	1	0	1	1	0	1	1
y	0	0	0	0.5	0.5	0.5	1	1	1
z	1	1	0	1	1	0	1	1	0
w	1	$1/\sqrt{2}$	1	1	$1/\sqrt{2}$	1	1	$1/\sqrt{2}$	1

The initial mesh of the Scordelis-Lo roof (see Section 5.3) is listed in Table B.2. It consists of bi-quadratic order and 3×3 control points. Similarly as for the cylinder, the initial mesh is a combination of an 80° cutout of a circle and a straight line.

Table B.2: Control points and weights for the initial mesh of the Scordelis-Lo roof.

(i,j)	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
x	0	0	0	0.5	0.5	0.5	1	1	1
y	-0.643	0	0.643	-0.643	0	0.643	-0.643	0	0.643
z	0.766	1.305	0.766	0.766	1.305	0.766	1.305	1	0.766
w	1	0.766	1	1	0.766	1	1	0.766	1

Table B.3 lists the control points of the hemisphere with hole (see Section 5.4). It is a quarter of a hemisphere with a hole on to of 18°. The initial mesh consists of 3×3 control points and bi-quadratic order.

Table B.3: Control points and weights for the initial mesh of the hemisphere with hole.

(i,j)	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
x	1	1	0.309	1	1	0.309	0	0	0
y	0	0	0	1	1	0.309	1	1	0.309
z	0	0.727	0.951	0	0.727	0.951	0	0.727	0.951
w	1	0.809	1	$1/\sqrt{2}$	0.572	$1/\sqrt{2}$	1	0.809	1

STATUTORY DECLARATION

I hereby declare in lieu of an oath that I have completed the present Master thesis entitled "Development and implementation of an isogeometric scaled boundary shell formulation" independently and without illegitimate assistance from third parties. I have used no other than the specified sources and aids. In case that the thesis is additionally submitted in an electronic format, I declare that the written and electronic versions are fully identical. The thesis has not been submitted to any examination body in this, or similar, form.

Mathias Reichle, 345942 Aachen, May 5, 2021