TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Statik

Isogeometric Analysis and Shape Optimal Design of Shell Structures

Josef M. Kiendl

Vollständiger Abdruck der von der Fakultät für Bauingenieur- und Vermessungswesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:
Univ.-Prof. Dr.-Ing. Gerhard H. Müller

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Kai-Uwe Bletzinger
2. Univ.-Prof. Dr. rer. nat. Ernst Rank
3. Univ.-Prof. Dr. Thomas J.R. Hughes, Univ. of Texas at Austin, USA

Die Dissertation wurde am 23.11.2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingnieur- und Vermessungswesen am 14.03.2011 angenommen.

Isogeometric Analysis and Shape Optimal Design of Shell Structures

Abstract

Isogeometric analysis is a new method of computational analysis with the goal of merging design and analysis into one model by using a unified geometric representation. NURBS (Non-Uniform Rational B-Splines) are the most widespread technology in today's CAD modeling tools and therefore are adopted as basis functions for analysis.

In this thesis, the isogeometric concept is applied to the analysis and shape optimization of shell structures. A new, rotation-free shell element is developed, using the Kirchhoff-Love shell theory and NURBS as basis functions. NURBS-based analysis provides advantages especially for shells, since the structural behavior of a shell is mainly determined by its geometry and therefore a good geometric description is essential. Furthermore, due to the exact geometry description with NURBS, curvatures can be evaluated directly on the surface without rotational degrees of freedom or nodal directors.

Different examples show the good performance and accuracy of the method, for geometrically linear and nonlinear problems. Aspects concerning boundary conditions and the treatment of multiple patch structures are investigated, and solutions are proposed which allow the use of this method for a broad variety of problems. Furthermore, the developed shell formulation proves as very well suited for a direct integration into a CAD model, which is also realized in a commercial CAD software. The practical application of this integrated method for different examples also reveals problems and limitations of the present approach, which are discussed subsequently. Another goal of this thesis is to extend the isogeometric concept to shape optimization. After a brief review of shape optimization using CAD-based or FE-based design models, isogeometric shape optimization is introduced as a combination of both existing approaches which enhances flexibility in choosing the design space.

In the context of a cooperation project, the developed structural formulation is integrated into a fluid-structure interaction (FSI) environment and is applied to the three-dimensional FSI simulation of a wind turbine blade rotating in the air flow. This example shows the relevance of this method to large industrial applications.

Isogeometrische Analyse und Formoptimierung von Schalen

Zusammenfassung
Isogeometrische Analyse ist ein neuer Ansatz für computergestützte Berechnungsverfahren, mit dem Ziel, Entwurf und Berechnung durch eine gemeinsame geometrische Darstellung in ein gemeinsames Modell zusammenzuführen. Die am weitesten verbreitete Technologie in heutigen CAD Systemen sind NURBS (Non-Uniform Rational B-Splines). Sie werden daher als Ansatzfunktionen für das Berechnungsmodell übernommen.

Im Rahmen der vorliegenden Arbeit wird dieses Konzept für die Berechnung und Formoptimierung von Schalen angewandt. Es wird ein neues, rotationsfreies Schalenelement nach der Kirchhoff-Love Schalentheorie mit NURBS als Ansatzfunktionen entwickelt. Der Einsatz von NURBS für die Berechnung zeigt sich speziell für Schalen von Vorteil, da das Tragverhalten einer Schale vornehmlich durch ihre Geometrie bestimmt wird und somit eine gute Geometriebeschreibung von großer Bedeutung ist. Des Weiteren ermöglicht die exakte Geometriebeschreibung mit NURBS die Berechnung von Krümmungen direkt auf der Fläche, wodurch auf Rotationsfreiheitsgrade und Knotendirektoren verzichtet werden kann.

In verschiedenen Beispielen wird die Zuverlässigkeit und Genauigkeit dieser Methode für geometrisch lineare sowie nichtlineare Probleme gezeigt. Es werden verschiedene Aspekte bezüglich Randbedingungen sowie das Modellieren von Strukturen, welche aus mehreren Flächen bestehen, untersucht und passende Lösungsmethoden entwickelt, welche die Anwendung dieser Methode für eine breite Vielfalt von Strukturen ermöglichen. Des Weiteren erweist sich das entwickelte Schalenmodell als sehr geeignet für die direkte Integration in ein CAD Modell, was mittels eines kommerziellen CAD Programms auch verwirklicht wird. Durch den praktischen Einsatz dieses integrierten Modells für verschiedene Beispiele zeigen sich ferner die Grenzen und Probleme dieses Ansatzes, welche im Anschluss diskutiert werden. Ein weiterer Arbeitspunkt ist es, das isogeometrische Konzept auf Formoptimierung zu erweitern. Nach einem Überblick über Formoptimierung mit CAD-basierten oder FE-basierten Methoden wird isogeometrische Formoptimierung als eine Kombination dieser beiden Methoden vorgestellt, die eine weitaus größere Flexibilität bei der Wahl des Entwurfsraums gestattet.

Das entwickelte Strukturmodell wird im Zusammenhang eines Kooperationsprojektes in ein Programm für Fluid-Struktur-Interaktion (FSI) implementiert und für die Berechnung eines in der Windströmung rotierenden Windturbinenblattes eingesetzt. Dieses Beispiel verdeutlicht die Relevanz dieser Methode für industrielle Anwendungen.

Acknowledgments

This dissertation was written from 2007 to 2011 during my time as research scholar at the Chair of Structural Analysis (Lehrstuhl für Statik) at the Technische Universität München, Munich, Germany. I would like to thank sincerely Prof. Dr.-Ing. Kai-Uwe Bletzinger for giving me the possibility to work in his research group and for his helpful guidance as my doctoral supervisor. I also want to express my thanks to Dr.-Ing. Roland Wüchner. As Project Team Leader of my research team, he also guided and supervised me throughout the whole project.

Furthermore, I would like to address my thanks to the members of my examining jury, Univ.-Prof. Dr. rer. nat. Ernst Rank and Prof. Dr. Thomas J.R. Hughes. Their interest in my work is gratefully appreciated. Also, I want to thank Prof. Dr.-Ing. habil. Gerhard Müller for chairing the jury.

In the course of my research project, I spent three months at the University of California, San Diego (UCSD) at the institute of Prof. Yuri Bazilevs. I want to thank him for his kind hospitality and the good cooperation that we continued also afterwards.

The funding for my whole work as research scholar was granted by the International Graduate School of Science and Engineering (IGSSE). This funding is gratefully acknowledged.

Many thanks are addressed to David Eames for proofreading my thesis.

I want to thank all coworkers at the Chair of Structural Analysis for the friendly cooperation and the pleasant time that I had working with them.

Finally, I want to thank my family and my dear girlfriend Julia for their help and support at all times.

Munich, April 2011 Josef Kiendl

Contents

Chapter 1

Introduction

It is a basic principle in engineering science to describe the objects in nature by reduced and simplified models. A main aspect in this model building is the dimensional reduction. While every object in reality is a three-dimensional body, it might be described by a model of reduced dimensionality, according to its extensions. Objects that are significantly larger in one dimension than the other two, are described by one-dimensional models, such as trusses and beams. If only one dimension is much smaller than the other two, two-dimensional models are employed, such as plates, membranes, and shells. A shell is a thin-walled structure with arbitrary curvature in the three-dimensional space. Both plates and membranes can be considered as special cases of shells. A plate is a plane shell, whereas a membrane is a shell that can carry tangential forces only. With increasing dimensionality, the models become mathematically more complicated, i.e. a plate is more complex than beams and trusses, and a shell is more complex than a plate. Therefore, it might seem somewhat surprising that a fully three-dimensional continuum model, which has even more complex stress and strain states, is usually less difficult to describe than a shell model. This is because for a three-dimensional solid, the general formula of continuum mechanics can be applied without regard to its shape, while for a shell an appropriate mathematical description of its geometry and of geometrical properties like curvatures is necessary. There is a long history of shell theories and new developments are still being made even today [YSMK00, BWBR04]. The need for shell theories lies in the fact that shell structures are ubiquitous in nature and technology. A shell carries the load “through its shape” and thus is very efficient in saving material and weight. It is due to the curvature that transversal loads can be carried by tension and compression, while bending moments are minimized. This load carrying behavior allows a very efficient use of the material. This effect was already understood by the Romans, who built large circular cupolas, for example in the Pantheon. Nowadays, shells are used everywhere, where a minimization of weight is important, e.g. in the aerospace and automotive industry. In civil engineering and architecture they are used to span large distances, e.g. large roofs and domes, and for aesthetic reasons. Slender structures with a smoothly curved shape are perceived as being elegant and natural.

The first shell theory goes back to Gustav R. Kirchhoff, who in 1850 developed the first plate theory [Kir50]. The Kirchhoff plate theory is also called the "classical plate theory", and it is the basis for different tables for plates (Czerny, Stiglat-Wippel, etc.), that are still used today in civil engineering. August E.H. Love developed a shell theory [Lov88] that is based on Kirchhoff's assumptions, and therefore it is known as the Kirchhoff-Love shell theory. A second prominent shell theory is the Reissner-Mindlin theory which, in contrast to the Kirchhoff-Love theory, takes into account transverse shear deformations [Rei45]. These are negligible for thin shells but can be of importance for thick shells. The limit between thick and thin is defined by the shell's slenderness which is defined as the ratio of curvature radius over thickness. A thick shell is defined by a slenderness of $R/t < 20$. Although most shells in practical applications can be classified as thin shells, the Reissner-Mindlin theory plays a dominant role in shell analysis using the finite element method.

The finite element method [ZTZ05, Hug00], which is a computer-based method, had its origins in the early 1960s and is nowadays the predominating method in structural analysis. While the solutions from classical plate and shell theory are restricted to regular shapes like, for example, rectangular plates or rotationally symmetric shells, arbitrary structures can be analyzed by the finite element method. It decomposes the structural domain into many small, "finite" elements with simple shapes. These elements are defined by a set of nodal points, which are connected by basis functions. Linear polynomials are the most used basis functions, due to their simplicity and the versatility of the resulting elements. The problem is that with such elements there is usually no continuity higher than C^0 between elements possible. Even with higher order polynomials it is not possible to guarantee C^1 continuity for arbitrarily shaped elements. This limitation prohibits a straight-forward use of the Kirchhoff-Love shell theory since it requires integrability of second derivatives and therefore at least C^1 continuity. The Reissner-Mindlin shell theory, where rotations and deflections are treated as two independent fields, requires only C^0 continuity. Therefore, most shell element formulations are based on this theory, independent of the distinction between thick and thin shells. The problem is that especially for thin shells, these elements suffer from various locking phenomena which are mostly attributed to the use of low-order basis functions. Locking is a typical problem for shell elements, and huge effort has been devoted and many different methods have been developed to prevent or at least reduce locking effects [Gel88, Kab92, BBR00, CdSNJFVAA02, Bra98, L.Y98, Kos04, BS92a, AB97, SB82, BS92b, Bis99].

As mentioned above, the introduction of the finite element method made it possible to analyze and therefore design structures of arbitrary shape for which no solutions had existed before. The design of such free-from shapes by mathematical methods is another field of science, termed CAD. CAD is the abbreviation of Computer Aided Design and had its origins in the late 1960s, i.e. a bit later than the upcoming of the finite element method which is a part of Computer Aided Engineering, in short CAE. The first method to construct free-form curves and surfaces was developed by Pierre Bézier at Renault in

1966 [Béz66]. Actually, another French engineer, Paul de Casteljau at Citroën developed the same technology some years earlier, but his work was never published and therefore the name of Beziér was attributed. A further development to Beziér's method were B-Splines which provide more flexibility in the modeling of free-form curves and surfaces. Rational techniques had already been applied to Beziér curves, which allows the exact description of all conic sections, such as ellipses and circles. The development of non-uniform rational B-Splines (NURBS) provided a technology that can exactly describe circular shapes (cylinders, spheres, etc.) which are basic elements in geometric modeling, but also allows very flexible modeling of free-form surfaces. For this reason, NURBS have become the standard technique in CAD modeling until today.

These historic remarks show that the methods in CAE and CAD were developed separately and as a consequence they use completely different methods to describe the same structural object. The design needs to be transformed into an analysis-suitable geometry, i.e. a finite element mesh. Although mesh generation to a large extent is performed automatically, it still requires manual control and adjustments by the engineer. In order to obtain reliable results, a series of analyses with different mesh refinements has to be performed. The fact that every FE mesh is only an approximation to the original geometry means that a finer mesh has to be completely rebuilt from the CAD geometry. The creation of an analysis-suitable geometry from the CAD geometry is nowadays the bottleneck for large engineering computations, and mesh creation is one part of this model transfer. A new form of analysis, called isogeometric analysis, tries to close this gap between the CAD and CAE world such that both disciplines work on the same geometric models.

The term "isogeometric analysis" was coined by Hughes et al. in 2005 [HCB05]. The basic idea is to avoid the meshing process by adopting the geometry description from design for analysis. As mentioned above, NURBS are the most widespread technology in CAD programs and thus, they are used as basis functions for analysis. It has been shown that they fulfill the necessary conditions for basis functions and therefore, analysis can be performed on the NURBS model without geometry conversion. Since then, NURBS-based isogeometric analysis has gained increasing interest and it has been applied with great success to the study of solids, shells, fluids, fluid-structure interaction, turbulence, and structural optimization [BCHZ08, BH08, HBC+10, EBCH08, CRBH06, BCZH06, WFC08, KBLW09, KBH+10, BBHH10, CH09, HCC10].

In this thesis, the isogeometric concept is applied for shell analysis and shape optimization. Firstly, NURBS as basis for geometric modeling are studied profoundly as well as their use for isogeometric analysis. A new isogeometric element, a shell element using Kirchhoff-Love kinematics and NURBS as basis functions, is developed. It is formulated geometrically nonlinear so that it is applicable for large deformations. Different benchmark examples are performed to test this element formulation. Special aspects concerning boundary conditions and the modeling with multiple NURBS patches are discussed, and solutions are presented and tested. After a thorough testing of the presented method, it is shown how it can be used to integrate isogeometric analysis into a commercial CAD software. Furthermore, the concept of isogeometric analysis is extended to isogeometric shape optimization. After a review on shape optimization, the advantages of having a unified geometry description in all involved models are presented. In the last chapter, the presented method is applied to a coupled fluid-structure-analysis of a wind turbine blade which serves also as a demonstration of the relevance of this method for realistic industrial problems.

The thesis is outlined as follows:

Chapter 2: The geometrical basics needed in this thesis are reviewed. First, the fundamentals for geometric modeling of curves and surfaces are presented, which are necessary in order to understand NURBS. Secondly, the formulas of the differential geometry of surfaces are reviewed, which are needed for the kinematic equations of the shell theory.

Chapter 3: In the third chapter, the fundamentals of structural mechanics are reviewed. First, the general equations of continuum mechanics are presented. Then, they are specified to the Kirchhoff-Love shell theory, employing the respective kinematic assumptions. Furthermore, a short review of the laminated plate theory is given which will be used for the example in Chapter 10. The last part of this chapter provides a general procedure for computing physical stress values from the solution obtained by a displacement-based analysis.

Chapter 4: The basics of isogeometric analysis are reviewed, and particularities and differences to traditional FE analysis are discussed.

Chapter 5: The formulation of the isogeometric Kirchhoff-Love shell is presented. The equations are derived in detail such that the reader can use them for implementing this element. As a special aspect, the treatment of rotational boundary conditions with this formulation is discussed.

Chapter 6: The presented shell formulation is tested on a set of benchmark examples for geometrically linear and nonlinear problems.

Chapter 7: The problem of structures consisting of multiple NURBS patches is discussed and two different solutions are presented. The second one, the so-called bending strip method, is further applied to couple shell elements with solid elements.

Chapter 8: In this chapter, the integration of CAD and CAE is studied. A commercial CAD program is used to demonstrate the feasibility of this integration using the developed shell formulation. Furthermore, general problems and limitations of NURBS-based analysis for the CAD-CAE integration are discussed.

Chapter 9: The concept of isogeometric shell analysis is extended to shape optimization. First, the basics of shape optimization are reviewed. The two existing approaches, CAD-based and FE-based shape optimization, are reviewed and isogeometric shape optimization is presented as a new way to combine the advantages of both approaches.

Chapter 10: In this chapter, the presented method is applied to fluid-structure interaction and it shows its application on a realistic, large-scale example. It presents the isogeometric FSI simulation of a wind turbine blade, which was done in cooperation with the department of structural engineering at UC San Diego.

Chapter 11: The final chapter gives conclusions and outlooks. The results of the various studies performed are summarized and discussed, and ideas for future research are proposed.

Chapter 2

Geometric Fundamentals

In this chapter, the geometric fundamentals of surfaces are reviewed. The first part gives a short overview of methods for the mathematical description of free-form curves and surfaces, with focus on NURBS, which are the basis for the following chapters. In the second part, the most important formulas of the differential geometry of surfaces are presented. These formulas are the basis for the kinematic equations of shells which are derived in Chapter 3.

2.1 Mathematical Description of Curves and Surfaces

There are different ways for the mathematical description of curves and surfaces. The three basic types are the explicit, implicit and parametric description. Each of them has different formulations for analyzing derivatives, continuities and geometrical properties of the curve or surface, and each of them has advantages as well as disadvantages for different applications. In the following, these three representations are shortly presented with the focus on curves. However, the conclusions regarding the applicability for geometrical modeling are valid for surfaces as well.

2.1.1 Explicit Representation

The explicit representation of a curve is the simplest but also the most limited one. Here, one coordinate is a function of the other one, usually $y = f(x)$ (for surfaces, usually the z coordinate is a function of x and y: $z = f(x, y)$). As an example, the quadratic parabola $y = x^2 - 2$ is plotted in Figure 2.1. The advantage of the explicit representation is that it is easy to evaluate the derivatives and by this obtain geometric properties like slope, curvature, etc. It is furthermore easy to determine if a given point is on the curve and to find the intersection of two curves. The significant disadvantage is that the set of possible curves is very limited, since each x-value can take only one y-value. Furthermore, this representation is axis-dependent, i.e. a quadratic interpolant through three points is different for every different coordinate system. For this reason, the explicit representation is rarely used for modeling in computer aided design.

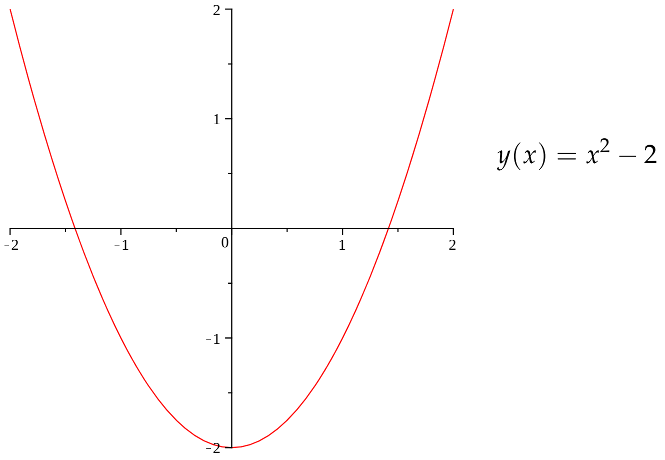


Figure 2.1: Explicit representation of a quadratic parabola.

2.1.2 Implicit Representation

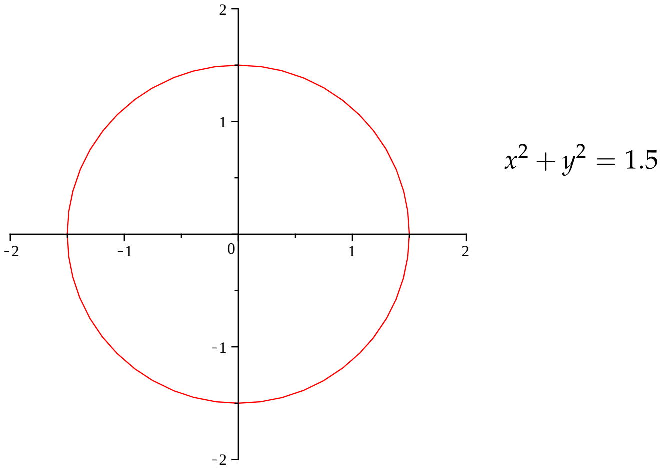
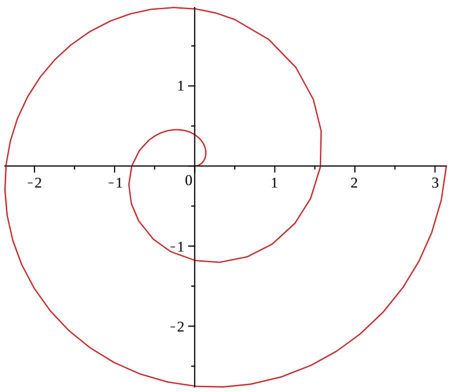


Figure 2.2: Implicit representation of a circle.

A curve in implicit representation is the solution set to an equation of the form $f(x, y) = 0$ (for surfaces: $f(x, y, z) = 0$). With this description, it is possible that more than one point has the same x-value and therefore important geometric objects like, for example, a circle can be described, see Figure 2.2. Obviously, every explicit curve can be expressed in implicit representation but not vice versa. Similar to the explicit representation, it is easy to determine whether a point is on the curve or not. However, it is rather difficult to find the intersection of two curves. Although the variety of possible curves is bigger than in the case of explicit representation, it is still limited. Nevertheless, the implicit description is also used in computer aided design.

For the modeling of free-form curves and surfaces it is in general difficult to find the correct analytic function, which makes both the explicit and the implicit description inconvenient.

2.1.3 Parametric Representation



$$x(t) = 0.25\,t\,cos(t)$$
$$y(t) = 0.25\,t\,sin(t)$$

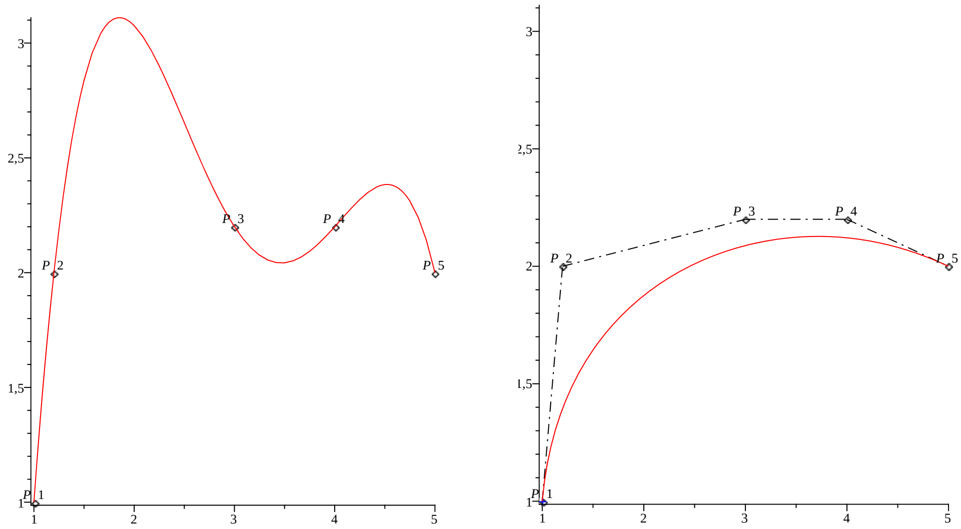
Figure 2.3: Parametric representation of an archimedean spiral.

The most suitable representation for free-form geometries is the parametric description. Here, the coordinates x, y and z are explicit functions of an independent parameter (for surfaces two parameters). This representation is extremely flexible and gives the biggest variety of possible geometries. Figure 2.3 shows an archimedean spiral and its parametric description. With parametric representation, it is also possible to define space curves, whereas with explicit and implicit representation curves can only be defined on a plane. The independent parameter, often denoted as t, is defined in an interval $a \leq t \leq b$. For designing curves, this gives the additional advantage that the curve has a start and an end point which is not the case for the explicit and implicit representation. Although not necessary, this interval is usually normalized to $[0,1]$. As a drawback of the parametric representation, it is difficult to determine whether a point lies on the curve or not and to find the intersection of two curves.
All the methods presented in the next section for modeling free form curves and surfaces are based on the parametric representation.

2.2 NURBS Curves and Surfaces

The first antecedents of NURBS were Bézier curves. B-Splines have emanated from Bézier curves and NURBS from B-Splines. Therefore, this chapter starts with a short review of Bézier curves as an introduction to B-Splines. B-Spline curves and surfaces are explained in more detail since most of the definitions and properties of B-Splines apply to NURBS as well. Finally, NURBS as a generalization of B-Splines are presented.

2.2.1 Bézier Curves



(a)

(b)

Figure 2.4: Fitting data points. (a) Interpolating polynomial. (b) Approximating Bézier curve

A fitting curve is a curve that should approximate a set of given points. If an interpolating polynomial is used, the data points are represented exactly. However, oscillations between the points can occur, see Figure 2.4(a). A Bézier curve is an approximating curve, where the data points, called control points, are not interpolated but only approximated. By this, a smooth and non-oscillating curve is obtained, and the curve "stays inside" the control polygon, see Figure 2.4(b). The control polygon is the linear connection of the control points. As can be seen, only the first and the last point of the control polygon are interpolated. This is important for curve design, because a designer usually wants to be able to exactly specify the beginning and the end of the curve. A Bézier curve is defined by the linear combination of basis functions and control points:

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} B_{i,p}(\xi)\mathbf{P}_i \tag{2.1}$$

where n is the number of control points and $B_{i,p}(\xi)$ are the Bernstein polynomials of polynomial degree p. The polynomial degree is related to the number of control points by: $p = n - 1$. The Bernstein polynomials are defined by [BSMM00]:

$$B_{i,p}(\xi) = \frac{n!}{i!(n-i)!}\xi^i(1-\xi)^{n-i} \tag{2.2}$$

In this form, ξ is defined as $\xi \in [0,1]$.

The problem of Bézier curves is that with an increasing number of control points, the polynomial degree has to be increased. However, with increasing polynomial degree, the approximation to the control polygon deteriorates and furthermore, the algorithms get numerically instable. Furthermore, the global support of the basis functions is a problem for geometric modeling, because it means that any modification of a control point has influence on the whole curve and no local changes can be made to the curve. Another drawback is the fact that no points of reduced continuity, such as kinks, can be inserted inside the curve. These problems can be overcome by using B-Splines.

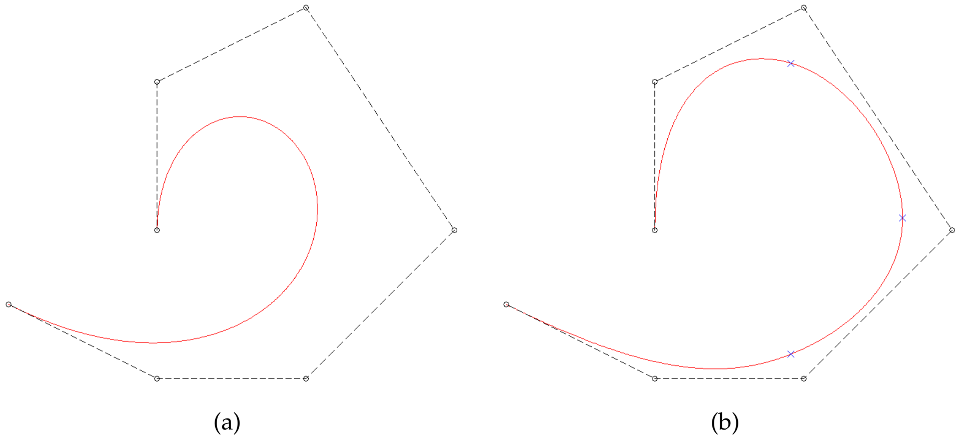


Figure 2.5: Fitting data points. (a) Bézier curve, $p = 6$. (b) B-Spline curve, $p = 3$.

2.2.2 B-Splines

Similar to Bézier curves, B-Spline curves are defined by a linear combination of control points and basis functions over a parametric space. The basis functions are called B-Splines (short for Basis-Splines). The parametric space is divided into intervals and the B-Splines are defined piecewise on these intervals, with certain continuity requirements between the intervals. Since the number of intervals is arbitrary, the polynomial degree can be chosen independently of the number of control points. Therefore, a large set of data points can be approximated while using a low polynomial degree. This is illustrated in Figure 2.5. It shows seven control points which are approximated once by a Bézier curve with $p = 6$ (a) and once by a B-Spline curve with $p = 3$ (b). As a consequence of the lower polynomial degree, the B-Spline curve consists of four sections. Their limits, the so-called knots, are indicated by small crosses on the curve. As can be seen, the B-Spline curve stays closer to the control polygon due to the lower polynomial degree. B-Spline basis functions are defined to be unequal to zero only on a restricted range of intervals which means a local influence of the control points on the curve. Furthermore, it is possible to reduce the continuity in the basis functions between intervals and therefore create kinks inside a curve.

2.2.2.1 Knot Vector

The parametric space is defined by the so called knot vector $\Xi = [\xi_1, \xi_2, \ldots, \xi_{n+p+1}]$. It is a set of parametric coordinates ξ_i in non-descending order which divide the parametric space into sections. If all knots are equally spaced, the knot vector is called uniform. A B-Spline basis function is C^∞ continuous inside a knot span, i.e. between two distinct knots, and C^{p-1} continuous at a single knot. A knot value can appear more than once and is then called a multiple knot. At a knot of multiplicity k the continuity is C^{p-k}, i.e. by increasing the multiplicity of a knot the continuity can be decreased.
If the first and the last knot have the multiplicity $p+1$, the knot vector is called open, clamped, or nonperiodic [PT97]. In this thesis, the term "open" is used. In a B-Spline with an open knot vector the first and the last control point are interpolated and the curve is tangential to the control polygon at the start and the end of the curve, which will be explained in more detail. Since for designing a curve one usually wants to specify its start and end point, open knot vectors are standard in CAD applications and are assumed for the remainder of this thesis, unless stated otherwise.

2.2.2.2 Basis Functions

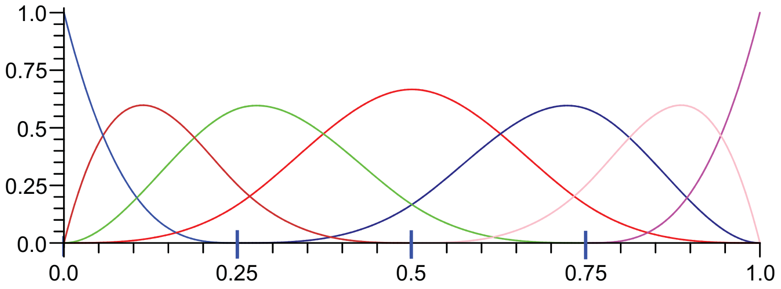


Figure 2.6: Cubic B-Spline basis functions with open knot vector $\Xi = [0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1]$.

Based on the knot vector and the polynomial degree, B-Spline basis functions are computed by the *Cox-deBoor* recursion formula. It starts for $p = 0$ with:

$$N_{i,0}(\xi) = \begin{cases} 1 & \xi_i \leq \xi < \xi_{i+1} \\ 0 & otherwise \end{cases} \tag{2.3}$$

For $p \geq 1$ it is

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{2.4}$$

From this formulation some important properties of B-Spline basis functions can be deduced:

- Local support, i.e. a basis function $N_{i,p}(\xi)$ is non-zero only in the interval $[\xi_i, \xi_{i+p+1}]$.
- Partition of unity, i.e. $\sum\limits_{i=1}^{n} N_{i,p}(\xi) = 1$.
- Non-negativity, i.e. $N_{i,p}(\xi) \geq 0$.
- Linear independence, i.e. $\sum\limits_{i=1}^{n} \alpha_i N_{i,p}(\xi) = 0 \quad \Leftrightarrow \alpha_j = 0, j = 1,2,\ldots,n$.

Figure 2.6 shows an example of cubic B-Spline basis functions with an open knot vector. These are the basis functions corresponding to the curve in Figure 2.5(b). The longer vertical lines on the x-axis indicate the inner knot locations which divide the curve into sections.

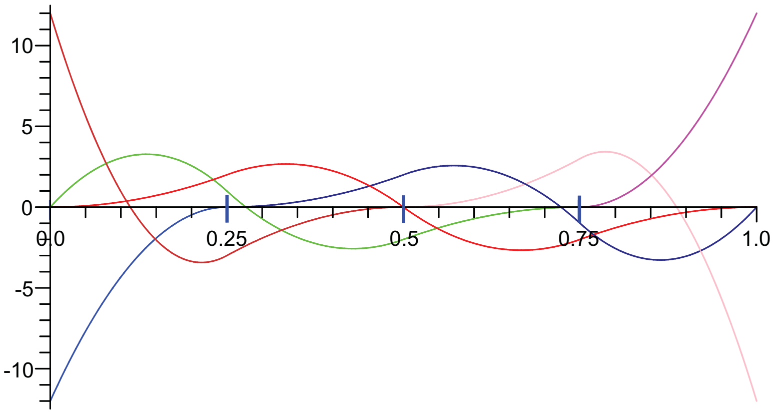


Figure 2.7: First derivatives of cubic B-Spline basis functions with open knot vector $\Xi = [0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1]$.

The first derivative of a B-Spline basis function is computed by the following formula:

$$N'_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{2.5}$$

Figure 2.7 shows the first derivatives of the cubic basis functions from Figure 2.6.

The open knot vector has the following effects on the basis functions and their derivatives:
At $\xi = 0$ all basis functions except the first one vanish:

$$\begin{aligned} N_{1,p}(0) &= 1 \\ N_{i,p}(0) &= 0, \quad i \neq 1 \end{aligned} \tag{2.6}$$

For the derivatives, only the first two are non-zero and $N'_{1,p}(0) = -N'_{2,p}(0)$ holds:

$$\begin{aligned} N'_{1,p}(0) &= -\frac{p}{\xi_{p+2}} \\ N'_{2,p}(0) &= \frac{p}{\xi_{p+2}} \\ N'_{i,p}(0) &= 0, \quad i > 2 \end{aligned} \tag{2.7}$$

The same applies to the basis functions at the end of the parametric space, which can be seen in Figures 2.6 and 2.7.

As mentioned above, at multiple knots the basis functions have reduced continuity C^{p-k}, with k as the knot multiplicity. If $k = p$, the basis functions are C^0 continuous at this knot. In such a case, all basis functions at this knot vanish, except one which takes on the value 1. Figure 2.8 shows the cubic basis functions for a knot vector with an inner knot of multiplicity $k = p = 3$.

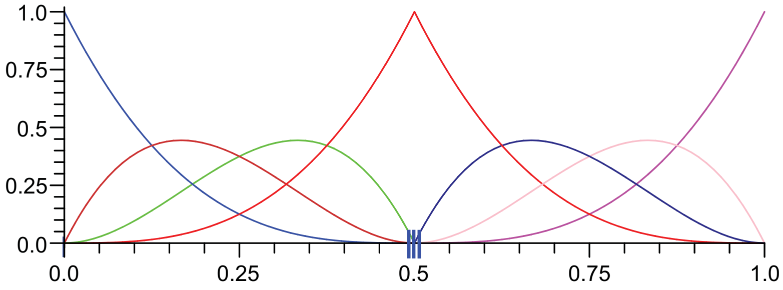


Figure 2.8: Cubic B-Spline basis functions with a multiple inner knot, $\Xi = [0, 0, 0, 0, 0.5, 0.5, 0.5, 1, 1, 1, 1]$.

2.2.2.3 B-Spline Curves

Similar to Bézier curves, a B-Spline curve of degree p is computed by the linear combination of control points and the respective basis functions:

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

The first derivative of the curve is obtained by the linear combination of control points and the derived basis functions:

$$\mathbf{C}'(\xi) = \sum_{i=1}^{n} N'_{i,p}(\xi)\mathbf{P}_i \tag{2.9}$$

In Figure 2.5(b) a cubic B-Spline curve with open knot vector is given. As can be seen, the first and last control point are interpolated and the curve is tangential to the control polygon at its start and end. This is the effect of the open knot vector which can be understood by inserting equations (2.6) and (2.7) into equations (2.8) and (2.9):

$$\mathbf{C}(0) = \sum_{i=1}^{n} N_{i,p}(0)\mathbf{P}_i = \mathbf{P}_1 \tag{2.10}$$

$$\mathbf{C}'(0) = \sum_{i=1}^{n} N'_{i,p}(0)\mathbf{P}_i = \frac{p}{\xi_{p+2}}(\mathbf{P}_2 - \mathbf{P}_1) \tag{2.11}$$

Equation (2.10) shows that for an open knot vector, the first control point is interpolated at $\xi = 0$, while equation (2.11) shows that the curve is tangential to $\overline{\mathbf{P}_1\mathbf{P}_2}$ at $\xi = 0$. The same holds for the other end of the curve, i.e. for $\xi = 1$ and the respective control points. Understanding these properties is very important for defining continuity conditions between B-Spline curves and surfaces, which will be treated in Section 2.3.

If an inner knot has the multiplicity $k = p$, the continuity at this point is reduced to C^0, i.e. by this a kink can be inserted in the curve. As can be seen in Figure 2.8, one basis function takes on the value 1 at this point which means that the corresponding control point is interpolated by the curve. Figure 2.9 shows a cubic B-Spline with a multiple inner knot of $k = p$. For this curve, the basis functions from Figure 2.8 have been applied to the control points from Figure 2.5.

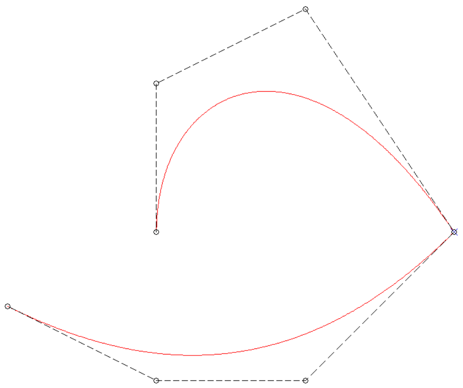


Figure 2.9: Interpolating cubic B-Spline curve due to a multiple inner knot, $\Xi = [0,0,0,0,0.5,0.5,0.5,1,1,1,1]$.

The local support property of B-Splines is demonstrated in Figure 2.10. A B-Spline curve is modified by changing the y-coordinate of the last control point. As shown in Figure 2.6, the corresponding basis function has support only in the last knot span. Therefore, this modification has effect on the curve only in the last section, which can be seen in Figure 2.10(b). If one of the inner control points is modified, it has influence on a couple of sections, at maximum on $p+1$ sections.

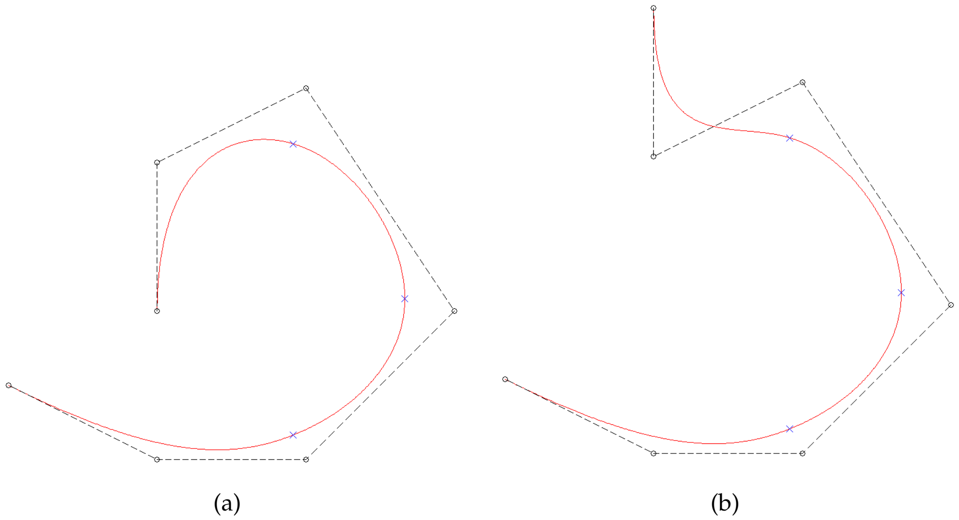


Figure 2.10: Local influence of control points on B-Spline curves. (a) Initial curve. (b) The last control point is modified.

Important properties of B-Spline curves are:

- Convex hull property: the curve is contained inside the convex hull of the control polygon.
- In general, the control points are not interpolated.
- The control points have influence on maximum $p+1$ sections.
- For open knot vectors, the first and the last control point are interpolated and the curve is tangential to the control polygon at the start and the end of the curve.
- The curve is C^{∞} continuous between two knots and C^{p-k} continuous at a knot of multiplicity k.
- Affine transformations of the B-Spline curve are performed by transforming the control points correspondingly.
- A Bézier curve is a B-Spline curve with only one knot interval.

2.2.2.4 B-Spline Surfaces

A B-Spline surface is computed by the tensor product of B-Spline basis functions in two parametric dimensions ξ and η. It is defined by a net of $n \times m$ control points, two knot vectors Ξ and H, two polynomial degrees p and q (which do not need to be equal), and correspondingly the basis functions $N_{i,p}(\xi)$ and $M_{j,q}(\eta)$. It is described as:

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi) M_{j,q}(\eta) \mathbf{P}_{i,j} \tag{2.12}$$

Figure 2.11 shows an example of a quadratic B-Spline surface and its control net (dashed lines). The isocurves on the surface mark the knots which divide the surface into elements. Open knot vectors are employed in both directions. Therefore, the boundaries of the surface are solely defined by the control points at the boundary and the vertices of the surface are interpolated. Analogously to curves, the slopes along a boundary are defined by the first two rows of control points from the boundary.

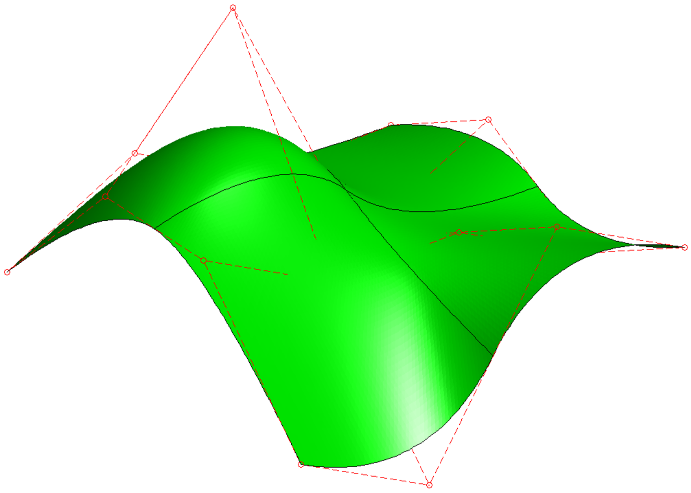


Figure 2.11: B-Spline surface.

2.2.2.5 B-Spline Solids

Similar to surfaces, a B-Spline solid is obtained by the tensor product of B-Spline basis functions in three parametric dimensions ξ, η and ζ:

$$\mathbf{B}(\xi,\eta,\zeta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l} N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) \mathbf{P}_{i,j,k} \tag{2.13}$$

Since this thesis focuses on surface structures, B-Spline solids are not regarded in more detail here.

2.2.3 NURBS

"NURBS" is the abbreviation for Non-Uniform Rational B-Splines. The term "non-uniform" refers to the knot vector which in general is not uniform. The term "rational" refers to the basis functions. While for B-Splines, the basis functions are piecewise polynomials, for NURBS they are piecewise rational polynomials.

For a NURBS curve, each control point has additionally to its coordinates an individual weight w_i. Such a point $P_i(x_i, y_i, z_i, w_i)$ can be represented with homogeneous coordinates $P_i^w(w_i x_i, w_i y_i, w_i z_i, w_i)$ in a projective $\mathbb{R}^4$ space. A NURBS curve is the projection of a B-Spline in $\mathbb{R}^4$ with homogeneous control points onto $\mathbb{R}^3$ [PT97]:

$$\mathbf{C}(\xi) = \frac{\sum\limits_{i=1}^{n} N_{i,p}(\xi) w_i \mathbf{P}_i}{\sum\limits_{i=1}^{n} N_{i,p}(\xi) w_i} \tag{2.14}$$

Defining NURBS basis functions as:

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi) w_i}{\sum\limits_{i=1}^{n} N_{i,p}(\xi) w_i} \tag{2.15}$$

one can write a NURBS curve in the common way as the sum of control points times the respective basis functions:

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} R_{i,p}(\xi) \mathbf{P}_i \tag{2.16}$$

If all control weights are equal, the rational functions in equation (2.15) reduce to the normal B-Spline functions. This means that a B-Spline is a special case of NURBS with equal control weights, and all properties of B-Splines listed in Section 2.2.2.3 apply to NURBS as well.

In Figure 2.12, a cubic NURBS curve is defined through the control points of the previous example, with an increased weighting on the uppermost control point $w_5 = 10$. It can be seen that due to this higher control weight, the curve is "pulled" towards this control point.

The significant advantage of the rational basis functions is that they allow an exact representation of conic sections, which includes circles and ellipses. Figure 2.13 shows a circle represented by a NURBS curve. Therefore, NURBS are able to represent smooth free form shapes as well as linear shapes, sharp edges, and kinks, and also such important geometric objects like spheres, cylinders, etc. This is why NURBS have become established as a standard in CAD modeling.

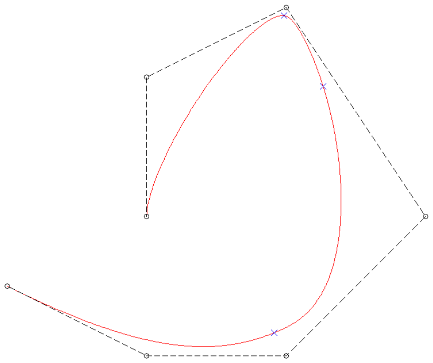


Figure 2.12: NURBS curve with increased weighting on the uppermost control point $w_5 = 10$.

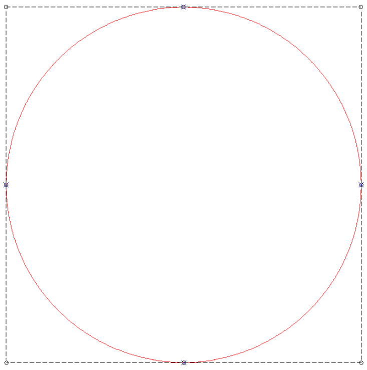


Figure 2.13: Exact circle represented by a NURBS curve.

Similar to B-Splines, a NURBS surface is defined as:

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\mathbf{P}_{i,j} \tag{2.17}$$

with the basis functions:

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}}{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m} N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}} \tag{2.18}$$

and a NURBS solid is defined as:

$$\mathbf{B}(\xi,\eta,\zeta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l} R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)\mathbf{P}_{i,j,k} \tag{2.19}$$

with the basis functions:

$$R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)w_{i,j,k}}{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}\sum\limits_{k=1}^{l} N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)w_{i,j,k}} \tag{2.20}$$

It is important to note that two- and three-dimensional NURBS basis functions (equations (2.18) and (2.20)) are not tensor products of one-dimensional NURBS basis functions (equation (2.15)). They are obtained as the weighted ratio of tensor products of B-Spline basis functions.

In Figure 2.14(a), a sphere which is modeled by a NURBS surface is depicted. Figure 2.14(b) shows the geometry and the control points.

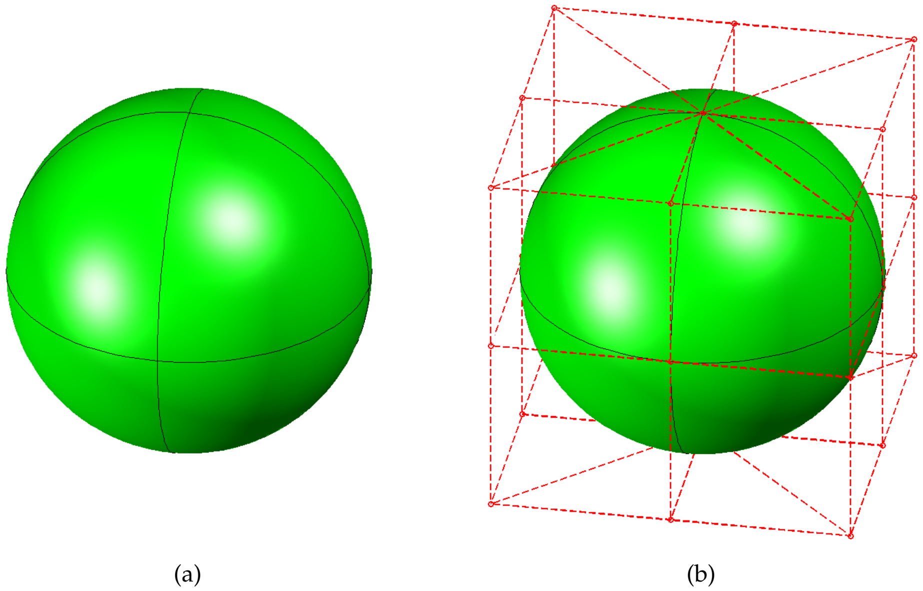


Figure 2.14: Exact sphere represented by a NURBS surface. (a) Surface. (b) Surface and control grid.

2.2.4 Refinement

There are two ways of mesh refining a NURBS curve or surface, namely knot insertion and order elevation. Both methods enhance the design space by adding control points to the geometry. In knot insertion, the knot spans are divided into smaller ones by inserting new knots. As a consequence, at this point the continuity is reduced by one. For each additional knot, an additional control point is inserted. In order elevation, the number of knot intervals remains unchanged but the polynomial degree of the basis functions is increased. While increasing the order, existing knots are repeated so that the continuity at these points remains unchanged. For surfaces, these refinement procedures can be applied to both parametric directions ξ and η independently from each other. A very important feature of both methods is that they do not change either the geometry or the parametrization. There are standard algorithms for knot refining and order elevating B-Splines, see [PT97]. For NURBS, the same algorithms can be used, however, they have to be applied to the homogeneous control coordinates $\mathbf{P}_i^w$, which means refining a B-

Spline in the projective $\mathbb{R}^4$ space. After having obtained the refined control points in the projective space, they are projected back to the $\mathbb{R}^3$ space.

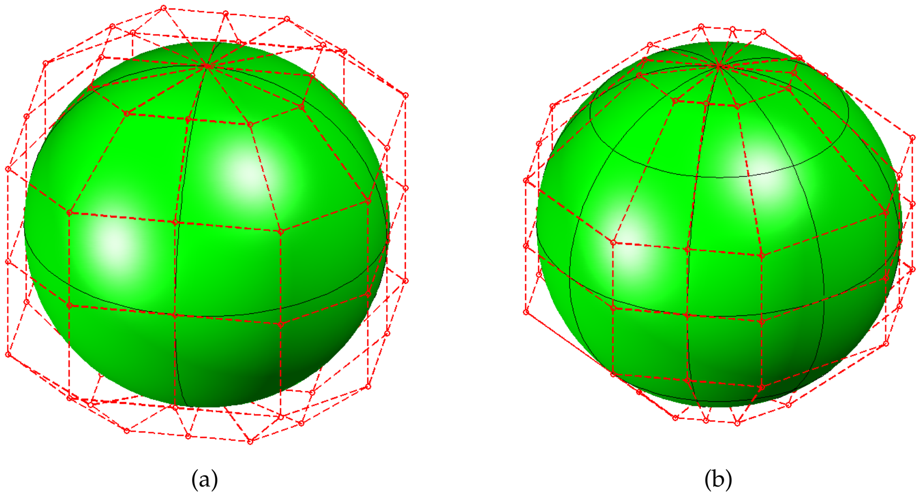


Figure 2.15: NURBS refinement. (a) Order elevation. (b) Knot refinement.

Figure 2.15(a) shows the sphere after order elevation. The control grid is refined, while the number of knot spans remains unchanged. In Figure 2.15(b), knot refinement has been performed. Both the knot mesh and the control grid are refined.

2.3 Continuity

For the isogeometric shell formulation presented in Chapter 5, the continuity between elements and patches plays a crucial role. Therefore, in this section the continuity conditions for B-Splines and NURBS are investigated.

For parametric curves and surfaces there are two kinds of continuity, the geometric and the parametric continuity. For the zeroth order continuity they are equal $G^0 = C^0$, but for a degree of continuity $k \geq 1$ they have to be distinguished. Generally, the parametric continuity C^k implies the geometric continuity G^k but not vice versa. For the proposed method, the geometric continuity G^1 between surfaces is needed, so at first the difference between G^1 and C^1 is briefly discussed.

2.3.1 Geometric vs. Parametric Continuity

Given are two curves $\mathbf{C}^1(\xi)$ and $\mathbf{C}^2(\xi)$, $0 \leq \xi \leq 1$, which join at their ends:

$$\mathbf{C}^1(1) = \mathbf{C}^2(0) \tag{2.21}$$

The curves are C^1 continuous if their first derivatives at the joint are equal:

$$\frac{\partial \mathbf{C}^1(1)}{\partial \xi} = \frac{\partial \mathbf{C}^2(0)}{\partial \xi} \tag{2.22}$$

This means that their tangent vectors at the joint are parallel and have the same magnitude. For G^1 continuity, the tangent vectors only have to be parallel but not necessarily of the same magnitude [Rog01]. So for G^1 the following equation must hold:

$$\frac{\partial \mathbf{C}^1(1)}{\partial \xi} = c \cdot \frac{\partial \mathbf{C}^2(0)}{\partial \xi} \tag{2.23}$$

where c is a scalar multiplier.

2.3.2 G^1 Continuity for B-Spline Curves

The first derivatives at the endpoints of a B-Spline curve are given by equation (2.11):

$$\frac{\partial \mathbf{C}(0)}{\partial \xi} = \frac{p}{\xi_{p+2}}(\mathbf{P}_2 - \mathbf{P}_1) \tag{2.24}$$

$$\frac{\partial \mathbf{C}(1)}{\partial \xi} = \frac{p}{1-\xi_n}(\mathbf{P}_n - \mathbf{P}_{n-1}) \tag{2.25}$$

The factors $\frac{p}{\xi_{p+2}}$ and $\frac{p}{1-\xi_n}$ are scalar multipliers of the tangent vectors and therefore irrelevant for the geometric continuity. The last control point of the first curve is equal to the first control point of the second curve, $\mathbf{P}^1_n = \mathbf{P}^2_1$, so the curves are G^1 continuous if the following condition holds:

$$(\mathbf{P}^2_2 - \mathbf{P}^1_n) = c \cdot (\mathbf{P}^1_n - \mathbf{P}^1_{n-1}) \tag{2.26}$$

This means that the control points $\mathbf{P}^1_{n-1}$, $\mathbf{P}^1_n$ and $\mathbf{P}^2_2$ are collinear, as illustrated in Figure 2.16.

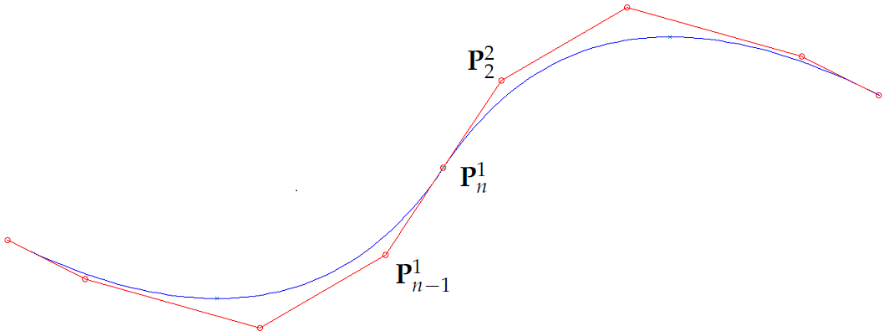


Figure 2.16: G^1-continuous B-spline curves.

2.3.3 G^1 Continuity for B-Spline Surfaces

For two surfaces being G^1 continuous along a common edge, the derivatives w.r.t. both parameters ξ and η have to fulfill the condition of equation (2.23). The first derivatives of a B-Spline surface at the edges $\xi = 0$ and $\xi = 1$ are:

$$\frac{\partial \mathbf{S}(0,\eta)}{\partial \xi} = \frac{p}{\xi_{p+2}} \sum_{j=1}^{m} M_{j,q}(\eta)(\mathbf{P}_{2,j} - \mathbf{P}_{1,j}) \tag{2.27}$$

$$\frac{\partial \mathbf{S}(0,\eta)}{\partial \eta} = \sum_{j=1}^{m} \frac{\partial M_{j,q}(\eta)}{\partial \eta} \mathbf{P}_{1,j} \tag{2.28}$$

$$\frac{\partial \mathbf{S}(1,\eta)}{\partial \xi} = \frac{p}{1-\xi_n} \sum_{j=1}^{m} M_{j,q}(\eta)(\mathbf{P}_{n,j} - \mathbf{P}_{n-1,j}) \tag{2.29}$$

$$\frac{\partial \mathbf{S}(1,\eta)}{\partial \eta} = \sum_{j=1}^{m} \frac{\partial M_{j,q}(\eta)}{\partial \eta} \mathbf{P}_{n,j} \tag{2.30}$$

Given are two patches with a common edge and the same parametrization along the common edge:

$$\mathbf{S}^1(1,\eta) = \mathbf{S}^2(0,\eta) \tag{2.31}$$

$$\mathbf{P}^1_{n,j} = \mathbf{P}^2_{1,j} \qquad M^1_{j,q}(\eta) = M^2_{j,q}(\eta) \qquad j = 1,\ldots,m \tag{2.32}$$

Then, the derivative w.r.t. η along the edge is equal on both surfaces for any point on the edge:

$$\frac{\partial \mathbf{S}^1(1,\eta)}{\partial \eta} = \frac{\partial \mathbf{S}^2(0,\eta)}{\partial \eta} \tag{2.33}$$

The condition:

$$\frac{\partial \mathbf{S}^1(1,\eta)}{\partial \xi} = c \cdot \frac{\partial \mathbf{S}^2(0,\eta)}{\partial \xi} \tag{2.34}$$

is fulfilled if for all control points across the edge the following relation holds:

$$(\mathbf{P}^2_{2,j} - \mathbf{P}^1_{n,j}) = c \cdot (\mathbf{P}^1_{n,j} - \mathbf{P}^1_{n-1,j}) \qquad j = 1,\ldots,m \tag{2.35}$$

2.3.4 G^1 Continuity for NURBS Surfaces

For the G^1 continuity between NURBS patches, the surfaces have to be G^1 continuous in the homogeneous setting, i.e. equation (2.35) has to be applied to the homogeneous control points [FHK02]:

$$(\mathbf{P}^{w,2}_{2,j} - \mathbf{P}^{w,1}_{n,j}) = c \cdot (\mathbf{P}^{w,1}_{n,j} - \mathbf{P}^{w,1}_{n-1,j}) \qquad j = 1,\ldots,m \tag{2.36}$$

2.4 Differential Geometry of Surfaces

In this section, the basics of the differential geometry of surfaces are reviewed which are the basis for the structural shell model described in Chapter 3.

Each point in the three-dimensional space can be identified by its position vector $\mathbf{x}$:

$$\mathbf{x} = x^1\mathbf{e}_1 + x^2\mathbf{e}_2 + x^3\mathbf{e}_3 = x^i\mathbf{e}_i \tag{2.37}$$

where $\mathbf{e_i}$ are the global Cartesian base vectors and x^i the respective coordinates. The Einstein summation convention is used as well as the convention that indices in Latin letters take the values $\{1,2,3\}$ whereas indices in Greek letters take the values $\{1,2\}$.
For the description of free-form geometries, especially surfaces, it is advantageous to use curvilinear coordinates and local bases. Two important bases are the covariant basis $\mathbf{g}_i$ and the contravariant basis $\mathbf{g}^i$. With the corresponding contravariant coordinates θ^i and covariant coordinates θ_i, the position vector $\mathbf{x}$ can be expressed as:

$$\mathbf{x} = \theta^i\mathbf{g}_i = \theta_i\mathbf{g}^i \tag{2.38}$$

The covariant base vectors are defined as:

$$\mathbf{g}_i = \frac{\partial \mathbf{x}}{\partial \theta^i} = \mathbf{x}_{,i} \tag{2.39}$$

Covariant and contravariant base vectors are related by the following condition:

$$\mathbf{g}_i \cdot \mathbf{g}^j = \delta_i^j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \tag{2.40}$$

A surface is a parametrically two-dimensional geometry and each point on the surface is described by two curvilinear coordinates (θ^1, θ^2). This means that the first two covariant base vectors $\mathbf{g}_\alpha$ can be computed as in equation (2.39) whereas the third covariant base vector $\mathbf{g}_3$ is defined as the normalized vector orthogonal to $\mathbf{g}_1$ and $\mathbf{g}_2$:

$$\mathbf{g}_3 = \frac{\mathbf{g}_1 \times \mathbf{g}_2}{|\mathbf{g}_1 \times \mathbf{g}_2|} \tag{2.41}$$

As can be deduced from equation (2.40), the contravariant base vectors $\mathbf{g}^\alpha$ lie in the tangential plane spanned by the covariant base vectors $\mathbf{g}_\alpha$. Therefore, the third contravariant base vector $\mathbf{g}^3$ is equal to $\mathbf{g}_3$:

$$\mathbf{g}^3 = \mathbf{g}_3 \tag{2.42}$$

The base vectors $\mathbf{g}_1$, $\mathbf{g}_2$ can also be used to define a local Cartesian basis. This is an orthogonal and normalized basis with an arbitrary orientation. Here, it is defined such that its first base vector $\mathbf{e}_1$ is parallel to $\mathbf{g}_1$, and $\mathbf{e}_2$ is orthogonal to it, lying in the plane of $\mathbf{g}_1$, $\mathbf{g}_2$. The third base vector $\mathbf{e}_3$ is equal to $\mathbf{g}_3$:

$$\mathbf{e}_1 = \frac{\mathbf{g}_1}{\|\mathbf{g}_1\|} \tag{2.43}$$

$$\mathbf{e}_2 = \frac{\mathbf{g}_2 - (\mathbf{g}_2 \cdot \mathbf{e}_1)\mathbf{e}_1}{\|\mathbf{g}_2 - (\mathbf{g}_2 \cdot \mathbf{e}_1)\mathbf{e}_1\|} \tag{2.44}$$

$$\mathbf{e}_3 = \mathbf{g}_3 \tag{2.45}$$

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

$$\mathbf{g} = g^{\alpha\beta}\, \mathbf{g}_\alpha \otimes \mathbf{g}_\beta = g_{\alpha\beta}\, \mathbf{g}^\alpha \otimes \mathbf{g}^\beta \tag{2.46}$$

The covariant metric coefficients $g_{\alpha\beta}$ are computed by the scalar product of covariant base vectors [Kli89]:

$$g_{\alpha\beta} = \mathbf{g}_\alpha \cdot \mathbf{g}_\beta \tag{2.47}$$

Equation (2.47) is called the *first fundamental form of surfaces*. It contains important properties of the surface as length of the base vectors and the angles between them. The contravariant metric coefficients $g^{\alpha\beta}$ are obtained by the inverse of the covariant coefficient matrix:

$$\left[g^{\alpha\beta}\right] = \left[g_{\alpha\beta}\right]^{-1} \tag{2.48}$$

With the contravariant metric coefficient matrix $g^{\alpha\beta}$, the contravariant base vectors can be computed from the covariant base vectors:

$$\mathbf{g}^\alpha = g^{\alpha\beta}\, \mathbf{g}_\alpha \tag{2.49}$$

and vice versa:

$$\mathbf{g}_\alpha = g_{\alpha\beta}\, \mathbf{g}^\alpha \tag{2.50}$$

The *second fundamental form of surfaces* (equation (2.51)) describes the curvature properties of a surface. The curvature tensor coefficients $b_{\alpha\beta}$ are defined as [Kli89, BK85a]:

$$b_{\alpha\beta} = -\mathbf{g}_\alpha \cdot \mathbf{g}_{3,\beta} = -\mathbf{g}_\beta \cdot \mathbf{g}_{3,\alpha} = \mathbf{g}_{\alpha,\beta} \cdot \mathbf{g}_3 \tag{2.51}$$

In the following chapters, transformations of tensor coefficients between different bases are needed several times. Here, a general approach for such transformations is explained on a general example:
Two different bases $\mathbf{a}_i \otimes \mathbf{a}_j$ and $\mathbf{b}_i \otimes \mathbf{b}_j$ as well as their contravariant counterparts $\mathbf{a}^i \otimes \mathbf{a}^j$ and $\mathbf{b}^i \otimes \mathbf{b}^j$ are given. A tensor $\mathbf{M}$ can be expressed using these bases as:

$$\mathbf{M} = a^{ij}\mathbf{a}_i \otimes \mathbf{a}_j = a_{ij}\mathbf{a}^i \otimes \mathbf{a}^j = b^{ij}\mathbf{b}_i \otimes \mathbf{b}_j = b_{ij}\mathbf{b}^i \otimes \mathbf{b}^j \tag{2.52}$$

Exemplarily, the covariant coefficients a_{ij} shall be transformed to the covariant coefficients b_{ij}:

$$a_{ij}\mathbf{a}^i \otimes \mathbf{a}^j = b_{ij}\mathbf{b}^i \otimes \mathbf{b}^j \tag{2.53}$$

The trick is multiplying the covariant base vectors $\mathbf{b}_k, \mathbf{b}_l$ from two sides, using the fact that co- and contravariant base vectors are related by $\mathbf{b}_i \cdot \mathbf{b}^j = \delta_i^j$:

$$\begin{aligned} a_{ij}\mathbf{b}_k(\mathbf{a}^i \otimes \mathbf{a}^j)\mathbf{b}_l &= b_{ij}\mathbf{b}_k(\mathbf{b}^i \otimes \mathbf{b}^j)\mathbf{b}_l \\ a_{ij}(\mathbf{b}_k \cdot \mathbf{a}^i)(\mathbf{a}^j \cdot \mathbf{b}_l) &= b_{ij}(\mathbf{b}_k \cdot \mathbf{b}^i)(\mathbf{b}^j \cdot \mathbf{b}_l) \\ &= b_{ij}\delta_k^i\delta_l^j \\ &= b_{kl} \end{aligned} \tag{2.54}$$

The same can be done for contravariant coefficients for both bases. So there are four possible transformations:

$$b_{kl} = a_{ij}(\mathbf{b}_k \cdot \mathbf{a}^i)(\mathbf{a}^j \cdot \mathbf{b}_l) \tag{2.55}$$

$$b^{kl} = a_{ij}(\mathbf{b}^k \cdot \mathbf{a}^i)(\mathbf{a}^j \cdot \mathbf{b}^l) \tag{2.56}$$

$$b_{kl} = a^{ij}(\mathbf{b}_k \cdot \mathbf{a}_i)(\mathbf{a}_j \cdot \mathbf{b}_l) \tag{2.57}$$

$$b^{kl} = a^{ij}(\mathbf{b}^k \cdot \mathbf{a}_i)(\mathbf{a}_j \cdot \mathbf{b}^l) \tag{2.58}$$

Equations (2.55) to (2.58) provide the transformation rules for all cases of coefficient transformations that are needed in the following chapters.

Chapter 3

Structural Mechanics of Shells

3.1 Fundamentals of Continuum Mechanics

In this section, the fundamentals of continuum mechanics are reviewed. The basic quantities of differential geometry have been introduced in Section 2.4 and based on these, the most important kinematic equations are derived in the following. Large displacements and small strains are assumed and the Lagrangian description is used.

3.1.1 Kinematics

The kinematics describe the deformation of a body. For a material point on the body it has to be distinguished between the reference (undeformed) and the actual (deformed) configuration. All quantities in the reference configuration are denoted by upper case letters, those referring to the actual configuration by lower case letters.

The deformation $\mathbf{u}$ of a material point is defined by its position vectors in the actual and reference configuration:

$$\mathbf{u} = \mathbf{x} - \mathbf{X} \tag{3.1}$$

The mapping of a differential line element in the reference configuration $\mathrm{d}\mathbf{X}$ into a line element in the deformed configuration $\mathrm{d}\mathbf{x}$ is described by the deformation gradient $\mathbf{F}$ [Hol04]:

$$\mathrm{d}\mathbf{x} = \mathbf{F} \cdot \mathrm{d}\mathbf{X} \tag{3.2}$$

The deformation gradient is defined by the base vectors in the reference and actual configuration [BW00]:

$$\mathbf{F} = \mathbf{g}_i \otimes \mathbf{G}^i \qquad \mathbf{F}^T = \mathbf{G}^i \otimes \mathbf{g}_i \tag{3.3}$$

$$\mathbf{F}^{-1} = \mathbf{G}_i \otimes \mathbf{g}^i \qquad \mathbf{F}^{-T} = \mathbf{g}^i \otimes \mathbf{G}_i \tag{3.4}$$

and can be used for the mapping between deformed and undeformed base vectors:

$$\mathbf{g}_i = \mathbf{F} \cdot \mathbf{G}_i \qquad \mathbf{G}_i = \mathbf{F}^{-1} \cdot \mathbf{g}_i \tag{3.5}$$

$$\mathbf{g}^i = \mathbf{F}^{-T} \cdot \mathbf{G}^i \qquad \mathbf{G}^i = \mathbf{F}^T \cdot \mathbf{g}^i \tag{3.6}$$

The deformation gradient describes the deformation of a body including rigid body motions. Therefore, it cannot be used directly as a measure for strains. Different strain measures exist, the one used in this thesis is the Green-Lagrange strain tensor **E**. It describes a nonlinear relation between deformations and strains and therefore it is an appropriate measure for strains under large deformations. It is defined through the deformation gradient and the identity tensor **I**:

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) = E_{ij}\, \mathbf{G}^i \otimes \mathbf{G}^j \tag{3.7}$$

Inserting equation (3.3) into (3.7) and recalling that the identity tensor is identical to the metric tensor yields:

$$\begin{aligned} \mathbf{E} &= \frac{1}{2}((\mathbf{G}^i \otimes \mathbf{g}_i) \cdot (\mathbf{g}_j \otimes \mathbf{G}^j) - G_{ij}\, \mathbf{G}^i \otimes \mathbf{G}^j) \\ &= \frac{1}{2}(g_{ij} - G_{ij})\, \mathbf{G}^i \otimes \mathbf{G}^j \end{aligned} \tag{3.8}$$

Thus, the Green-Lagrange strain coefficients E_{ij} are computed from the metric coefficients in the actual and reference configuration:

$$E_{ij} = \frac{1}{2}(g_{ij} - G_{ij}) \tag{3.9}$$

and they refer to the contravariant basis $\mathbf{G}^i \otimes \mathbf{G}^j$ of the undeformed configuration.

3.1.2 Constitutive Equations

The constitutive equations describe the relation between strains and stresses via a material law. Similar to strain tensors, there exist different definitions of stress tensors. The energetically conjugate quantity to the Green-Lagrange strain tensor **E** is the second Piola-Kirchhoff (PK2) stress tensor **S** [BW00]. It can be derived from the strain energy W^{int}:

$$\mathbf{S} = \frac{\partial W^{int}}{\partial \mathbf{E}} \tag{3.10}$$

Stress and strain tensor are related by the elasticity tensor **C**, also called material tensor. It is a fourth order tensor and is defined as:

$$\mathbf{C} = \frac{\partial \mathbf{S}}{\partial \mathbf{E}} = \frac{\partial^2 W^{int}}{\partial \mathbf{E}^2} \tag{3.11}$$

In this thesis, a St.Venant-Kirchhoff material model is used, which means that a linear relation between strains and stresses is assumed:

$$\mathbf{S} = \mathbf{C} : \mathbf{E} \tag{3.12}$$

$$S^{ij} = C^{ijkl} E_{kl} \tag{3.13}$$

$$\mathbf{S} = S^{ij} \mathbf{G}_i \otimes \mathbf{G}_j \tag{3.14}$$

For the description of an isotropic elastic material, two independent parameters are sufficient. There are different parameters used in mathematical and engineering literature. Usually, in the engineering literature the Young's modulus E and Poisson's ratio ν are used whereas in mathematical literature, the Lamé constants λ and μ can be found. They are connected by the following relations:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \qquad \mu = \frac{E}{2(1+\nu)} \tag{3.15}$$

As mentioned above, the second Piola-Kirchhoff stress tensor $\mathbf{S}$ is the energetically conjugate to the Green-Lagrange strain tensor. However, it does not represent physical stresses. Those are described by the Cauchy stress tensor $\boldsymbol{\sigma}$. The Cauchy and the PK2 stress tensor are related by the deformation gradient as follows:

$$\boldsymbol{\sigma} = (\det\mathbf{F})^{-1} \cdot \mathbf{F} \cdot \mathbf{S} \cdot \mathbf{F}^T \tag{3.16}$$

$$\mathbf{S} = \det\mathbf{F} \cdot \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \tag{3.17}$$

Another common stress tensor is the first Piola-Kirchhoff (PK1) stress tensor $\mathbf{P}$, which can be obtained by:

$$\mathbf{P} = \det\mathbf{F} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} = \mathbf{F} \cdot \mathbf{S} \tag{3.18}$$

3.1.3 Equilibrium

The equilibrium equations describe the balance between internal and external forces. If these equations are satisfied, the system is in equilibrium. In the reference configuration, the equilibrium is formulated as:

$$\mathrm{div}\mathbf{P} + \rho_0\mathbf{B} = \mathrm{div}(\mathbf{F} \cdot \mathbf{S}) + \rho_0\mathbf{B} = \mathbf{0} \tag{3.19}$$

where ρ_0 is the density and $\mathbf{B}$ the vector of body forces, both in the reference configuration.

Equations (3.7), (3.12) and (3.19), together with the appropriate boundary conditions, represent the strong form of the boundary value problem. For general three dimensional problems, the strong form of the problem cannot be solved exactly and discretization methods are employed, like the Finite Element Method, which is also used in this thesis. In the Finite Element Method, the field equations and boundary conditions are not satisfied point-wise, but in an integral sense only. The resulting equation of equilibrium is called the weak form of the problem. The finite elements developed in this thesis are based on the Principle of Virtual Work, to be more precise, the Principle of Virtual Displacements. It says that if an infinite small virtual displacement $\delta\mathbf{u}$ is applied to a system, the sum of internal and external virtual work, done by the internal and external forces on the virtual displacement, vanishes if the system is in equilibrium [WP03]:

$$\delta W = \delta W_{int} + \delta W_{ext} = 0 \tag{3.20}$$

The internal and external virtual work are defined as:

$$\delta W_{int} = -\int\limits_{\Omega} \delta \mathbf{E} : \mathbf{S} \, \mathrm{d}\Omega \tag{3.21}$$

$$\delta W_{ext} = \int\limits_{\Gamma} \mathbf{T} \cdot \delta \mathbf{u} \, \mathrm{d}\Gamma + \int\limits_{\Omega} \rho \mathbf{B} \cdot \delta \mathbf{u} \, \mathrm{d}\Omega \tag{3.22}$$

where Ω is the domain and Γ the domain boundary in the reference configuration and $\mathbf{T}$ is the vector of boundary forces.

3.2 Kirchhoff-Love Shell Theory

The Kirchhoff-Love theory is a shell theory based on the "direct approach". It means that the shell formulation is not derived from three-dimensional continuum mechanics, but the shell is regarded ab initio as a two-dimensional surface and proper kinematic assumptions, representing the three-dimensional behavior, are postulated. The director, which is a vector field on the shell's middle surface, describes the thickness extension of the shell. For a comprehensive review of the different approaches to shell theories, reference is made to [BWBR04].

For shell theories based on the direct approach, normal strains and stresses in through-the-thickness direction are not regarded. In the Kirchhoff-Love shell theory, cross sections are assumed to remain straight during deformation, which corresponds to a linear strain distribution through the thickness. Furthermore, it assumes that the cross sections that are normal to the middle surface, remain normal to the middle surface in the deformed configuration. This means that the director is always normal to the middle surface and as a consequence the definition of an independent director is redundant and the shell can be completely represented by its middle surface. Mechanically, the assumption of cross sections remaining normal to the midsurface means that transverse shear strains are neglected. This is a reasonable assumption for thin structures. The range where this assumption is valid, is defined by the shell's slenderness $R/t > 20$, with R as the radius of curvature and t as the shell's thickness. Therefore, most shells in practical application can be classified as thin shells.

In the following, these assumptions are applied to the kinematic, constitutive, and equilibrium equations introduced in the previous chapters.

Since both transversal normal strains and transversal shear strains are neglected, only the in-plane strain coefficients are considered and equation (3.7) reduces to:

$$\mathbf{E} = E_{\alpha\beta}\, \mathbf{G}^{\alpha} \otimes \mathbf{G}^{\beta} \tag{3.23}$$

According to equation (3.9), the strain coefficients $E_{\alpha\beta}$ are defined by:

$$E_{\alpha\beta} = \frac{1}{2}(g_{\alpha\beta} - G_{\alpha\beta}) \tag{3.24}$$

Due to the assumption of straight cross sections, every point in the shell continuum can be described by the middle surface and its normal vector. With t as the shell thickness and θ^3 as the thickness coordinate ranging from $(-0.5t \leq \theta^3 \leq 0.5t)$, the middle surface is defined by $\mathbf{x}(\theta^3 = 0)$. The base vectors on the middle surface are denoted by $\mathbf{a}_i$ and obtained by:

$$\mathbf{a}_\alpha = \mathbf{x}_{,\alpha}\,(\theta^3 = 0) \tag{3.25}$$

$$\mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{|\mathbf{a}_1 \times \mathbf{a}_2|} \tag{3.26}$$

Corresponding to equations (2.47) and (2.51), the metric and curvature coefficients of the middle surface are defined by:

$$a_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta \tag{3.27}$$

$$b_{\alpha\beta} = -\mathbf{a}_\alpha \cdot \mathbf{a}_{3,\beta} = -\mathbf{a}_\beta \cdot \mathbf{a}_{3,\alpha} = \mathbf{a}_{\alpha,\beta} \cdot \mathbf{a}_3 \tag{3.28}$$

The position vector $\mathbf{x}$ of a point in the shell continuum is then defined by:

$$\mathbf{x} = \theta^\alpha \mathbf{a}_\alpha + \theta^3 \mathbf{a}_3 \tag{3.29}$$

which yields for the base vectors $\mathbf{g}_\alpha$:

$$\mathbf{g}_\alpha = \mathbf{a}_\alpha + \theta^3 \mathbf{a}_{3,\alpha} \tag{3.30}$$

and for the metric coefficients $g_{\alpha\beta}$:

$$\begin{aligned} g_{\alpha\beta} &= (\mathbf{a}_\alpha + \theta^3 \mathbf{a}_{3,\alpha}) \cdot (\mathbf{a}_\beta + \theta^3 \mathbf{a}_{3,\beta}) \\ &= a_{\alpha\beta} - 2\,\theta^3\, b_{\alpha\beta} + (\theta^3)^2\, \mathbf{a}_{3,\alpha} \cdot \mathbf{a}_{3,\beta} \end{aligned} \tag{3.31}$$

For thin and moderately thick shells, the quadratic term with respect to θ^3 is neglected [BWBR04], which yields:

$$g_{\alpha\beta} = a_{\alpha\beta} - 2\,\theta^3\, b_{\alpha\beta} \tag{3.32}$$

Inserting (3.32) into (3.24) yields:

$$E_{\alpha\beta} = \frac{1}{2}(a_{\alpha\beta} - A_{\alpha\beta}) + \theta^3 (B_{\alpha\beta} - b_{\alpha\beta}) \tag{3.33}$$

With this formula, the strains in the shell continuum are represented by the metric and curvature coefficients of the middle surface. It can be seen that the strains consist of a constant and a linear part. The constant part describes the strains in the middle surface and corresponds to membrane action. Correspondingly, the membrane strains $\varepsilon_{\alpha\beta}$ are defined as:

$$\varepsilon_{\alpha\beta} = \frac{1}{2}(a_{\alpha\beta} - A_{\alpha\beta}) \tag{3.34}$$

The linear part (symmetric with respect to the middle surface) represents the change in curvature and is the effect of bending. The change in curvature is denoted by $\kappa_{\alpha\beta}$:

$$\kappa_{\alpha\beta} = B_{\alpha\beta} - b_{\alpha\beta} \tag{3.35}$$

With $\varepsilon_{\alpha\beta}$ and $\kappa_{\alpha\beta}$, equation (3.33) can be rewritten as:

$$E_{\alpha\beta} = \varepsilon_{\alpha\beta} + \theta^3 \kappa_{\alpha\beta} \tag{3.36}$$

The separation into membrane and bending action is also applied to the stresses, and an integration through the thickness is performed, which yields the stress resultants **n** for normal forces and **m** for bending moments. Since the stress distribution through the thickness is defined to be linear, a pre-integration can be performed analytically:

$$S^{\alpha\beta} = C^{\alpha\beta\gamma\delta} E_{\gamma\delta} \tag{3.37}$$

$$n^{\alpha\beta} = \int_{-t/2}^{t/2} S^{\alpha\beta} \, \mathrm{d}\theta^3 = C^{\alpha\beta\gamma\delta} \varepsilon_{\gamma\delta} \cdot t \tag{3.38}$$

$$m^{\alpha\beta} = \int_{-t/2}^{t/2} S^{\alpha\beta} \cdot \theta^3 \, \mathrm{d}\theta^3 = C^{\alpha\beta\gamma\delta} \kappa_{\gamma\delta} \cdot \frac{t^3}{12} \tag{3.39}$$

As the strain and the stress tensor are both symmetric, there are only three independent strain coefficients E_{11}, E_{22} and E_{12} as well as three independent stress coefficients S^{11}, S^{22} and S^{12}. They can be gathered in vectors, and the constitutive equation can be written in Voigt notation:

$$\begin{bmatrix} S^{11} \\ S^{22} \\ S^{12} \end{bmatrix} = \widetilde{\mathbf{D}} \cdot \begin{bmatrix} E_{11} \\ E_{22} \\ 2E_{12} \end{bmatrix} \tag{3.40}$$

where $\widetilde{\mathbf{D}}$ is the material matrix. Usually, the material matrix is built using physical material parameters like the Young's modulus E. Since these quantities refer to normalized units (like m or mm), the strains and stresses in equation (3.40) need to be expressed in a local Cartesian coordinate system. Equation (2.55) can be used to transform the strain coefficients from $E_{\alpha\beta}$ to $\bar{E}_{\alpha\beta}$, where the upper bar signifies that these coefficients refer to a local Cartesian basis:

$$\bar{E}_{\gamma\delta} = E_{\alpha\beta}(\mathbf{E}_\gamma \cdot \mathbf{G}^\alpha)(\mathbf{G}^\beta \cdot \mathbf{E}_\delta) \tag{3.41}$$

The local Cartesian base vectors $\mathbf{E}_\gamma, \mathbf{E}_\delta$ are obtained by equations (2.43)-(2.44) and are denoted by capital letters here, to make clear that they refer to the reference configuration.

Now, a material matrix **D** using physical components can be used to compute the PK2 stress coefficients $\bar{S}^{\alpha\beta}$. As for the strains, the upper bar notation refers to a local Cartesian basis:

$$\begin{bmatrix} \bar{S}^{11} \\ \bar{S}^{22} \\ \bar{S}^{12} \end{bmatrix} = \mathbf{D} \cdot \begin{bmatrix} \bar{E}_{11} \\ \bar{E}_{22} \\ 2\bar{E}_{12} \end{bmatrix} \tag{3.42}$$

For an isotropic material, the material matrix is defined as:

$$\mathbf{D}^{iso} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{3.43}$$

with E as the Young's modulus and ν as the Poisson's ratio.
In the case of an orthotropic material, there are different Young's moduli E_1, E_2 and Poisson's ratios ν_{12}, ν_{21} for the two material directions, where $\nu_{21}E_1 = \nu_{12}E_2$ must hold to ensure the symmetry of the material matrix. Together with the shear modulus G_{12}, the orthotropic material matrix is defined as:

$$\mathbf{D}^{ort} = \begin{bmatrix} \dfrac{E_1}{(1-\nu_{12}\nu_{21})} & \dfrac{\nu_{21}E_1}{(1-\nu_{12}\nu_{21})} & 0 \\ \dfrac{\nu_{12}E_2}{(1-\nu_{12}\nu_{21})} & \dfrac{E_2}{(1-\nu_{12}\nu_{21})} & 0 \\ 0 & 0 & G_{12} \end{bmatrix} \tag{3.44}$$

Like the stress tensor **S**, the tensors **n** and **m** are symmetric and their coefficients can be computed using the material matrix and Voigt notation, analogously to equation (3.42). The bar over the coefficients $(\bar{\ })$ again refers to a local Cartesian basis:

$$\begin{bmatrix} \bar{n}^{11} \\ \bar{n}^{22} \\ \bar{n}^{12} \end{bmatrix} = t \cdot \mathbf{D} \cdot \begin{bmatrix} \bar{\varepsilon}_{11} \\ \bar{\varepsilon}_{22} \\ 2\bar{\varepsilon}_{12} \end{bmatrix} \tag{3.45}$$

$$\begin{bmatrix} \bar{m}^{11} \\ \bar{m}^{22} \\ \bar{m}^{12} \end{bmatrix} = \frac{t^3}{12} \cdot \mathbf{D} \cdot \begin{bmatrix} \bar{\kappa}_{11} \\ \bar{\kappa}_{22} \\ 2\bar{\kappa}_{12} \end{bmatrix} \tag{3.46}$$

With normal forces and bending moments, the internal virtual work can now be formulated as:

$$\delta W_{int} = -\int_{\Omega} (\mathbf{S} : \delta\mathbf{E})\, \mathrm{d}\Omega = -\int_{A} (\mathbf{n} : \delta\boldsymbol{\varepsilon} + \mathbf{m} : \delta\boldsymbol{\kappa})\, \mathrm{d}A \tag{3.47}$$

where dA is the differential area of the midsurface in the reference configuration. Equation (3.47) is a weak form of the partial differential equations for the Kirchhoff-Love shell.

3.3 Laminated Plate Theory

A composite laminate is a material that is composed of a series of plies with orthotropic material behavior, where the plies can have different orientations of their principal material coordinates. For thin shells, such materials can be modeled by the classical laminated plate theory [Red04, Sch05, Bai07], which creates homogenized material matrices, so that the stress resultants **n** and **m** can be directly obtained from $\boldsymbol{\varepsilon}$ and $\boldsymbol{\kappa}$. In the case of a non-symmetric layout of the plies, a coupling between membrane strains and bending moments, as well as curvatures and normal forces occurs. Therefore, three material matrices are defined, the extensional stiffness **A**, the coupling stiffness **B** and the bending stiffness **D**.

The homogenized material matrices are obtained by integrating the material matrices of each ply over the height.

For a single ply, the material is described by an orthotropic material matrix $\mathbf{D}^{ort}$ as defined in (3.44). Since this matrix is referred to the ply's principal direction, i.e. its fiber direction, it needs to be transformed to the shell's principal directions by a rotation matrix $\mathbf{T}(\phi)$:

$$\mathbf{T}(\phi) = \begin{bmatrix} \cos^2\phi & \sin^2\phi & \sin\phi\cos\phi \\ \sin^2\phi & \cos^2\phi & -\sin\phi\cos\phi \\ -2\sin\phi\cos\phi & 2\sin\phi\cos\phi & \cos^2\phi - \sin^2\phi \end{bmatrix} \tag{3.48}$$

where ϕ is the fiber angle. The rotated material matrix of a ply k is then obtained by:

$$\overline{\mathbf{D}}_k = \mathbf{T}^T(\phi)\,\mathbf{D}_k^{ort}\,\mathbf{T}(\phi) \tag{3.49}$$

For a laminate with n plies of equal thickness t/n, the matrices **A**, **B** and **D** are obtained by the following formulas:

$$\mathbf{A} = \frac{t}{n}\sum_{k=1}^{n}\overline{\mathbf{D}}_k \tag{3.50}$$

$$\mathbf{B} = \frac{t^2}{n^2}\sum_{k=1}^{n}\overline{\mathbf{D}}_k\left(k - \frac{n}{2} - \frac{1}{2}\right) \tag{3.51}$$

$$\mathbf{D} = \frac{t^3}{n^3}\sum_{k=1}^{n}\overline{\mathbf{D}}_k\left[\left(k - \frac{n}{2} - \frac{1}{2}\right)^2 + \frac{1}{12}\right] \tag{3.52}$$

Normal forces and bending moments are then obtained by:

$$\begin{bmatrix} \bar{n}^{11} \\ \bar{n}^{22} \\ \bar{n}^{12} \end{bmatrix} = \mathbf{A}\cdot\begin{bmatrix} \bar{\varepsilon}_{11} \\ \bar{\varepsilon}_{22} \\ 2\bar{\varepsilon}_{12} \end{bmatrix} + \mathbf{D}\cdot\begin{bmatrix} \bar{\kappa}_{11} \\ \bar{\kappa}_{22} \\ 2\bar{\kappa}_{12} \end{bmatrix} \tag{3.53}$$

$$\begin{bmatrix} \bar{m}^{11} \\ \bar{m}^{22} \\ \bar{m}^{12} \end{bmatrix} = \mathbf{D} \cdot \begin{bmatrix} \bar{\varepsilon}_{11} \\ \bar{\varepsilon}_{22} \\ 2\bar{\varepsilon}_{12} \end{bmatrix} + \mathbf{B} \cdot \begin{bmatrix} \bar{\kappa}_{11} \\ \bar{\kappa}_{22} \\ 2\bar{\kappa}_{12} \end{bmatrix} \tag{3.54}$$

Having computed normal forces and bending moments by this modified material law, the shell formulation of equation (3.47) can be used without any further modifications.

3.4 Stress Recovery

The derivations and equations in Section 3.2 are the basis for a structural analysis using the Kirchhoff-Love theory. If such a problem is solved by a finite element analysis, the results are usually displacements. This means that the analysis yields as result the deformed configuration of the geometry. Other important quantities of interest in structural analysis are the stresses and stress resultants. Computing the stresses from the obtained displacements is called stress recovery.
In the following, a procedure for stress recovery is presented, which is generally valid and not related to a finite element discretization. The only aspect that refers to a finite element analysis is the fact that the deformed geometry is obtained as solution of the structural analysis and stresses are to be computed from this. The basic formulas for this are already given in Sections 3.1 and 3.2. However, in order to actually obtain meaningful stress values, a series of postprocessing steps is necessary, which is presented here in more detail. In the first part of this section, the procedure is presented for general continuum mechanics, and in the second part it is applied to the Kirchhoff-Love shell theory.

The result of the analysis provides the displacements and therefore, the geometry in the deformed configuration. As shown in equation (3.9), the Green-Lagrange strain coefficients are obtained by the metric coefficients in the deformed and reference configuration:

$$E_{ij} = \frac{1}{2}(g_{ij} - G_{ij})$$

The second Piola-Kirchhoff (PK2) stresses are obtained via the material law. As described in Section 3.2, the coefficients in the constitutive equation are usually referred to a local Cartesian coordinate system, which is denoted by a bar ($\bar{\ }$) over the coefficients:

$$\overline{S}^{ij} = \overline{C}^{ijkl}\overline{E}_{kl} \tag{3.55}$$

The PK2 stresses S^{ij} are obtained from $\overline{S}^{ij}$ by a transformation from the local Cartesian basis $\mathbf{E}_i \otimes \mathbf{E}_j$ (reference configuration) to the covariant basis $\mathbf{G}_i \otimes \mathbf{G}_j$ (reference configuration), according to equation (2.58):

$$S^{kl} = \overline{S}^{ij}(\mathbf{G}^k \cdot \mathbf{E}_i)(\mathbf{E}_j \cdot \mathbf{G}^l) \tag{3.56}$$

As has been mentioned above, the PK2 stresses do not represent physical stresses, which are described by the Cauchy stress tensor. The relation between the second Piola-Kirchhoff stress tensor $\mathbf{S}$ and the Cauchy stress tensor $\boldsymbol{\sigma}$ is given in equation (3.17) and repeated here for clarity:

$$\mathbf{S} = \det\mathbf{F} \cdot \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T}$$

Expressing the stress tensors with coefficients and bases, it can be written as:

$$S^{ij}\mathbf{G}_i \otimes \mathbf{G}_j = \det\mathbf{F} \cdot \mathbf{F}^{-1} \left(\sigma^{ij}\mathbf{g}_i \otimes \mathbf{g}_j\right) \mathbf{F}^{-T} \tag{3.57}$$

With the transformation rule given in equation (3.5):

$$\mathbf{G_i} = \mathbf{F}^{-1}\mathbf{g}_i$$

equation (3.57) can be rewritten:

$$S^{ij}\mathbf{G}_i \otimes \mathbf{G}_j = \det\mathbf{F} \cdot \sigma^{ij}\mathbf{G}_i \otimes \mathbf{G}_j \tag{3.58}$$

In equation (3.58) there is the same basis on the left and the right side. Therefore, Cauchy and PK2 stress coefficients are related by:

$$\sigma^{ij} = \frac{1}{\det\mathbf{F}} S^{ij} \tag{3.59}$$

The stress coefficients σ^{ij} refer to the actual basis $\mathbf{g}_i \otimes \mathbf{g}_j$ which is (in general) not normalized and therefore these values are not expressed in normalized units (e.g. N/mm^2). Thus, they are transformed into a local Cartesian coordinate system in the actual configuration $\mathbf{e}_k \otimes \mathbf{e}_l$, according to equation (2.58):

$$\widehat{\sigma}^{kl} = \sigma^{ij}(\mathbf{e}^k \cdot \mathbf{g}_i)(\mathbf{g}_j \cdot \mathbf{e}^l) \tag{3.60}$$

The stress coefficients $\widehat{\sigma}^{kl}$ finally are real physical stress values.

This procedure of transformations for the stresses can be applied equivalently to the stress resultants in shell analysis. This means that $\bar{n}^{\alpha\beta}$ and $\bar{m}^{\alpha\beta}$ can be computed according to equations (3.45) and (3.46) and then be transformed into physical values $\widehat{n}^{\alpha\beta}$, $\widehat{m}^{\alpha\beta}$ by the procedure shown above. With normal forces and bending moments, the in-plane

stresses $\widehat{\sigma}^{\alpha\beta}$ can be determined at any point in the shell continuum
Since transverse shear strains are neglected in the Kirchhoff-Love shell theory, the transverse shear forces $\bar{q}^{\alpha}$ cannot be obtained via the material law, but from equilibrium considerations. For shallow shells, the shear force is defined by the derivatives of the bending moments [BK85a]:

$$\bar{q}^{\alpha} = \frac{\partial \bar{m}^{\alpha\alpha}}{\partial s_{\alpha}} + \frac{\partial \bar{m}^{\alpha\beta}}{\partial s_{\beta}} \tag{3.61}$$

where s_{α} are the arc length parameters on the surface, defined by:

$$\mathrm{d}s_{\alpha} = \sqrt{a_{\alpha\alpha}}\,\mathrm{d}\theta^{\alpha} \tag{3.62}$$

with $a_{\alpha\alpha}$ as the metric coefficients of the middle surface (equation (3.27)). Thus, equation (3.61) can be rewritten:

$$\bar{q}^{\alpha} = \frac{\partial \bar{m}^{\alpha\alpha}}{\partial \theta^{\alpha}} \sqrt{a_{\alpha\alpha}}^{-1} + \frac{\partial \bar{m}^{\alpha\beta}}{\partial \theta^{\beta}} \sqrt{a_{\beta\beta}}^{-1} \tag{3.63}$$

Similar to normal forces and bending moments, the shear forces $\bar{q}^{\alpha}$ need to be transformed into physical values $\widehat{q}^{\,\alpha}$ as shown above. The transverse shear stresses $\widehat{\sigma}^{3\alpha}$ in the shell continuum can be computed from the transverse shear force $\widehat{q}^{\,\alpha}$, where a quadratic stress distribution through the shell thickness is assumed.

Chapter 4

Isogeometric analysis

4.1 Motivation

The term “isogeometric analysis” was defined by Hughes et al. [HCB05, CHB09] and means that the analysis model uses the same mathematical description as the geometry model. It is an enhancement to “isoparametric analysis”. The isoparametric concept states that the same functions are used to describe the initial geometry and the unknown solution field, e.g. displacements [ZTZ05]. Note that in this context, “initial geometry” refers to the initial geometry of the analysis model. The isoparametric concept is an important prerequisite for the correct treatment of rigid body motions. In traditional FEA, low-order, mostly linear, Lagrange polynomials are used as basis functions for the analysis, whereas computer aided geometry modeling is based on techniques like spline-functions and subdivision surfaces. As a consequence, a model conversion is necessary if a geometry designed in a CAD program is to be analyzed by FEA. For analysis, the geometry is converted into a mesh of finite elements, which is why this process is called “meshing”. This model conversion causes a series of problems. The most obvious problem is that due to the model conversion, geometric information is lost. The finite element geometry is only an approximation to the original geometry and the quality of this approximation depends on the mesh density. However, an exact description of the geometry is crucial if small geometric imperfections can decide about the overall structural behavior, like in buckling of thin shells. The second aspect is the time impact of meshing, which is a serious problem in industrial applications, especially since the whole process has to be redone every time a mesh needs to be refined or modified.

The idea of isogeometric analysis is that the functions used for the geometry description in CAD are adopted by the analysis for the geometry and the solution field (isoparametric concept). By this, the whole process of meshing can be omitted and the two models for design and analysis merge into one.

4.2 NURBS-based Isogeometric Analysis

Basically, all functions used in computer aided design could be used as basis for isogeometric analysis, provided that they fulfill the necessary conditions for basis functions, such as linear independence and partition of unity. NURBS (Non-Uniform Rational B-Splines) are the most widespread technology in today's CAD programs and they fulfill the necessary conditions mentioned above, therefore they are adopted for analysis. Alternative methods which have already been applied to analysis are subdivision surfaces [COS00, CO01] and T-Splines [BCC$^+$10, DJS10, UY09, UKY08, BBDL$^+$10].

Similar to finite element analysis, isogeometric analysis works with elements. For using NURBS-based isogeometric analysis, there are two possible definitions of an element. In the approach that is used in this thesis, the NURBS elements are defined by the knot spans of the knot vectors. This means that the domain consists of a couple of NURBS patches and each patch is a subdomain that is divided into elements by the knot vectors. An alternative definition is that the whole patch is considered as one NURBS element. There are reasons and counter-arguments for both approaches, and in the end it is only a question of terminology. In the following, the definitions for isogeometric NURBS-elements are presented, as well as their consequences for analysis and the differences to classical finite element analysis.

4.2.1 Elements

As shown in Chapter 2, a NURBS patch is defined over a parametric domain, which is divided into intervals by the knot vectors. These intervals are defined as elements. The reason for this definition is that inside a knot interval, B-Spline basis functions are polynomials and therefore Gauss quadrature can be used for integration on element level. NURBS basis functions are not polynomials but rational polynomials. Therefore, the integration with Gauss quadrature is only approximative for NURBS basis functions. But it is important to remind that not the basis functions per se are integrated, but the element formulation, which in general results in the integration of rational polynomials anyhow. For example, the standard formulation of a solid element includes the integration over the inverse Jacobian determinant. Only in the case of rectangular or paralleloid elements, the integrand is a non-rational polynomial. The use of Gauss quadrature for NURBS elements has been investigated and proven as reliable in the literature [HCB05] as well as in the benchmark examples in this thesis. Hughes et al. have described a new integration rule for NURBS that makes use of the higher continuities between elements, and therefore is more efficient than Gauss quadrature [HRS10]. For the examples presented in this thesis, however, Gauss integration has been used.

Equivalently to finite elements, a NURBS element is defined by a set of nodes and corresponding basis functions. The nodes are the NURBS control points. They carry the degrees of freedom for the analysis and boundary conditions are applied to them. Since the element formulation in this thesis is displacement-based, the degrees of freedom are the displacements of the control points. For three-dimensional structures this means that every control point has three degrees of freedom, namely the displacements in x-, y- and z-direction.

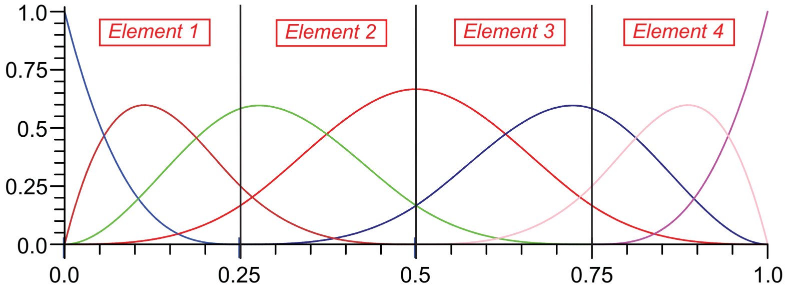


Figure 4.1: Isogeometric elements. The basis functions extend over a series of elements.

It is important to note that with this definition of elements, the basis functions are not confined to one element but extend over a series of elements, as illustrated in Figure 4.1. This is a very important difference to classical finite elements because it allows higher continuities of shape functions over the element boundaries. As in the p-version of the finite element method [SDR04, RDN$^+$04], the high-order nature of the basis functions generally results in higher accuracy compared to low-order elements. In contrast to p-version elements, NURBS-elements also have high-order continuities between elements, which is the basis for the element formulation presented in the next chapter. On the other hand, it means that the elements are interconnected and not independent of each other. The basis functions inside a knot span are defined by the *Cox-deBoor* recursion formula and depend on the neighboring knot spans, see equation (2.4). Therefore, it is not possible to define a single NURBS element without a complete NURBS patch. In this context, it is worth discussing the term “elements” since they are not independent, elementary parts that can be assembled arbitrarily to form a bigger model. Nevertheless, for implementation, these elements can be treated exactly in the same way as classical finite elements. The stiffness matrix, for example, is evaluated on element level and assembled to the global stiffness matrix. The only difference is the use of different shape functions. The fact that the corresponding nodes, i.e. control points, usually lie outside the element, is solely a consequence of the used basis functions and does not make any difference in the treatment of these elements in a finite element code.

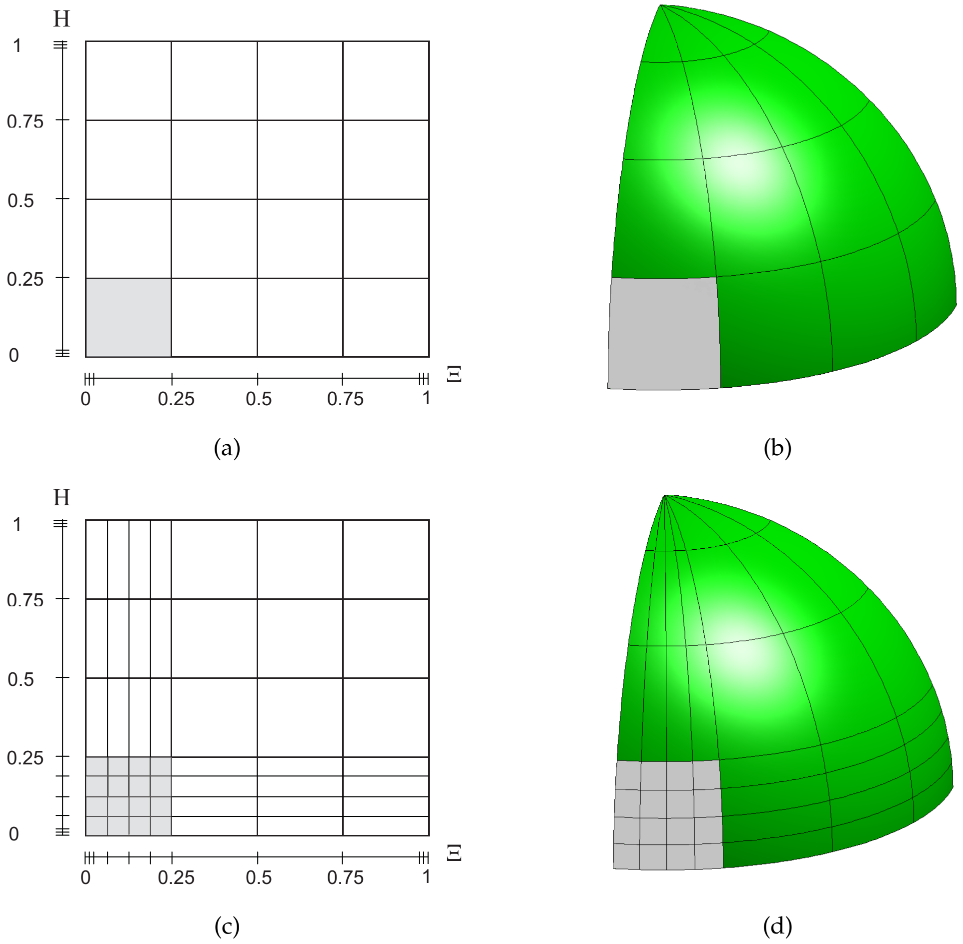
Many locking phenomena in structural analysis, and especially in shell and plate analysis, are a consequence of the low-order basis functions that cannot correctly represent the physical behavior [Kos04, KBCB04, KJ03, Cha04, Lee05]. Since NURBS are higher order functions, these locking effects can be avoided ab initio.

The following important properties of NURBS as basis for analysis are summarized:

- The basis functions fulfill the requirements of linear independence and partition of unity. They have a local support, depending on the polynomial degree.
- Basis functions have higher-order continuities over element boundaries.
- Degrees of freedom are defined on the control points.
- The isoparametric concept is used.
- Rigid body motions are treated correctly (zero strains) due to the affine covariance property of NURBS.
- Locking effects stemming from low-order basis functions can be precluded ab initio.

4.2.2 Mesh Refinement

The methods of knot insertion and order elevation presented in Chapter 2.2.4, are used for mesh refinement in the analysis. Here, knot insertion corresponds to h-refinement of classical FEA since the number of elements is increased and order elevation corresponds to p-refinement. An important difference to refinement in classical FEA is that the refinement for NURBS does not change the geometry. This means that in each refinement step, the geometry is represented exactly and therefore a refined mesh can be further refined without the necessity of going back to the original model. As described in Chapter 2, a NURBS curve can be refined by knot insertion, where knots can be inserted arbitrarily. This means that local refinement for a NURBS curve is possible. For NURBS surfaces however, a knot inserted in ξ-directions extends over the whole patch in η-direction and vice versa, see Figure 4.2. In (a), the unrefined parametric space is shown. The first element, which is highlighted, shall be refined. Figure (b) shows the corresponding physical model. In (c), the refined parametric space is depicted and in (d) the corresponding physical model. As can be seen, the inserted knots extend over the whole patch in the respective direction. Therefore, pure local refinement is not possible for NURBS patches [CHR07]. The reason for this is the tensor product structure of NURBS surfaces. An alternative could be the use of T-Splines, which are not confined to a tensor product structure. Knot refinement and order elevation can also be combined, but it is important to know that the sequence is not interchangeable. While order elevation preserves all continuities, the insertion of a knot decreases the continuity at this location. This means that performing order elevation before knot insertion yields higher continuities in the refined geometry than vice versa. For better understanding, this is explained on a simple example. A curve of polynomial degree $p=2$ shall be order elevated to $p=3$ and a knot shall be inserted at a location $\overline{\xi}$. Now, the two possible cases are considered:



(a) (b) (c) (d)

Figure 4.2: "Local" refinement of a NURBS element. The Element is highlighted. (a) Unrefined parametric space. (b) Unrefined physical model. (c) Refined parametric space. (d) Refined physical model.

Case A: First, order elevation is performed, the polynomial degree is now $p = 3$. Then, the knot is inserted. The continuity at this knot is $C^{p-k} = C^{3-1} = C^2$.

Case B: First, the knot is inserted. The continuity at this knot is then $C^{p-k} = C^{2-1} = C^1$. Then, order elevation is performed. Since order elevation preserves all continuities, the continuity at $\overline{\xi}$ remains C^1.

Unless low continuities are desired, order elevation is always performed before knot insertion. This is often referred to as *k-refinement* [HCB05, CHR07, CHB09].

Chapter 5

The NURBS-based Kirchhoff-Love shell

5.1 Element Formulation

In Section 3.2, the Kirchhoff-Love shell theory was presented and the corresponding differential equations were formulated in the weak form. In this chapter, these equations are derived for a discretized system. The variational variables are the nodal displacement variables. This formulation is generally valid for a displacement-based formulation of the Kirchhoff-Love theory and is not specific to NURBS. However, the advantage of the NURBS discretization is that an exact description of the surface is provided and therefore all geometric quantities which appear in the kinematic equations can be evaluated without further assumptions [KBLW09]. This is important in particular for the computation of curvatures.

The equilibrium condition of virtual work must be fulfilled for any arbitrary variation of the displacement variables δu_r:

$$\delta W = \frac{\partial W}{\partial u_r} \delta u_r = 0 \tag{5.1}$$

$$\frac{\partial W}{\partial u_r} = 0 \tag{5.2}$$

Equation (5.2) represents a nonlinear equation system which is linearized in order to solve it. For solving the linearized equation system the Newton-Raphson method is used:

$$\frac{\partial W}{\partial u_r} + \frac{\partial^2 W}{\partial u_r \partial u_s} \Delta u_s = 0 \tag{5.3}$$

The virtual work is defined as the sum of internal and external virtual work (3.20). The internal virtual work was formulated in equation (3.47) and is repeated here for a better understanding of the following formulas:

$$\delta W_{int} = - \int_A (\mathbf{n} : \delta \boldsymbol{\varepsilon} + \mathbf{m} : \delta \boldsymbol{\kappa}) \, \mathrm{d}A$$

The first derivative of the virtual work w.r.t. a displacement variable yields the residual force vector **R**:

$$R_r = \left(\frac{\partial W_{int}}{\partial u_r} + \frac{\partial W_{ext}}{\partial u_r} \right) = F_r^{int} + F_r^{ext} \tag{5.4}$$

where $\mathbf{F}^{ext}$ is the vector of external nodal loads, and $\mathbf{F}^{int}$ is the vector of internal nodal forces:

$$F_r^{int} = - \int_A \left(\mathbf{n} : \frac{\partial \boldsymbol{\varepsilon}}{\partial u_r} + \mathbf{m} : \frac{\partial \boldsymbol{\kappa}}{\partial u_r} \right) \mathrm{d}A \tag{5.5}$$

The second derivatives of the virtual work yield the stiffness matrix. Splitting into internal and external virtual work, the stiffness matrix **K** is:

$$K_{rs} = - \left(\frac{\partial^2 W_{int}}{\partial u_r \partial u_s} + \frac{\partial^2 W_{ext}}{\partial u_r \partial u_s} \right) = K_{rs}^{int} + K_{rs}^{ext} \tag{5.6}$$

The stiffness matrix $\mathbf{K}^{ext}$ is the derivative of the external loads w.r.t. the displacement variables, i.e. it is considered only in the case of displacement-dependent loads [SR84]. The internal stiffness matrix $\mathbf{K}^{int}$ is obtained by deriving the above term for the internal virtual work w.r.t. the displacement variables:

$$K_{rs}^{int} = \int_A \left(\frac{\partial \mathbf{n}}{\partial u_s} : \frac{\partial \boldsymbol{\varepsilon}}{\partial u_r} + \mathbf{n} : \frac{\partial^2 \boldsymbol{\varepsilon}}{\partial u_r \partial u_s} + \frac{\partial \mathbf{m}}{\partial u_s} : \frac{\partial \boldsymbol{\kappa}}{\partial u_r} + \mathbf{m} : \frac{\partial^2 \boldsymbol{\kappa}}{\partial u_r \partial u_s} \right) \mathrm{d}A \tag{5.7}$$

where the first two terms represent the membrane stiffness and the latter two the bending stiffness. Inserting equations (5.4)–(5.7) into (5.3) yields the equation system:

$$\mathbf{K}\, \Delta \mathbf{u} = \mathbf{R} \tag{5.8}$$

Equation (5.8) is a linearized step of the nonlinear problem, which is solved for $\Delta\mathbf{u}$, where $\mathbf{u}$ is the vector of nodal displacements.

For problems where only small deformations appear, it is appropriate to perform a geometrically linear analysis. This could be done by performing only one iteration step of the nonlinear analysis. However, for efficiency reasons it is better to use a geometrically linear formulation. In the linear case, the actual configuration is equal to the reference configuration. Since strain and change of curvature are computed by the difference between actual and reference configuration, they become zero, and also the normal forces and bending moments. As a consequence, the internal nodal forces (5.5) vanish and equation (5.7) reduces to:

$$K_{rs}^{int,lin} = \int_A \left(\frac{\partial \mathbf{n}}{\partial u_s} : \frac{\partial \boldsymbol{\varepsilon}}{\partial u_r} + \frac{\partial \mathbf{m}}{\partial u_s} : \frac{\partial \boldsymbol{\kappa}}{\partial u_r} \right) \mathrm{d}A \tag{5.9}$$

which leads to the typical equation system of a geometrically linear problem:

$$\mathbf{K}^{lin}\, \mathbf{u} = \mathbf{F}^{ext} \tag{5.10}$$

As can be seen in equation (5.9), the second derivatives disappear in the linear formulation which means an exponential gain in efficiency with increasing number of degrees of freedom, compared to the nonlinear formulation.

In the equations above, the derivatives of strains and curvatures w.r.t. the displacement variables $\partial u_r, \partial u_s$ are needed. The derivatives of the normal forces and moments can then be obtained via the material law. Therefore, all the kinematic quantities which lead to the computation of strains and curvatures as presented in Section 3.2 have to be derived w.r.t. $\partial u_r, \partial u_s$. This is shown in the following. Since the resulting expressions become very lengthy, intermediate steps are introduced to keep it clear and comprehensible.
We start with the position vector, which in a discretized system is defined by the linear combination of shape functions and discrete nodal values which are denoted by $(\hat{\ })$:

$$\mathbf{x} = \sum_i N^i \hat{\mathbf{x}}^i = \sum_i N^i (\hat{\mathbf{X}}^i + \hat{\mathbf{u}}^i) \tag{5.11}$$

where $\hat{\mathbf{x}}^i$ and $\hat{\mathbf{X}}^i$ are the nodal coordinates in the actual and reference configuration, respectively, and $\hat{\mathbf{u}}^i$ are the nodal displacements. Here, all parameters refer to the shell's middle surface.
The derivatives $\frac{\partial}{\partial u_r}$ will be denoted as $(\ldots)_{,r}$ for a shorter notation. Obviously, all quantities in the reference configuration are invariant to the variations ∂u_r and therefore the derivatives vanish:

$$\mathbf{x}_{,r} = \sum_i N^i (\hat{\mathbf{X}}^i + \hat{\mathbf{u}}^i)_{,r} = \sum_i N^i \hat{\mathbf{u}}^i{}_{,r} \tag{5.12}$$

First, all quantities needed for the membrane strains are derived. With the first derivative of the base vectors:

$$\mathbf{a}_{\alpha,r} = \mathbf{x}_{,\alpha\,,r} = \sum_i N^i{}_{,\alpha}\, \hat{\mathbf{u}}^i{}_{,r} \tag{5.13}$$

we get the derivation of the metric coefficients:

$$a_{\alpha\beta,r} = (\mathbf{a}_\alpha\, \mathbf{a}_\beta)_{,r} = \mathbf{a}_{\alpha,r}\, \mathbf{a}_\beta + \mathbf{a}_\alpha\, \mathbf{a}_{\beta,r} \tag{5.14}$$

and finally the derived strain coefficients:

$$\varepsilon_{\alpha\beta,r} = \frac{1}{2}(a_{\alpha\beta} - A_{\alpha\beta})_{,r} = \frac{1}{2} a_{\alpha\beta,r} \tag{5.15}$$

It is more efficient to compute and store the results of equations (5.12)-(5.14) rather than inserting these equations into equation (5.15), because they are needed again for other derived variables in the sequel.

In the next step, the second derivatives of the strain coefficients are computed, which is done in the same manner as for the first derivatives.

In the vector of nodal displacement variables $\hat{\mathbf{u}}^i$, the displacement variables appear linearly, and therefore the second derivatives $\hat{\mathbf{u}}^i{}_{,rs}$ vanish. As a consequence, the second derivatives of the position vector $\mathbf{x}_{,rs}$ vanish as well:

$$\mathbf{x}_{,rs} = \sum_i N^i \hat{\mathbf{u}}^i{}_{,rs} = \mathbf{0} \tag{5.16}$$

The same holds for the second derivatives of the base vectors:

$$\mathbf{a}_{\alpha,rs} = \sum_i N^i{}_{,\alpha}\, \hat{\mathbf{u}}^i{}_{,rs} = \mathbf{0} \tag{5.17}$$

Thus, the second derivatives of metric coefficients become:

$$\begin{aligned} a_{\alpha\beta,rs} &= \mathbf{a}_{\alpha,rs}\,\mathbf{a}_\beta + \mathbf{a}_{\alpha,r}\,\mathbf{a}_{\beta,s} + \mathbf{a}_\alpha\,\mathbf{a}_{\beta,rs} + \mathbf{a}_{\alpha,s}\,\mathbf{a}_{\beta,r} \\ &= \mathbf{a}_{\alpha,r}\,\mathbf{a}_{\beta,s} + \mathbf{a}_{\alpha,s}\,\mathbf{a}_{\beta,r} \end{aligned} \tag{5.18}$$

With this result, the second derivatives of the strain coefficients can be computed:

$$\varepsilon_{\alpha\beta,rs} = \frac{1}{2} a_{\alpha\beta,rs} \tag{5.19}$$

For the bending strains, the curvatures have to be derived w.r.t. the displacement variables ∂u_r. These are more involved than the derivations of the membrane strains and intermediate steps are introduced. These intermediate steps are also used in the implementation, i.e. they are computed and saved as temporary variables. This significantly increases efficiency because these variables appear several times in the subsequent formulas.

Recalling equations (3.35) and (3.28), the first derivative of the change in curvature w.r.t. ∂u_r is:

$$\kappa_{\alpha\beta,r} = (B_{\alpha\beta} - b_{\alpha\beta})_{,r} = -b_{\alpha\beta,r} = -(\mathbf{a}_{\alpha,\beta}\,\mathbf{a}_3)_{,r} \tag{5.20}$$

According to equation (5.20) the derivatives of $\mathbf{a}_{\alpha,\beta}$ and $\mathbf{a}_3$ are needed. Additionally to the unit normal vector $\mathbf{a}_3$, the not-normalized normal vector $\tilde{\mathbf{a}}_3$ and its length $\bar{a}_3$ are introduced.

$$\tilde{\mathbf{a}}_3 = \mathbf{a}_1 \times \mathbf{a}_2 \tag{5.21}$$

$$\bar{a}_3 = \sqrt{\tilde{\mathbf{a}}_3\,\tilde{\mathbf{a}}_3} \tag{5.22}$$

And therefore $\mathbf{a}_3$ can be written as:

$$\mathbf{a}_3 = \frac{\tilde{\mathbf{a}}_3}{\bar{a}_3} \tag{5.23}$$

These values are now derived w.r.t. ∂u_r.

$$\tilde{\mathbf{a}}_{3,r} = \mathbf{a}_{1,r} \times \mathbf{a}_2 + \mathbf{a}_1 \times \mathbf{a}_{2,r} \tag{5.24}$$

$$\bar{a}_{3,r} = \frac{1}{2\,\bar{a}_3}\, 2\,\tilde{\mathbf{a}}_3\,\tilde{\mathbf{a}}_{3,r} = \frac{\tilde{\mathbf{a}}_3\,\tilde{\mathbf{a}}_{3,r}}{\bar{a}_3} \tag{5.25}$$

$$\mathbf{a}_{3,r} = \frac{\tilde{\mathbf{a}}_{3,r}\,\bar{a}_3 - \tilde{\mathbf{a}}_3\,\bar{a}_{3,r}}{\bar{a}_3^2} \tag{5.26}$$

As can be seen, the result from (5.24) can be inserted into (5.25), and these results, together with (5.21) and (5.22), are used to compute the derivative $\mathbf{a}_{3,r}$ (5.26).
The derivative of $\mathbf{a}_{\alpha,\beta}$ reads as follows:

$$\mathbf{a}_{\alpha,\beta\,,r} = \sum N^i{}_{,\alpha\beta}\,\hat{\mathbf{u}}^i{}_{,r} \tag{5.27}$$

With (5.26) and (5.27) the derivative of the curvature coefficients can be computed:

$$b_{\alpha\beta,r} = \mathbf{a}_{\alpha,\beta\,,r}\;\mathbf{a}_3 + \mathbf{a}_{\alpha,\beta}\;\mathbf{a}_{3,r} \tag{5.28}$$

And finally:

$$\kappa_{\alpha\beta,r} = -b_{\alpha\beta,r} \tag{5.29}$$

For the second derivatives $\kappa_{\alpha\beta,rs}$, all the results from equation (5.24) to equation (5.29) have to be derived by ∂u_s. Starting with $\tilde{\mathbf{a}}_{3,rs}$:

$$\begin{aligned}\tilde{\mathbf{a}}_{3,rs} &= \mathbf{a}_{1,rs} \times \mathbf{a}_2 + \mathbf{a}_{1,r} \times \mathbf{a}_{2,s} + \mathbf{a}_{1,s} \times \mathbf{a}_{2,r} + \mathbf{a}_1 \times \mathbf{a}_{2,rs} \\ &= \mathbf{a}_{1,r} \times \mathbf{a}_{2,s} + \mathbf{a}_{1,s} \times \mathbf{a}_{2,r}\end{aligned} \tag{5.30}$$

the derivative $\bar{a}_{3,rs}$ can be computed:

$$\begin{aligned}\bar{a}_{3,rs} &= \frac{(\tilde{\mathbf{a}}_{3,rs}\,\tilde{\mathbf{a}}_3 + \tilde{\mathbf{a}}_{3,r}\,\tilde{\mathbf{a}}_{3,s}\,)\,\bar{a}_3 - \tilde{\mathbf{a}}_{3,r}\,\tilde{\mathbf{a}}_3\,\bar{a}_{3,s}}{\bar{a}_3^2} \\ &= \frac{\tilde{\mathbf{a}}_{3,rs}\,\tilde{\mathbf{a}}_3 + \tilde{\mathbf{a}}_{3,r}\,\tilde{\mathbf{a}}_{3,s}}{\bar{a}_3} - \frac{(\tilde{\mathbf{a}}_{3,r}\,\tilde{\mathbf{a}}_3)\,(\tilde{\mathbf{a}}_{3,s}\,\tilde{\mathbf{a}}_3)}{\bar{a}_3^3}\end{aligned} \tag{5.31}$$

and finally the second derivative of the normal unit vector:

$$\begin{aligned}\mathbf{a}_{3,rs} &= \frac{\tilde{\mathbf{a}}_{3,rs}\,\bar{a}_3 - \tilde{\mathbf{a}}_{3,r}\,\bar{a}_{3,s}}{\bar{a}_3^2} - \frac{(\tilde{\mathbf{a}}_{3,s}\,\bar{a}_{3,r} + \tilde{\mathbf{a}}_3\bar{a}_{3,rs}\,)\bar{a}_3^2 - \tilde{\mathbf{a}}_3\bar{a}_{3,r}\,2\bar{a}_3\bar{a}_{3,s}}{\bar{a}_3^4} \\ &= \frac{\tilde{\mathbf{a}}_{3,rs}}{\bar{a}_3} - \frac{\tilde{\mathbf{a}}_{3,r}\,\bar{a}_{3,s}}{\bar{a}_3^2} - \frac{\tilde{\mathbf{a}}_{3,s}\,\bar{a}_{3,r}}{\bar{a}_3^2} - \frac{\tilde{\mathbf{a}}_3\bar{a}_{3,rs}}{\bar{a}_3^2} + \frac{\tilde{\mathbf{a}}_3 2\bar{a}_{3,r}\,\bar{a}_{3,s}}{\bar{a}_3^3}\end{aligned} \tag{5.32}$$

All quantities in equation (5.32) have been computed in the previous formulas.
Similar to $\mathbf{x}_{,rs}$ and $\mathbf{a}_{\alpha,rs}$, the second derivatives $\mathbf{a}_{\alpha,\beta\,,rs}$ vanish:

$$\mathbf{a}_{\alpha,\beta\,,rs} = \sum_i N^i{}_{,\alpha\beta}\,\hat{\mathbf{u}}^i{}_{,rs} = \mathbf{0} \tag{5.33}$$

Finally, the second derivatives of the curvature coefficients can be computed:

$$\begin{aligned}b_{\alpha\beta,rs} &= \mathbf{a}_{\alpha,\beta\,,rs}\,\mathbf{a}_3 + \mathbf{a}_{\alpha,\beta\,,r}\,\mathbf{a}_{3,s} + \mathbf{a}_{\alpha,\beta\,,s}\,\mathbf{a}_{3,r} + \mathbf{a}_{\alpha,\beta}\,\mathbf{a}_{3,rs} \\ &= \mathbf{a}_{\alpha,\beta\,,r}\,\mathbf{a}_{3,s} + \mathbf{a}_{\alpha,\beta\,,s}\,\mathbf{a}_{3,r} + \mathbf{a}_{\alpha,\beta}\,\mathbf{a}_{3,rs}\end{aligned} \tag{5.34}$$

and with those, the second derivatives $\kappa_{\alpha\beta,rs}$ are obtained:

$$\kappa_{\alpha\beta,rs} = -b_{\alpha\beta,rs} \tag{5.35}$$

Equations (5.15) and (5.29) contain the first derivatives of $\boldsymbol{\varepsilon}$ and $\boldsymbol{\kappa}$ w.r.t. a displacement variable. The derivatives of the normal forces $\frac{\partial \mathbf{n}}{\partial u_s}$ and bending moments $\frac{\partial \mathbf{m}}{\partial u_s}$ are obtained via the material law:

$$\frac{\partial \mathbf{n}}{\partial u_s} = t \cdot \mathbf{C} : \frac{\partial \boldsymbol{\varepsilon}}{\partial u_s} \tag{5.36}$$

$$\frac{\partial \mathbf{m}}{\partial u_s} = \frac{t^3}{12} \cdot \mathbf{C} : \frac{\partial \boldsymbol{\kappa}}{\partial u_s} \tag{5.37}$$

Finally, with equations (5.15), (5.19), (5.29), (5.35), (5.36), and (5.37), all quantities are provided which are needed to compute the stiffness matrix as in equation (5.7) and the internal nodal forces as in (5.5):

$$F_r^{int} = -\int_A \left(\mathbf{n} : \frac{\partial \boldsymbol{\varepsilon}}{\partial u_r} + \mathbf{m} : \frac{\partial \boldsymbol{\kappa}}{\partial u_r} \right) \mathrm{d}A$$

$$K_{rs}^{int} = \int_A \left(\frac{\partial \mathbf{n}}{\partial u_s} : \frac{\partial \boldsymbol{\varepsilon}}{\partial u_r} + \mathbf{n} : \frac{\partial^2 \boldsymbol{\varepsilon}}{\partial u_r \partial u_s} + \frac{\partial \mathbf{m}}{\partial u_s} : \frac{\partial \boldsymbol{\kappa}}{\partial u_r} + \mathbf{m} : \frac{\partial^2 \boldsymbol{\kappa}}{\partial u_r \partial u_s} \right) \mathrm{d}A$$

It is important to point out that all the formulas above are generally valid for a displacement-based Kirchhoff-Love shell formulation and not specific to the discretization with NURBS. Since the curvatures $\boldsymbol{\kappa}$ contain the second derivatives of the geometry description, the variational index is two, and therefore C^1 continuity between elements is necessary to fulfill the compatibility conditions. This is not generally possible using Lagrange polynomials as basis functions. For bilinear elements, which are the most common shell elements, it is obvious that C^1 continuity between elements is not possible. With higher order Lagrange elements, C^1 continuity between all elements can only be obtained for regular shapes but not for arbitrary free-form surfaces. This is the reason why Kirchhoff-based formulations are less widespread in finite elements than Reissner-Mindlin elements. In the Reissner-Mindlin shell formulation, where the transverse shear strain is taken into account, the variational index is one and therefore C^0 continuity between elements is sufficient. It is also possible to use the Kirchhoff kinematics, avoiding the second derivatives. In this case, the computation of the curvature requires considering a patch of elements surrounding the element to be evaluated [OnZ00, OnF05, LWB07]. As has been shown in the previous chapters, NURBS basis functions offer C^1 and higher continuities between elements and therefore the Kirchhoff-Love shell formulation as presented above, can be implemented without further modifications.

5.2 Treatment of Rotational Boundary Conditions

It is an advantage of the NURBS-based Kirchhoff-Love shell that it needs neither rotational degrees of freedom nor nodal directors. However, there are cases where the rotations are to be described at the shell's boundary, e.g. in the case of clamped edges, symmetry conditions, and the coupling of NURBS patches. In all these cases, in addition to the displacement also the slope of the surface has to be prescribed along the boundary. As has been explained in Section 2.2.2.3, the slopes at the boundary of a NURBS surface are determined by the first two rows of control points from this boundary. For a clamped edge, this means that the first two rows of control points from this edge have to be fixed. It is important to note that this does not introduce a support inside the shell, but only concerns the support conditions at the boundary. This will be shown on an example in the next chapter. For symmetry conditions, the first two rows of control points along the edge have to be coupled so that the slope remains constant. This will be applied to the benchmark examples of the shell obstacle course in the next chapter. The same idea holds for the coupling of NURBS patches to multipatch structures which will be treated in detail in Chapter 7.

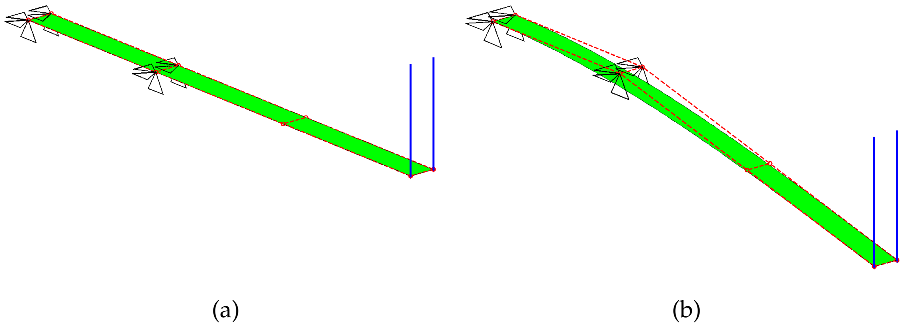
Chapter 6

Benchmarking

In this chapter, the presented shell formulation is tested on a series of benchmark examples. Tests for geometrically linear and nonlinear analyses are performed and, at the same time, modeling aspects such as clamped edges are presented.

6.1 Cantilever Plate

The first example is a simple cantilever plate and it serves to demonstrate the modeling of a clamped edge. Figure 6.1 shows the setup. The plate is modeled by one element with cubic basis functions in the length direction and linear in the width direction. The control net is shown by dashed lines. The plate is subjected to a vertical load at the right edge and clamped at its left edge. For constraining the rotation, the second row of control points is fixed, as can be seen in Figure 6.1. A linear computation is performed in order to compare the result to the one from linear beam theory $w_{beam} = \frac{PL^3}{3EI}$. As the analytical solution of the deformation is a cubic function, already one element with cubic basis functions in the length direction yields the exact solution. Due to this coarse discretization, the second row of control points, which is fixed for constraining the rotation, is quite far inside the


(a)

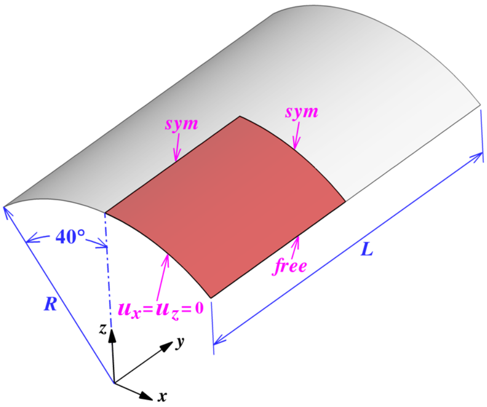
(b)

Figure 6.1: Cantilever plate under vertical load. (a) Undeformed. (b) Deformed. Fixing the second (from the left) row of control points keeps the tangent at the left end fixed, which represents the clamped support.

plate. It is important to note that fixing these control points does not fix the displacements inside the plate but only the rotation at the boundary.

6.2 Shell Obstacle Course

For testing the element's robustness and accuracy in complex strain states, the well-known shell obstacle course proposed by Belytschko et al. [BSL$^+$85] is applied. It consists of three geometrically linear problems, namely the Scordelis-Lo roof, the pinched cylinder with diaphragms and the hemispherical shell. For all three problems, a convergence study is performed where different mesh refinements (4, 8, 16 and 32 elements per edge) and different polynomial orders (p=2, 3, 4, 5) are used. In all problems, small displacements occur and therefore, geometrically linear computations are performed.



$$\begin{aligned} L &= 50.0 \\ R &= 25.0 \\ t &= 0.25 \\ E &= 4.32 \times 10^8 \\ \nu &= 0.0 \end{aligned}$$

Figure 6.2: Scordelis-Lo roof, problem setup. The roof is subjected to self weight of 90.0 per unit area. The ends are supported by rigid diaphragms.

6.2.1 Scordelis-Lo Roof

The Scordelis-Lo roof is a section of a cylindrical shell. It is supported by rigid diaphragms at its ends, whereas the side edges are free. The roof is subjected to a uniform gravity load. The problem setup and parameters are depicted in Figure 6.2. The vertical displacement at the midpoint of the side edge is given as the reference solution. Due to symmetry, only one quarter of the geometry is modeled. Figure 6.3 shows the convergence of the displacement for the different polynomial orders, and the reference solution. The converged solution is $u = 0.3006$ which is slightly lower than the reference value $u_{ref} = 0.3024$ from [BSL$^+$85, MH85]. For all polynomial orders, the convergence is quite fast. For quartic and quintic shape functions, already the second refinement step yields the converged solution. In Figure 6.4, the resulting deformation of the roof is shown. For visualization, the complete roof has been assembled and a scaling factor of 10 has been applied to the displacements.

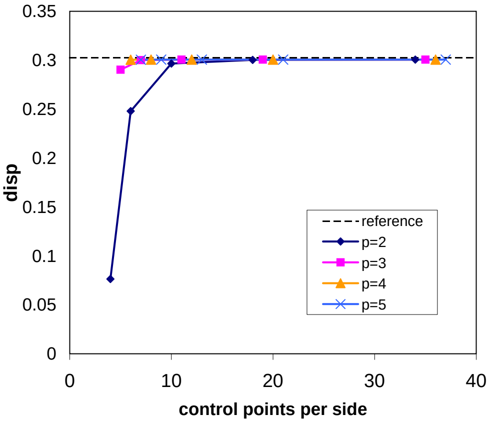


Figure 6.3: Scordelis-Lo roof. Convergence of the vertical displacement at the midpoint of the free edge.

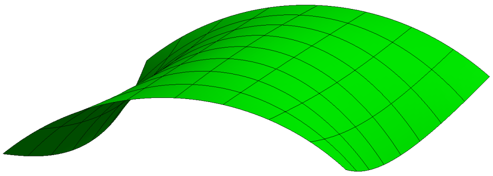
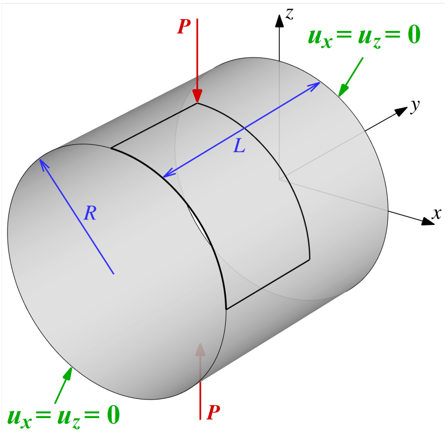


Figure 6.4: Scordelis-Lo roof, resulting deformation. The complete roof has been assembled for visualization and the displacements are scaled by a factor of 10.

6.2.2 Pinched Cylinder

The second problem of the shell obstacle course is a cylinder supported by rigid diaphragms at the ends and subjected to two opposite point loads in the middle, as shown in Figure 6.5. The reference solution $u_{ref} = 1.8248 \cdot 10^{-5}$ is given as the radial displacement under the point loads [BSL^{+}85]. Due to symmetry, only one eighth of the geometry is modeled. In Figure 6.6, the convergence curves are shown. The converged solution is $u = 1.8264 \cdot 10^{-5}$. Figure 6.7 shows the deformation of the pinched cylinder. For visualization, half of the cylinder has been assembled and the displacements are scaled by a factor of 3×10^{6}.



$$\begin{aligned} L &= 600.0 \\ R &= 300.0 \\ t &= 3.0 \\ E &= 3.0 \times 10^6 \\ \nu &= 0.3 \\ F &= 1.0 \end{aligned}$$

Figure 6.5: Pinched cylinder, problem setup. The cylinder is subjected to equal and opposite concentrated forces at its midspan. The ends are constrained by rigid diaphragms.

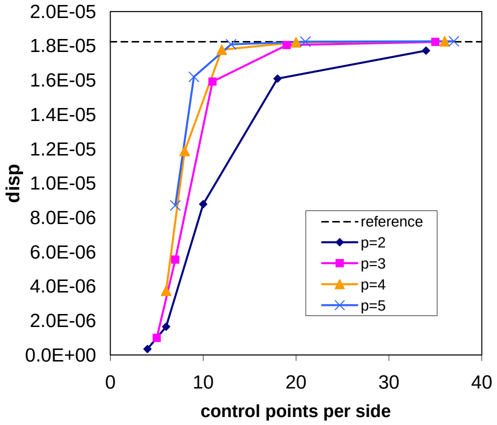


Figure 6.6: Pinched cylinder. Convergence of the displacement under the load.

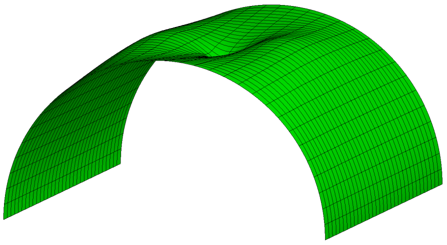
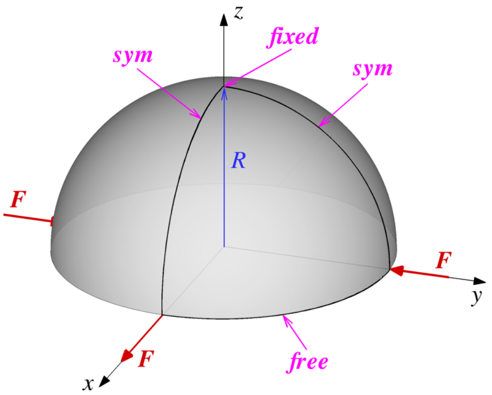


Figure 6.7: Pinched cylinder, resulting deformation. Half the cylinder has been assembled for visualization and the displacements are scaled by a factor of 3×10^6.



$$\begin{aligned} R &= 10.0 \\ t &= 0.04 \\ E &= 6.825 \times 10^7 \\ \nu &= 0.3 \\ F &= 2.0 \end{aligned}$$

Figure 6.8: Hemispherical shell, problem setup. The hemisphere is subjected to two diametrically opposite point loads. The bottom circumferential edge is free.

6.2.3 Hemispherical Shell

In this problem, a hemispherical shell is subjected to two opposite point loads. The bottom circumferential edge of the hemisphere is free. Figure 6.8 shows the problem setup. The reference solution is given as the radial displacement under the point loads. Due to symmetry, only one quarter of the geometry is modeled. Figure 6.9 shows the convergence curves. The converged solution $u = 0.0924$ is equal to the reference value given in [BSL$^+$85]. The deformed hemisphere, assembled for visualization with a scaling factor of 20, can be seen in Figure 6.10.

6.2.4 Stress Recovery

In the shell obstacle course literature [BSL$^+$85], there are only reference values for the displacements given, but not for stresses or stress resultants. In order to test the stress recovery as described in Section 3.4, the Scordelis-Lo roof is computed by a shell analysis in *Abaqus* [aba] and the results are compared to those of the isogeometric Kirchhoff-Love shell. Figure 6.11 shows the stress resultants $\widehat{n}^{11}, \widehat{m}^{11}, \widehat{q}^{1}$ obtained by an isogeometric shell analysis (θ^1 is the surface parameter in the short direction, and the notation $(\widehat{\ })$ refers to physical components, as described in Section 3.4). Figure 6.12 shows the corresponding results from *Abaqus*. These plots look slightly different compared to those in Figure 6.11, which is due to the fact that the two programs use different color interpolation schemes. Nevertheless, the good agreement in the distribution of the stress resultants can be observed and comparing Tables 6.1 and 6.2 also shows the good agreement of the maximum values.Special emphasis is put on the shear forces, Figure 6.11(c). As explained in Section 3.4, in the Kirchhoff theory, shear forces (and stresses) cannot be obtained from the corresponding strains since those are assumed to be zero. Instead, they can be obtained by the derivatives of the bending moments (equation (3.63)). Using higher-order functions

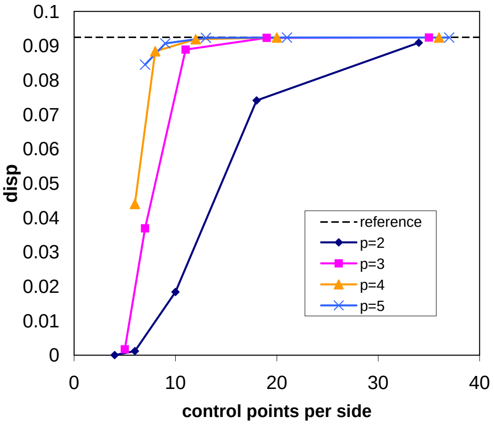


Figure 6.9: Hemispherical shell. Convergence of the radial displacement under the load.

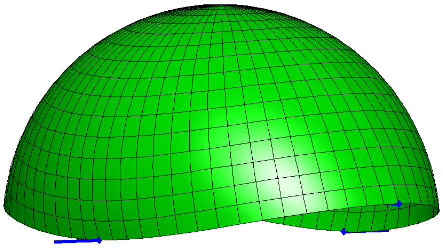


Figure 6.10: Hemispherical shell, resulting deformation. The complete hemisphere has been assembled for visualization and the displacements are scaled by a factor of 20.

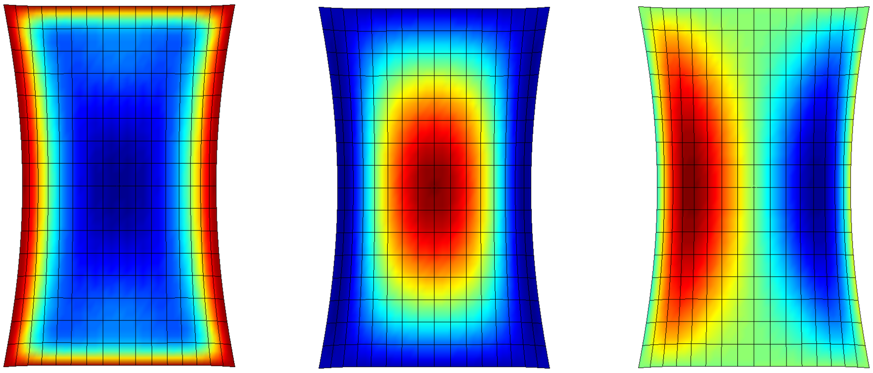
as basis functions, it is possible to evaluate the derivatives of the bending moments and to obtain a smooth distribution of the shear forces, as can be seen in Figure 6.11(c).

	$\widehat{n}^{11}$	$\widehat{m}^{11}$	$\widehat{q}^{1}$
min.	-3510	-91	-280
max.	25	2053	280

Table 6.1: Scordelis-Lo roof, maximum stress resultants by isogeometric Kirchhoff-Love shell analysis.

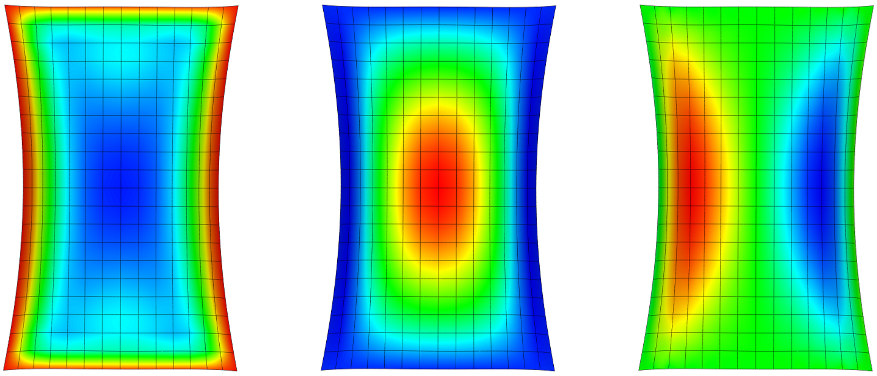
6.3 Benchmarks for Large Deformations

In the previous section, the shell formulation has successfully passed all problems of the shell obstacle course. Since these benchmark examples were all geometrically linear problems, the behavior under large deformations is tested in this section. Two examples are investigated, a plate under pure bending moment and a plate under pure twisting moment. The loading is chosen such that large deformations and rotations occur and the



(a) $\widehat{n}^{11}$ (b) $\widehat{m}^{11}$ (c) $\widehat{q}^{1}$

Figure 6.11: Stress resultants of the Scordelis-Lo roof. Isogeometric Kirchhoff-Love shell analysis.



(a) $\widehat{n}^{11}$ (b) $\widehat{m}^{11}$ (c) $\widehat{q}^{1}$

Figure 6.12: Stress resultants of the Scordelis-Lo roof. *Abaqus* shell analysis.

problems are solved by geometrically nonlinear analysis. For both examples an analytical solution is given as reference.

6.3.1 Plate bent to a Circle

In the first test, the cantilever plate from above is subjected to a constant bending moment. The moment is chosen such that a tip rotation of 360° is expected, i.e. the straight plate shall be bent to a circle. The bending moment is defined by $M = EI\kappa$, with E as the Young's modulus, I as the moment of inertia, and κ as the curvature which is the inverse of the radius ($\kappa = \frac{1}{R}$). The relation between the plate's length L and the radius R is $L = 2\pi R$. Thus, the corresponding moment is $M = \frac{2\pi EI}{L}$. Since there are no rotational degrees of freedom, the moment is modeled by pairs of forces. For this, an additional

	$\widehat{n}^{11}$	$\widehat{m}^{11}$	$\widehat{q}^{1}$
min.	-3417	-93	-278
max.	127	2079	278

Table 6.2: Scordelis-Lo roof, maximum stress resultants by *Abaqus* shell analysis.

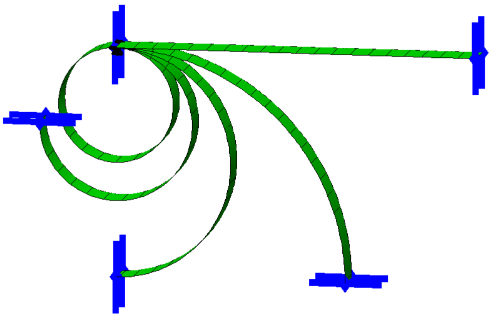


Figure 6.13: Plate under large deformation. A straight plate is bent to a circle.

row of control points is inserted close to the right edge by knot insertion. These forces need to be orthogonal to the shell, therefore they have to follow the deformation and the stiffness contribution from displacement-dependent loads has to be taken into account. The nonlinear calculation has been performed in four load steps. Quintic NURBS functions in the long direction have been used. Figure 6.13 shows the deformation of the plate in the four load steps. As can be seen, in each step the plate is deformed by an incremental rotation of 90° and the correct result is obtained.

Remark: In NURBS-based isogeometric analysis, only the control coordinates are variables, not the control weights. This means that in the resulting deformation, the control weights are those of the initial geometry. Therefore, the result in this example cannot be an exact circle, since a circle is defined by different control weights than a straight plate. Due to the fine discretization however, the approximation is very close.

The bent plate is tested further by increasing the moment to the double value, i.e. $M = \frac{4\pi EI}{L}$. For this case, a polynomial degree $p = 6$ in the long direction is used. Figure 6.14 shows the deformation for the additional four load steps after completing 360°. For a tip rotation larger than 360°, the plate overlaps itself. The figure shows that the overlapping parts follow precisely the same deformation such that the deformed geometry appears as one circle with decreasing radius, and the correct solution of 720° tip rotation is obtained.

6.3.2 Twisted Plate

For the next test, the plate is subjected to a pure twisting moment, which again is represented by a pair of follower forces with opposite directions. The problem setup is depicted in Figure 6.15(a). For this example, quadratic basis functions are used in the

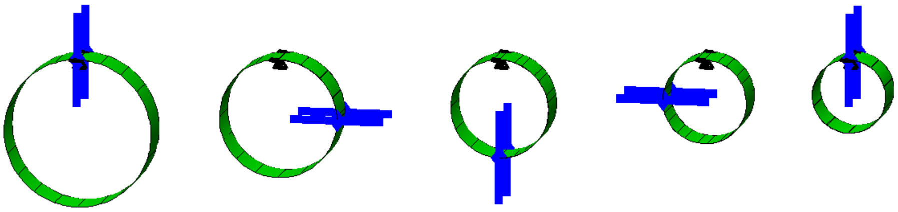
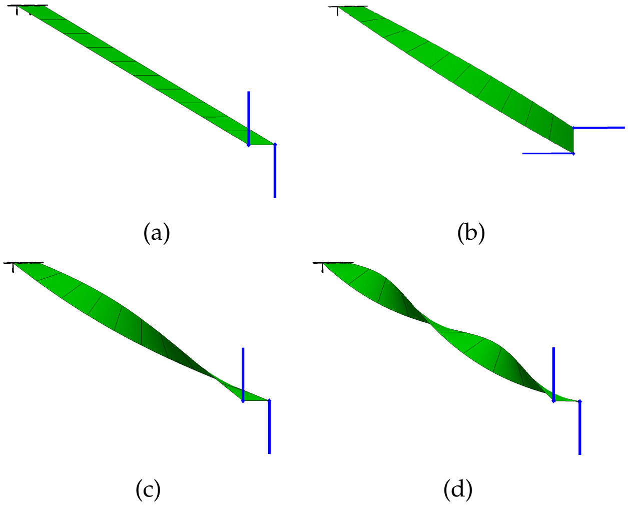


Figure 6.14: Plate under large deformation, case 2. The bending moment is increased corresponding to a tip rotation of 720°. For tip rotations $> 360°$, the plate overlaps itself.

width direction so that straight cross-sections, which are to be expected, are not defined ab initio. Quintic NURBS are employed in the length direction. The results are compared to the reference results from beam theory, i.e. $M_t = \theta \frac{GI_t}{L}$, where M_t is the torsional moment, θ is the twist angle at the tip, G is the shear modulus and L is the plate length. The torsion constant I_t is defined by $I_t = \frac{b\,t^3}{3}$, with b as the plate's width and t as the thickness. Moments are applied corresponding to expected tip rotations of 90°, 180°, and 360°. The results are collected in Figure 6.15(b)-(d). As can be seen, the correct deformations are obtained in all cases and furthermore, the plate cross-sections correctly remain straight in the deformed configurations.



(a) (b)

(c) (d)

Figure 6.15: Twisted plate under large rotations. (a) Initial configuration. (b) Twisted by 90°. (c) Twisted by 180°. (d) Twisted by 360°.

Chapter 7

Multipatches

In the previous chapter, it has been shown that NURBS are ideal as basis functions for a Kirchhoff-Love shell since they provide C^1 continuity between elements for arbitrarily curved surfaces. If the structure is made up of several NURBS patches, the question is how this continuity can be established between patches. If patches are connected with C^0 continuity only, it means that only the displacements at the interface are coupled, but not the slopes. In a mechanical sense, this represents a hinge connection where the respective bending moments cannot be transfered. The same applies if a C^0 continuity occurs inside a patch, due to knot multiplicity. In the following, different solutions are discussed for multipatches. However, all the methods apply to C^0 continuities inside patches as well.
For the connection of two patches, two basic cases need to be distinguished. There are smooth, i.e. G^1-continuous patches on the one hand, and patches forming a kink on the other hand. For G^1-continuous patch interfaces, this continuity must be maintained during analysis, whereas for patches enclosing a kink, the angle between the patches must be maintained.

7.1 Smooth Multipatches

In Section 2.3, the difference between parametric and geometric continuity has been discussed. The continuity between two patches for analysis has to be independent from the parametrization of the single patches and therefore, the geometric continuity is the one of interest. For parametric surfaces, G^1-continuity means that two surfaces joining at a common edge have a common tangent plane at each point along that edge. For B-Spline surfaces this results in the condition presented in equation (2.35), which is repeated here for clarity:

$$(\mathbf{P}^2_{2,j} - \mathbf{P}^1_{n,j}) = c \cdot (\mathbf{P}^1_{n,j} - \mathbf{P}^1_{n-1,j})$$

For NURBS, the same condition has to be applied to the homogeneous coordinates of the control points (2.36):

$$(\mathbf{P}^{w,2}_{2,j} - \mathbf{P}^{w,1}_{n,j}) = c \cdot (\mathbf{P}^{w,1}_{n,j} - \mathbf{P}^{w,1}_{n-1,j})$$

Since in isogeometric analysis only the physical coordinates are degrees of freedom, whereas control weights are invariant, the continuity condition for B-Splines (2.35) is employed. This is an approximation for NURBS and the continuity might not be fulfilled exactly at any point if the weights of the involved control points are different. However, as by refinement the differences in the weights of neighboring control points become minimal, the error is negligible.

For preserving G^1 continuity in the deformed configuration, equation (2.35) has to hold for the control points in the deformed configuration (denoted as $\overline{\mathbf{P}}$):

$$\begin{aligned}(\overline{\mathbf{P}}^2_{2,j} - \overline{\mathbf{P}}^1_{n,j}) &= c \cdot (\overline{\mathbf{P}}^1_{n,j} - \overline{\mathbf{P}}^1_{n-1,j}) \\ \overline{\mathbf{P}}^2_{2,j} &= (1+c) \cdot \overline{\mathbf{P}}^1_{n,j} - c \cdot \overline{\mathbf{P}}^1_{n-1,j}\end{aligned} \tag{7.1}$$

and therefore the same relation applies to the displacements:

$$\mathbf{u}^2_{2,j} = (1+c) \cdot \mathbf{u}^1_{n,j} - c \cdot \mathbf{u}^1_{n-1,j} \tag{7.2}$$

Equation (7.2) represents linear constraints between the system's degrees of freedom which can be fulfilled exactly by static condensation of the stiffness matrix, making $\mathbf{u}^2_{2,j}$ slaves of $\mathbf{u}^1_{n,j}$ and $\mathbf{u}^1_{n-1,j}$ [Fel04].

7.1.1 Cantilever Plate

The method is demonstrated on the example of a cantilever plate which has been used as a benchmark example in the previous chapter. There, the plate has been modeled by one element with cubic basis functions in the length direction and linear in the width direction (see Figure 6.1). That discretization has yielded the exact reference solution from linear beam theory, since this solution is a cubic function for the displacements. Now, the plate is modeled by two patches, each of them consisting of one cubic-linear element, see Figure 7.1. The control points which define the continuity at the interface are highlighted. They are coupled by the constraints of equation (7.2), and remain collinear in the deformed configuration. As a consequence, the resulting deformation is identical to the solution of a single patch with C^1 continuity everywhere.

In this example, the control grids of both patches are rectangular, i.e. the lines of control points which are to be coupled are orthogonal to the patch interface. It is important to note that this is not a necessary condition. In order to demonstrate this, the plate is now composed of two patches with a skew interface, see Figure 7.2(a). The continuity constraint is applied in the same way as in the previous example, and Figure 7.2(b) shows that the correct solution is obtained also in this case. A detailed study of the results yields that there is a very small error in the resulting displacements, which is due to the distortion in the mesh. The continuity across the interface, however, is maintained exactly.

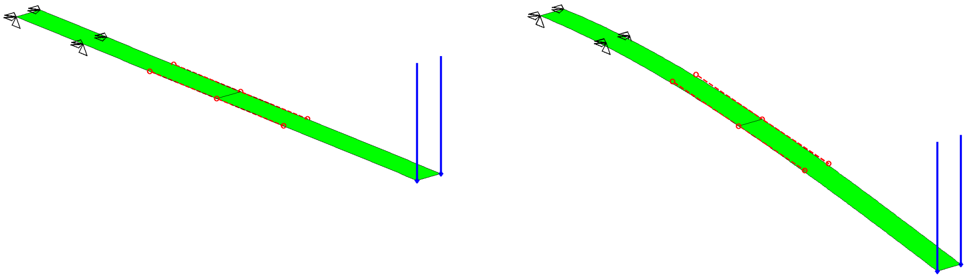


Figure 7.1: Cantilever plate, multipatch. Control points defining the continuity at the interface are highlighted.

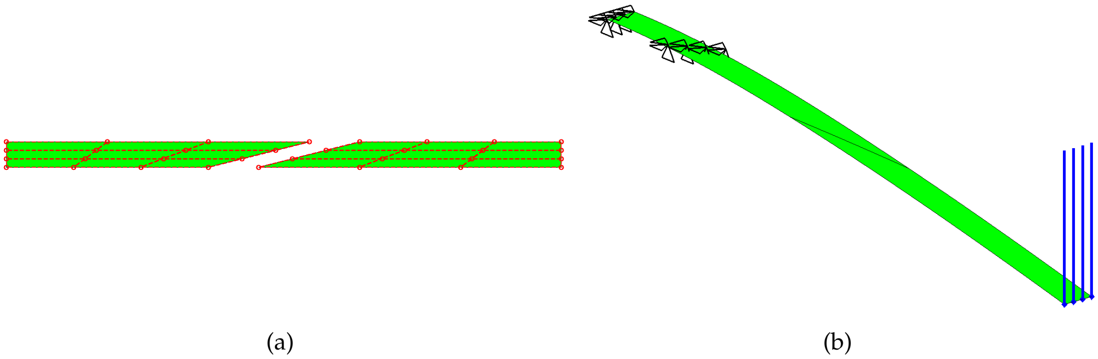
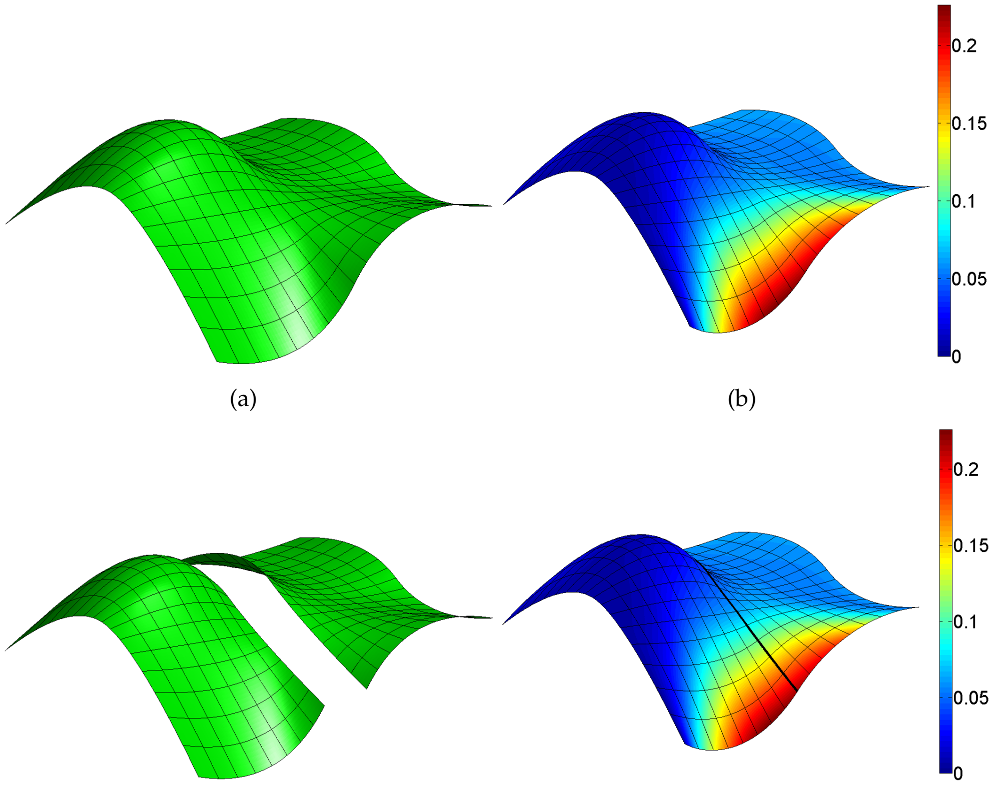


Figure 7.2: Cantilever plate, multipatch with a skew interface. (a) Two patches before assembly. (b) Resulting deformation.

7.1.2 Free Form Shell

In the previous two examples, simple plate geometries have been used to better illustrate the method. Since the continuity condition (2.35) defines the continuity for general three-dimensional B-Spline surfaces, the presented method is applicable to arbitrarily curved shells. This is exemplified on the free-form shell depicted in Figure 7.3. Figure (a) shows the free-form geometry. The shell is simply supported at the left (front) edge and the right (rear) edge, and is subjected to self-weight. The shell is computed once as a single patch and once as a multipatch geometry. For the multipatch case, the shell is divided into two parts as illustrated in Figure 7.3(c). A static analysis is performed for both cases and the results are compared to each other. Figure 7.3(b) shows the color plot of deformations for the single patch solution and (d) for the multipatch solution. As can be seen, the results are identical.



(a)

(b)

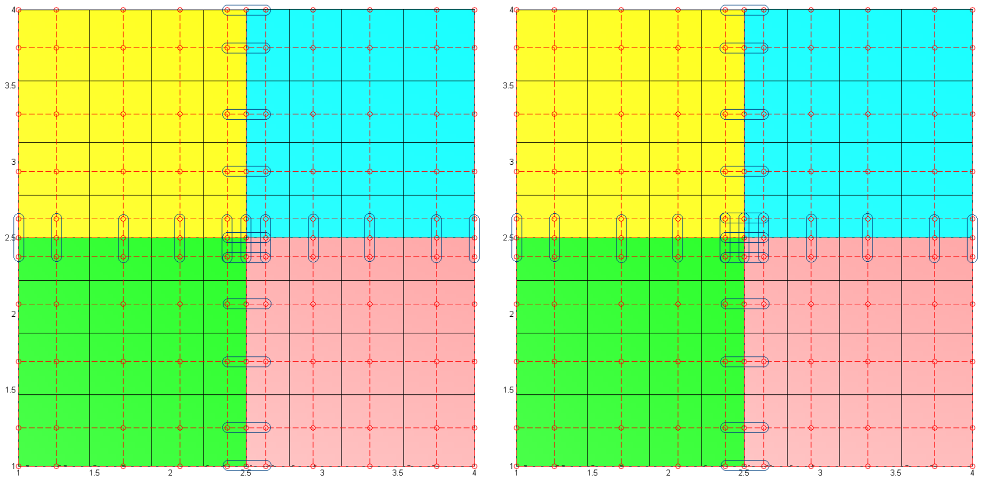
(c)

(d)

Figure 7.3: Free form shell under self-weight. (a) Single patch. (b) Deformation plot from the single patch solution. (c) Two patches, separated for visualization. (d) Deformation plot from the multi patch solution.

7.1.3 Automated Coupling of Multiple Patches

Due to the rectangular topology of NURBS control grids, it is easy to find the control points which have to be coupled for the connection of two patches, and the constraints between them are determined by their coordinates, see equations (2.35) and (7.2). Therefore, it is easy to find and set up all the constraints for merging two NURBS patches with G^1-continuity automatically. However, care has to be taken at points where more than two patches meet. This is illustrated in Figure 7.4. It shows a plate consisting of four patches. The four patches are indicated by different colors and the control grid is plotted as dashed lines. Figure 7.4(a) shows a correct coupling of control points for maintaining G^1-continuity everywhere on the plate. For having G^1-continuity between all patches at the middle point, the nine control points in the middle must lie on a bilinear surface. Since a bilinear surface has four degrees of freedom, four out of the nine control points are free variables, or master nodes, whereas the other five are slave nodes. Consequently, five constraint equations are necessary. The constellation in Figure 7.4(a) is one out of



(a)

(b)

Figure 7.4: Four patches meeting at one point. The control grid is depicted by dashed lines, the control points to be coupled are connected by small boxes. (a) Correct G^1 continuity at the middle point is maintained by five constraints. (b) Over-constraining at the middle point.

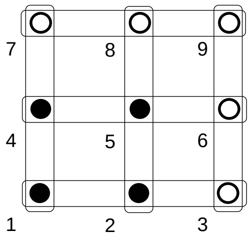


Figure 7.5: Detail sketch of the four-patch interface with over-constraining. Solid points are master nodes, non-solid points are slave nodes.

several possibilities for choosing the five constraints. However, performing the coupling automatically (always two patches at a time) results in six constraints as depicted in Figure 7.4(b). Including all six constraints by master-slave relations is an over-constraining which leads to a wrong result. This can be explained with the help of Figure 7.5. It shows a detail sketch of the nine nodes on the four-patch interface. Nodes $1, 2, 4$, and 5 are chosen as master nodes, indicated by solid points, whereas the rest are slave nodes. The boxes around the points illustrate the constraints. As can be seen, node 9 is a slave of nodes $7, 8$ and of $3, 6$. Nodes 3 and 7 however, are both slaves of node 1. As a consequence, the information (nodal stiffness) of node 9 is added to node 1 twice, once via node 3 and once via node 7. After solving, the solution (displacement) of node 1 is added to node 9 twice, which even leads to a gap in the resulting geometry. To obtain the correct solution, one of the constraints must be deleted where it is free to choose which one to delete. Figure 7.6 finally shows the result with the correct coupling applied. In (a), the

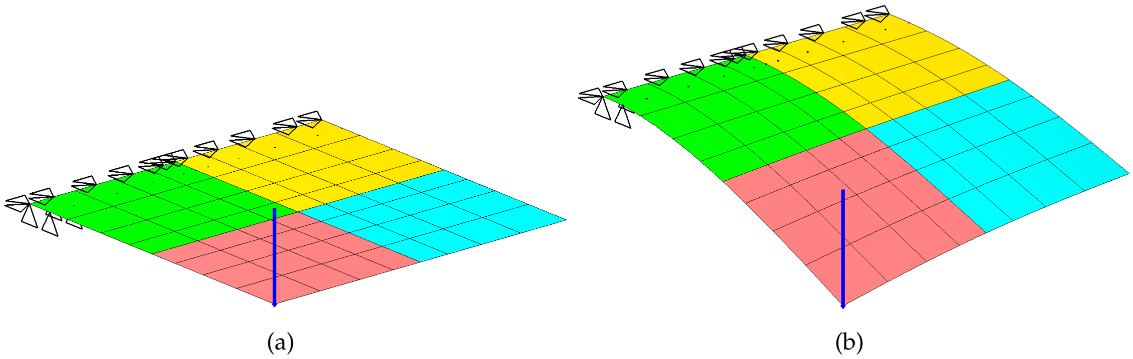


Figure 7.6: Cantilever plate consisting of four patches with single load. (a) Problem setup. (b) Resulting deformation.

problem setup is shown and in (b) the resulting deformation can be seen. The continuity in both directions is maintained correctly between all patches.

As a short summary, it must be said that it is no problem to correctly join more than two patches at one point, however, care has to be taken if an automated routine for patch coupling is used. Reference is also made to the "continuity elements" described in [Ble90].

7.2 The Bending Strip Method for Arbitrary Multipatches

While for smooth patch connections G^1 continuity has to be maintained, for patches meeting at a kink with a certain angle, the angle between the two patches has to be maintained in the deformed configuration. Similar to the smooth case, it is achieved by coupling the degrees of freedom defining this angle. These are the control points on the common edge and their neighboring points on each side. For each triple of control points $\mathbf{P}^2_{2,j}$, $\mathbf{P}^1_{n,j}$ and $\mathbf{P}^1_{n-1,j}$ along the edge, the angle spanned by these control points must remain constant during deformation. The angle can be expressed e.g. using the scalar product formula:

$$\alpha = \cos^{-1}\left(\frac{\left(\mathbf{P}^1_{n,j} - \mathbf{P}^1_{n-1,j}\right) \cdot \left(\mathbf{P}^2_{2,j} - \mathbf{P}^1_{n,j}\right)}{\left|\mathbf{P}^1_{n,j} - \mathbf{P}^1_{n-1,j}\right| \cdot \left|\mathbf{P}^2_{2,j} - \mathbf{P}^1_{n,j}\right|}\right) \tag{7.3}$$

which does not lead to a linear constraint relationship for the displacement degrees of freedom in the general case. Equation (7.3) is one possibility to express the angle spanned by three points. There are other alternatives (e.g. via the cross product), however, none of them results in a linear relationship between the affected degrees of freedom. As a result, the angle constraint cannot be enforced in a strong sense by direct coupling of degrees of freedom as in the G^1-continuous case. In this section, a method is presented that satisfies the constraints in a weak sense, which means the angle is kept constant approximately. The method can be applied to smooth multipatches as well, which can be regarded to form a kink of 180°. Thus, it can be used as a general method to handle multipatch structures.

The angle constraint is enforced by an additional NURBS patch with special properties. The triples of control points at the patch interface, consisting of the shared control point at the interface and one on each side, are extracted and used as control net for the additional patch, see Figure 7.7. The parametric domain of this patch consists of one quadratic element in the direction transverse to the interface and, for simplicity and computational efficiency, of $n-1$ linear elements along the interface where n is the number of control points in this direction. This additional patch, called "bending strip" [KBH$^+$10] has special material properties: it has no membrane stiffness and bending stiffness only in the direction transverse to the interface. By this, it is ensured that only the three control points $\mathbf{P}^2_{2,j}$, $\mathbf{P}^1_{n,j}$ and $\mathbf{P}^1_{n-1,j}$ in a triple across the edge are coupled with each other but no coupling between control points along the edge is made. This is a very important feature because it guarantees that no additional stiffness is added to the structure.

It is important to note that although the bending strips are implemented as additional structural patches, which makes implementation easy and efficient, they do not represent structural parts but only constraints on the angle between patches. The bending strips also have zero mass for the case of dynamic computations. Mathematically, the

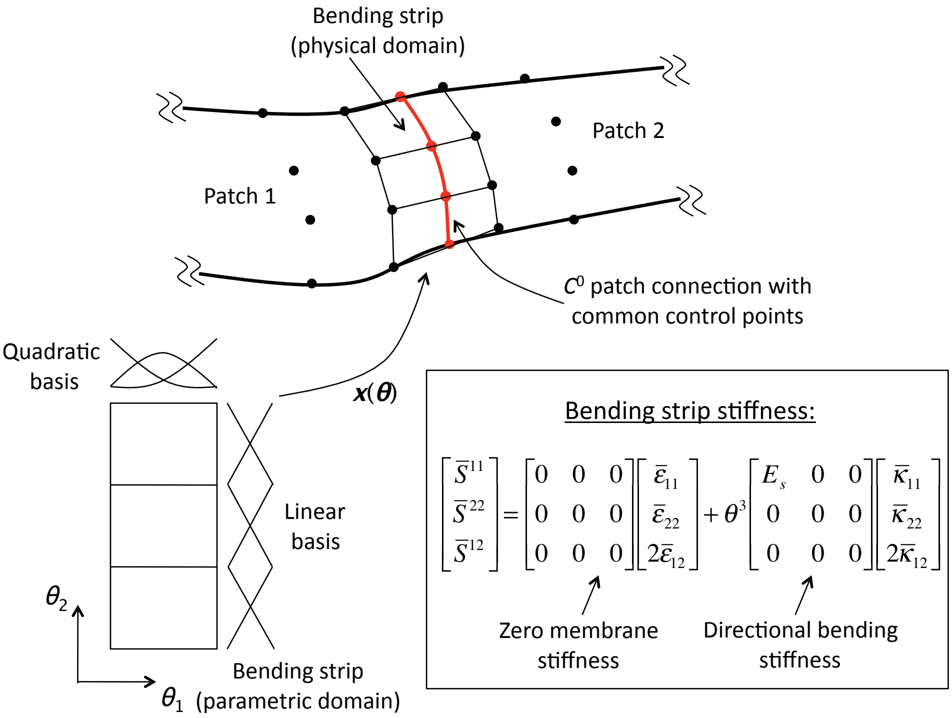

Bending strip
(physical domain)
Patch 2
Patch 1
C^0 patch connection with
common control points
Quadratic
basis
$x(\theta)$
Linear
basis
θ_2
θ_1
Bending strip
(parametric domain)
Bending strip stiffness:
$\begin{bmatrix} \bar{S}^{11} \\ \bar{S}^{22} \\ \bar{S}^{12} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{\varepsilon}_{11} \\ \bar{\varepsilon}_{22} \\ 2\bar{\varepsilon}_{12} \end{bmatrix} + \theta^3 \begin{bmatrix} E_s & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{\kappa}_{11} \\ \bar{\kappa}_{22} \\ 2\bar{\kappa}_{12} \end{bmatrix}$
Zero membrane
stiffness
Directional bending
stiffness

Figure 7.7: Schematic of the bending strip method.

method consists of augmenting the principle of virtual work with additional terms of the form:

$$\delta W^s_{int} = -\int_A \frac{t^3}{12} \, \overline{\boldsymbol{\kappa}}^T \overline{\mathbf{C}}_s \delta \overline{\boldsymbol{\kappa}} \, \mathrm{d}A \tag{7.4}$$

for every bending strip. The bending strip constitutive material matrix $\overline{\mathbf{C}}_s$ is given by:

$$\overline{\mathbf{C}}_s = \begin{bmatrix} E_s & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.5}$$

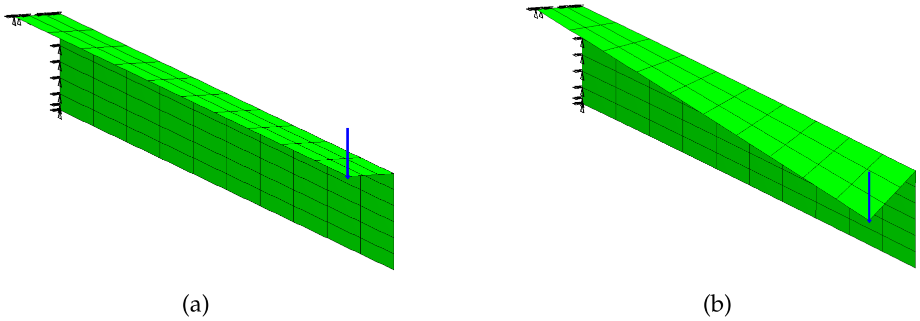
where E_s is the directional bending stiffness. The crucial point is the correct choice for E_s. It must be high enough that the change in angle is within an acceptable tolerance. However, if E_s is chosen too high, the global stiffness matrix may become badly conditioned and the solution unstable.

7.2.1 Choosing a Reliable Bending Strip Stiffness

In the following, the effect of different choices for E_s is investigated on different examples in order to obtain a range of values, which can be used in the general case. The bending strip stiffness E_s is chosen as a decimal power of the Young's modulus of the structure E in a range from $E_s/E = 10^0$ to $E_s/E = 10^{10}$.

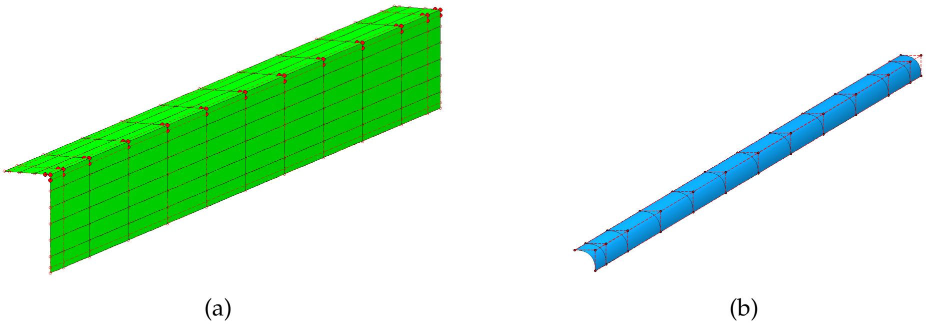
7.2.1.1 L-beam

The first example is an L-shaped beam consisting of two NURBS patches. It is clamped on the one end and subjected to a point load at the other end, see Figure 7.8(a). In Figure 7.8(b), it is demonstrated what happens if the patches are connected by only C^0 continuity without a bending strip. As can be seen, the structural behavior is the one of a hinge connection where bending moments are not transferred between the patches. A bending strip is added for the correct transfer of bending moments between the patches. In Figure 7.9(a), the control points to be coupled by the bending strip are highlighted, while Figure 7.9(b) shows the bending strip built from these control points. The computation is performed for different bending strip stiffnesses as described above, and the change in the angle between the patches is recorded at the tip. The results are collected in Figure 7.10. For stiffness ratios $E_s/E = 10^3$ and higher, the relative error is less than 10^{-3}. Further computations reveal that the solution becomes unstable not before a stiffness ratio $E_s/E > 10^{13}$, so there is a very large range for E_s in which the bending strip method yields correct and stable solutions. In Figure 7.11 the resulting deformation of the L-shape can-



(a) (b)

Figure 7.8: L-shaped cantilever consisting of two patches. (a) Problem setup. (b) The patches are connected by C^0 continuity only. The connection acts as a hinge.



(a) (b)

Figure 7.9: L-shaped cantilever. (a) Control points to be coupled by a bending strip are highlighted. (b) Bending strip built by these control points.

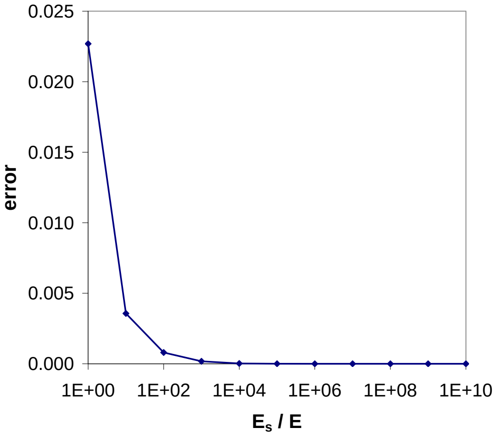


Figure 7.10: L-shape cantilever. Relative change in the angle between two patches at the cantilever tip as a function of the bending strip stiffness.

tilever using a bending strip is shown. As can be seen, the angle between the patches is maintained in the deformed configuration.

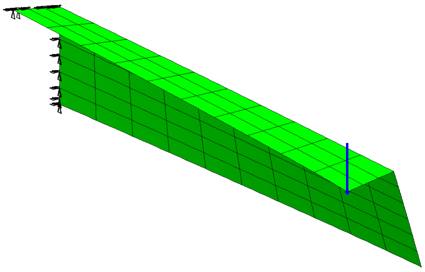


Figure 7.11: L-shaped cantilever with a point load. Deformed configuration using the bending strip method.

7.2.1.2 Cantilever Plate

In the second example, the bending strip is applied to a multipatch structure with C^1 continuity at the interface. For this, the same cantilever plate as in Section 7.1.1 is used, i.e. the plate is divided into two patches at half length and each patch consists of one cubic-linear element. It is obvious that this structure without continuity constraints at the interface would be kinematic, and therefore the overall structural response greatly depends on the behavior of the bending strip. The displacement at the tip of the cantilever is computed for different values of E_s, and is compared to the reference solution from linear beam theory. The results are collected in Figure 7.13. For all ratios $E_s/E \geq 10^3$ the relative error is less than 10^{-3}.

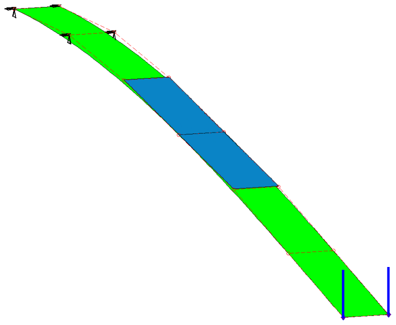


Figure 7.12: Two-patch model of a cantilever plate connected by a bending strip, deformed configuration.

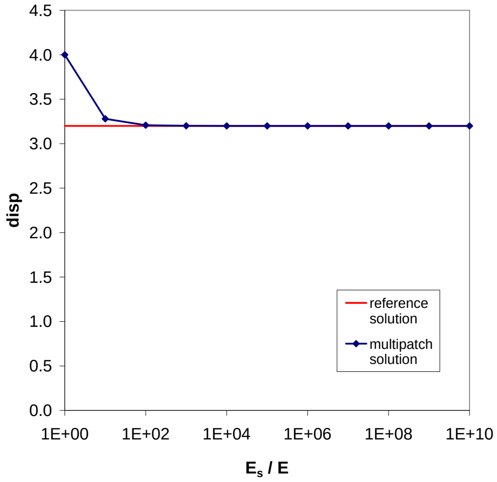


Figure 7.13: Cantilever plate tip displacement as a function of the bending strip stiffness.

This simple example allows to point out an important feature of the bending strips. The discretization of the plate has been chosen very coarse (one cubic element for each plate) which is enough since the reference solution is a cubic function. As a consequence, the bending strip is quite large, overlapping big parts of the plate, as can be seen in Figure 7.12. However, as shown in the computations, the result is correct and the bending strip does not influence the overlapped regions. This confirms what has been stated above, that the bending strip is not to be considered as an additional structural part but only as a constraint on the interface, independent of its extensions.

7.2.1.3 Hemispherical Shell

In the examples above, the relative error is less than 10^{-3} for a stiffness ratio $E_s/E = 10^3$ and continuously decreasing for higher ratios. Problems of numerical instability due to a bad conditioning of the stiffness matrix did not appear before a ratio of $E_s/E = 10^{13}$. For

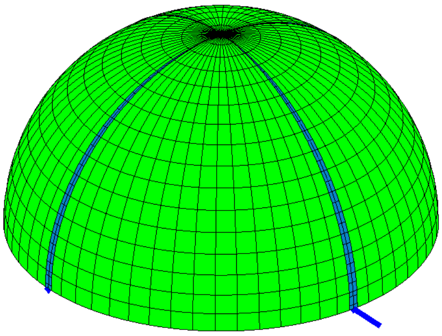


Figure 7.14: Hemispherical shell. The complete hemisphere is modeled by four patches with bending strips.

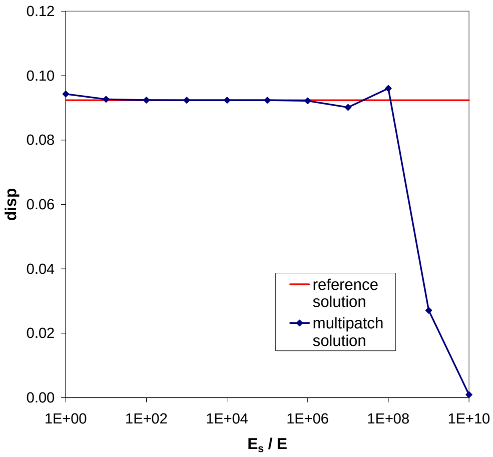


Figure 7.15: Pinched hemisphere. Displacement under the point load as a function of the bending strip stiffness.

determining an upper limit for the range of possible values, a more sensitive problem is chosen, where high differences in the stiffness matrix appear even before adding the bending strips. The hemispherical shell from the shell obstacle course is a good example for this test, since there are huge differences in element sizes on the rim and on the pole of the hemisphere. The hemisphere is modeled by four patches connected by bending strips, see Figure 7.14. Each patch is discretized by 16×16 quartic elements which has yielded the reference solution for the single patch case (see Chapter 6). Again, the displacement under the loaded points is observed and compared to the reference solution for different stiffness ratios E_s/E. The results are collected in Figure 7.15. For $10^2 \leq E_s/E \leq 10^5$ the reference solution is obtained. But for ratios $E_s/E \geq 10^6$ the stiffness matrix is badly conditioned and the solution becomes unstable. So even in this example with extreme differences in element sizes there is still quite a large range of bending strip stiffness values which yield reliable results.

Further tests on various geometries, which are not presented here, have been performed in order to determine the range of reliable stiffness values. Summarizing, all stiffness values in the range of $10^3 \leq E_s/E \leq 10^5$ gave reliable and stable results and as a consequence, this range is proposed as a rule of thumb for choosing the bending strip stiffness.

7.2.2 Automated Coupling of Multiple Patches with Bending Strips

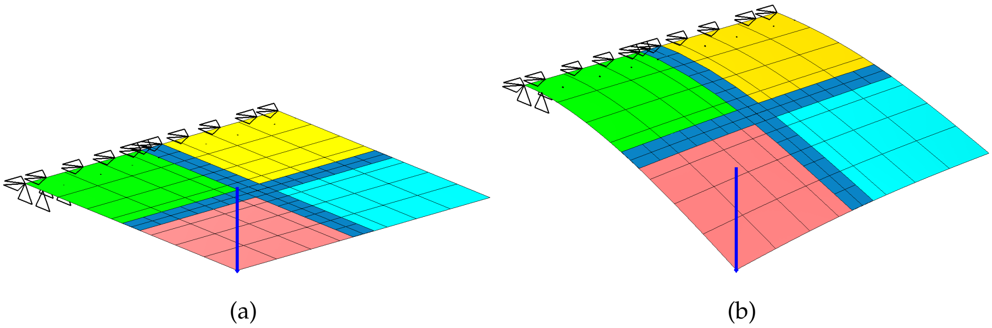


Figure 7.16: Four-patch cantilever plate with bending strips. Bending strips overlap each other at the middle point. (a) Problem setup. (b) Resulting deformation.

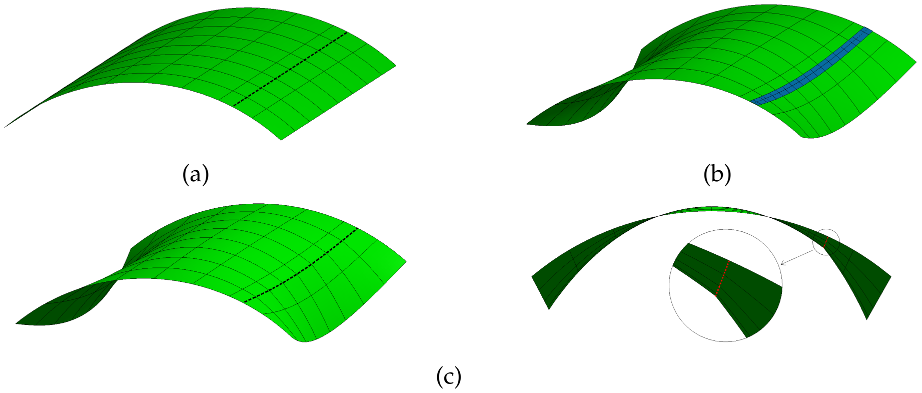
In Section 7.1.3 it has been discussed that for enforcing continuity constraints by direct coupling of displacements, care has to be taken at points where four patches meet (Figures 7.4, 7.6). This is because applying the corresponding constraint equation to all interfaces by an automated procedure produces one superfluous constraint which yields to a wrong result. This is not the case if bending strips are used. In Figure 7.16, the four-patch cantilever plate is shown again, where the patches are connected by bending strips. As can be seen, the bending strips overlap each other at the middle point where the four patches meet. In the overlap region, the bending strip stiffnesses are added. Since, theoretically, a bending strip stiffness $E_s \to \infty$ yields the correct solution, the strips can be added and can overlap arbitrarily without problem. The resulting deformation, shown in Figure 7.16(b), is identical to the correct solution presented in Section 7.1.3. This feature makes the bending strip method very practical for large structures consisting of many patches, because an automatic coupling of patches can be employed.

7.2.3 Numerical Benchmarks using Bending Strips

In Section 7.2.1, different examples have been used to find a reliable choice for the bending strip stiffness. In the following, this value is kept fixed at $E_s/E = 10^3$ and the bending strips are applied to the numerical benchmarks presented in Chapter 6 for a complete and thorough testing.

7.2.3.1 Shell Obstacle Course

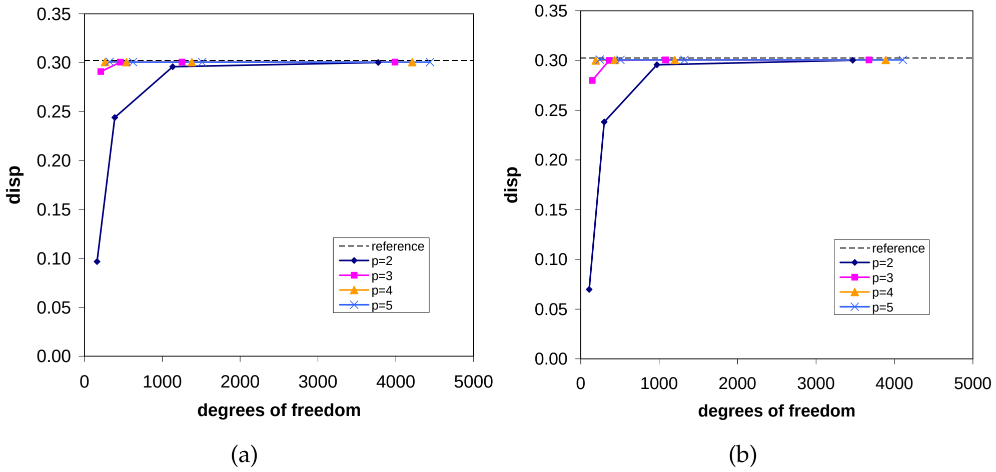
The examples of the shell obstacle course, presented in Section 6.2, are performed modeling the structures as multipatches with bending strips. For investigating their efficiency, convergence studies are performed for polynomial degrees $p = 2, 3, 4, 5$ and are compared to the results from the single patch solutions of Section 6.2. The hemispherical shell problem is omitted here since it has been investigated already in Section 6.2.


(a)

(b)

(c)

Figure 7.17: Two-patch model of the Scordelis-Lo roof. (a) Reference configuration. The patch interface is marked by a dashed line. (b) Deformed configuration using the bending strip method. (c) Deformed configuration without a bending strip. A kink in the deformed geometry is clearly visible.

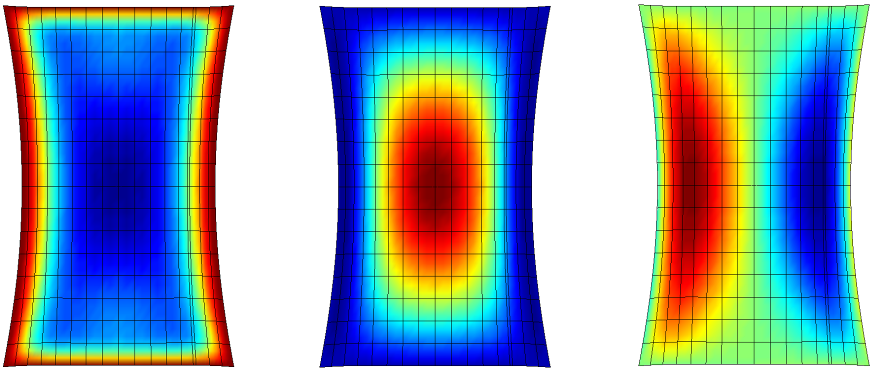


(a)

(b)

Figure 7.18: Two-patch model of the Scordelis-Lo roof. Convergence study. (a) Two-patch computation with a bending strip. (b) Single patch computation.

Scordelis-Lo roof: The Scordelis-Lo roof is divided into two patches by repeated knot insertion. Figure 7.17(a) depicts the roof consisting of two patches with the interface



(a) $\bar{n}^{11}$ (b) $\bar{m}^{11}$ (c) $\bar{q}^{1}$

Figure 7.19: Two-patch model of the Scordelis-Lo roof. Stress resultants.

	$\widehat{n}^{11}$	$\widehat{m}^{11}$	$\widehat{q}^{1}$
min.	-3510	-91	-280
max.	27	2053	280

Table 7.1: Two-patch model of the Scordelis-Lo roof. Maximum stress resultants.

marked by a dashed line. Figure 7.17(c) shows the resulting displacement if no bending strip is applied. As can be seen, a kink is developing along the patch interface, and the vertical displacement at the midpoint of the side edge $u = 0.3871$ is significantly higher than the reference value of $u_{ref} = 0.3024$. A bending strip is applied along the patch interface and the resulting deformation is shown in Figure 7.17(b), which corresponds to the correct solution from the single patch computation. The convergence results of the bending strip solution are plotted in Figure 7.18(a). For comparison, the complete roof is modeled by a single patch (i.e., no C^0 lines), and the convergence results are collected in Figure 7.18(b). It can be seen that the convergence behavior is nearly identical in both cases. Figure 7.19 shows the distribution of the stress resultants from the bending strip solution and in Table 7.1, the maximum values are collected. As can be seen, the stresses in the shell are smooth across the patch interface, and they are nearly identical to those of the single patch solution, compare to Figure 6.11 and Table 6.1. It is important to note that the stresses in the region where the bending strip overlaps the structure, are not carried by the bending strip but the structure itself. This affirms once more that the bending strips are not to be considered as structural patches but solely as an angle constraint between the patches.

Pinched cylinder: In this example, half of the pinched cylinder is modeled by four patches, connected by overlapping bending strips. Figure 7.20 shows the problem setup, in Figure 7.21 the convergence study is plotted and compared to the results from a single patch solution. As in the previous example, the convergence behavior of the bending

strip solution is nearly identical to the single patch solution.
Remark: Since the half cylinder cannot be modeled by one patch without C^0 continuity, the single patch computation is performed on one eighth of the cylinder, as in Section 6.2. As a consequence, the single patch computations use one quarter of degrees of freedom for corresponding results, see Figure 7.21.

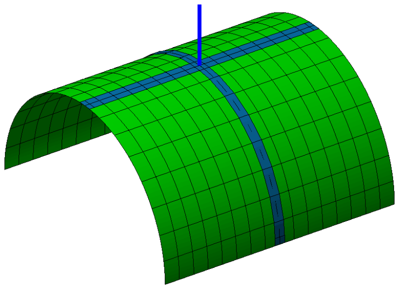
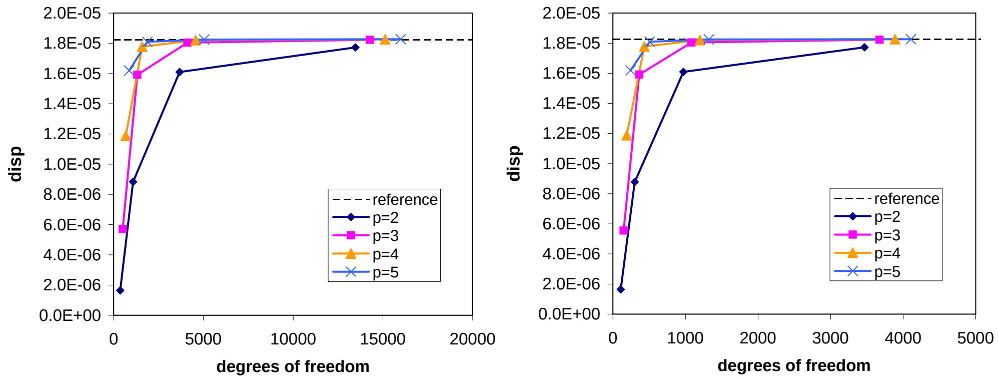


Figure 7.20: Pinched cylinder. Half of the cylinder is modeled by four patches connected by bending strips.

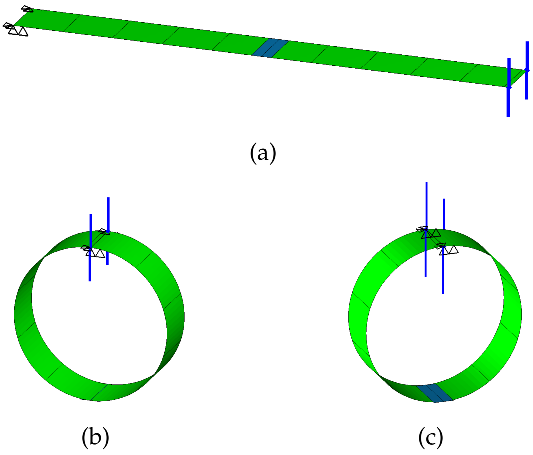


(a) (b)

Figure 7.21: Pinched cylinder. Convergence study. (a) Multi-patch computation with bending strips. (b) Single patch computation with symmetry boundary conditions. Convergence behavior is nearly identical in both cases.

7.2.3.2 Bending Strips for Large Deformations

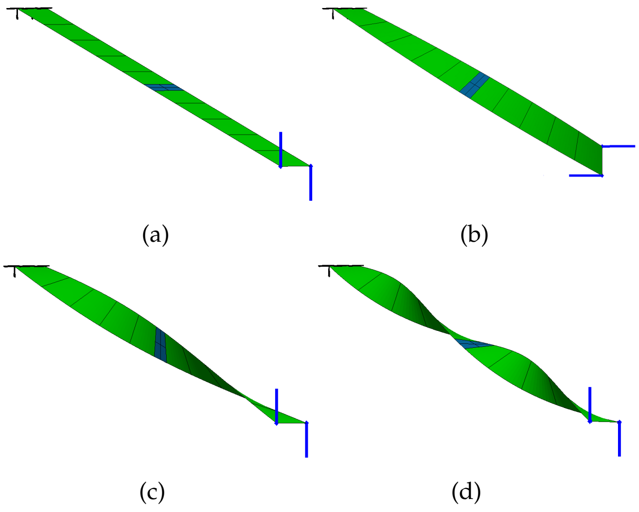
Applying the bending strips to the examples of the shell obstacle course has proven their validity for geometrically linear problems, i.e. small deformations. In this sections, the benchmark tests for large deformations, which have been used in Section 6.3, are revisited using multipatch geometry models and bending strips. For these tests, a cantilever plate is split into two patches which are joined by a bending strip. For the geometrically nonlinear tests, the plate is subjected to a pure bending moment and a pure twisting moment, as described in Section 6.3. The results are depicted in Figure 7.22 for the bending case and in Figure 7.23 for the twisting case. As can be seen, the correct results are obtained and the applicability of the bending strips for large deformations and rotations is demonstrated.


(a)

(b)

(c)

Figure 7.22: Bent plate with bending strip. (a) Initial configuration. (b) Deformed configuration. (c) Deformed configuration with a bending strip shown on the bottom.


(a)

(b)

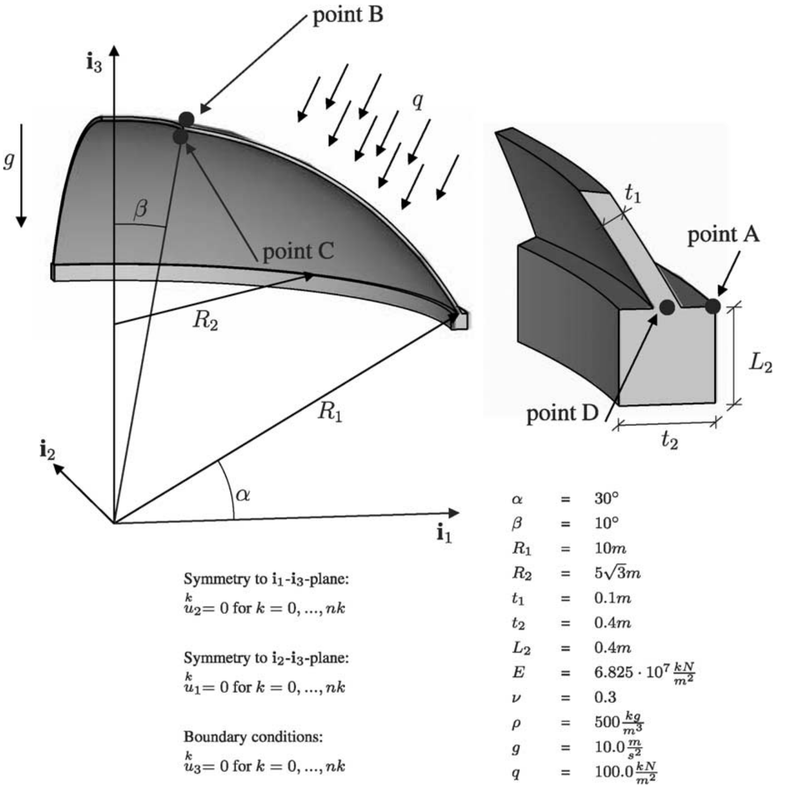
(c)

(d)

Figure 7.23: Twisted plate with bending strip. (a) Initial configuration. (b) Twisted by 90°. (c) Twisted by 180°. (d) Twisted by 360°.

7.2.4 Bending Strips for Coupling of Shells and Solids

Structures of practical interest are often assemblies of different parts, of which some can be classified as shells and others as solids. If both shell and solid elements shall be employed in one structural model, the difficulty is how to treat the interface between shell and solid elements. This is a general problem of shell analysis since most shell elements use rotational degrees of freedom whereas solids usually have only translational degrees of freedom. Thus, there is a misfit of discretization at the interface i.e. the shell's degrees of freedom cannot be transferred correctly to the solid. Using a rotation-free shell formulation, the shell has the same degrees of freedom as the solid. For transferring the rotations between shell and solid, the bending strips can be applied like in the case of coupling two shells. A further problem, when coupling shells and solids, is the fact that shells are represented as surfaces. It means that a shell geometry is a structure of zero thickness, which at the interface introduces a singularity in the solid. For correctly modeling the connection, the bending strips cover the area of the intersection shell-solid. For covering the whole intersection area, bending strips with more than one quadratic element in the transverse direction may be necessary. As a result, the strips maintain the correct width under mesh refinement.



Symmetry to $\mathbf{i}_1$-$\mathbf{i}_3$-plane:
$\overset{k}{u}_2 = 0$ for $k = 0, ..., nk$

Symmetry to $\mathbf{i}_2$-$\mathbf{i}_3$-plane:
$\overset{k}{u}_1 = 0$ for $k = 0, ..., nk$

Boundary conditions:
$\overset{k}{u}_3 = 0$ for $k = 0, ..., nk$

α	=	$30°$
β	=	$10°$
R_1	=	$10m$
R_2	=	$5\sqrt{3}m$
t_1	=	$0.1m$
t_2	=	$0.4m$
L_2	=	$0.4m$
E	=	$6.825 \cdot 10^7 \frac{kN}{m^2}$
ν	=	0.3
ρ	=	$500 \frac{kg}{m^3}$
g	=	$10.0 \frac{m}{s^2}$
q	=	$100.0 \frac{kN}{m^2}$

Figure 7.24: Hemispherical shell with a stiffener. Problem setup from [RDN⁺04]

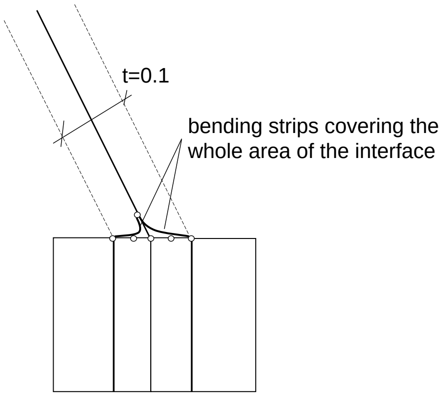
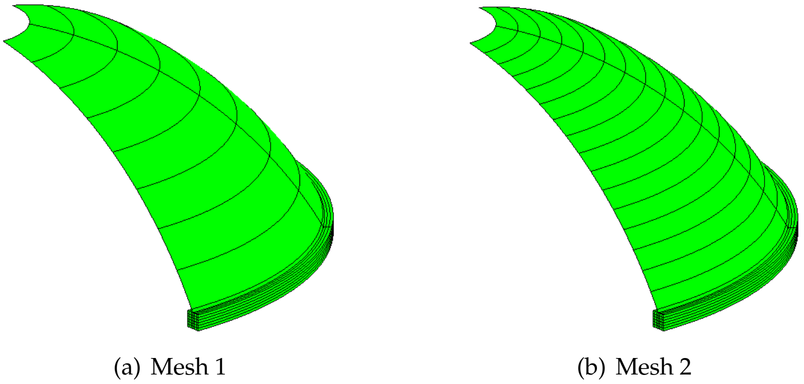


Figure 7.25: Hemispherical shell with a stiffener. Solid-shell coupling using bending strips.


(a) Mesh 1

(b) Mesh 2

Figure 7.26: Hemispherical shell with a stiffener. Meshes employed for a $p-$refinement study.

The "hemispherical shell with a stiffener" problem presented by Rank et al. [RDN$^+$04] is used to demonstrate the method. The problem setup is depicted in Figure 7.24. Rank et al. have used two different methods to solve this problem: a) a hierarchical shell approach where both shell and stiffener are modeled with high-order shell elements and b) a 3D solid formulation for both shell and stiffener based on $p-$version hexahedral finite elements [SDR04]. Here, a hybrid approach is used, where the stiffener is modeled as a 3D solid and the hemisphere as a shell. For the solid, a standard solid formulation using NURBS as basis functions, as presented in [HCB05], is used. The shell is modeled by the NURBS-based Kirchhoff-Love shell. As a consequence, shell and solid elements have the same discretization at the interface. Four solid patches are used to model the stiffener, and bending strips are defined to extend over the patches that cover the interface. By this, it is assured that the bending strips retain the correct width under mesh refinement. Figure 7.25 illustrates the shell-solid intersection and the bending strips. The dashed lines show the real 3D dimensions of the shell while its midsurface is depicted by a solid line. The bending strips, placed on both sides of the shell, cover the entire overlap region.
A $p-$refinement study was performed using two different NURBS meshes. The results

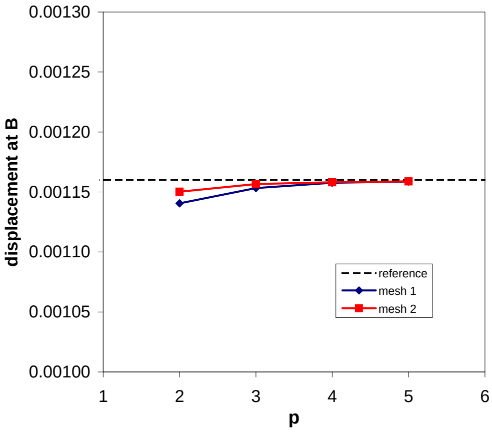


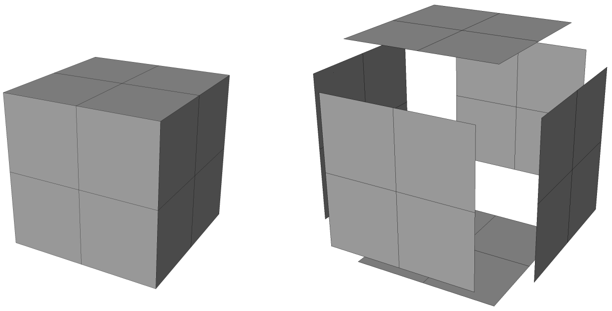
Figure 7.27: Hemispherical shell with a stiffener. Displacement convergence at point B.

for the displacement at point B, located at the top of the shell (see Figure 7.24), are compared to the value given in [RDN$^+$04]. Figure 7.26 shows the meshes. In both cases, two cubic elements are used in the circumferential direction. The stiffener cross section is discretized using 6×6 rectangular elements. Mesh 1 and Mesh 2 use 8 and 16 elements, respectively, in the meridian direction (see Figure 7.26). In the $p-$refinement study, the polynomial degree is raised from quadratic to quintic for both the shell and stiffener. No refinement is necessary in the circumferential direction due to axisymmetry. The displacement at point B is plotted in Figure 7.27. Rapid convergence to the reference value is observed for both discretizations.

Chapter 8

Integration of Design and Analysis

In this chapter, the isogeometric Kirchhoff-Love shell is applied to the integration of analysis into CAD. The suitability of the presented shell formulation for this integration is demonstrated and different aspects of geometric modeling and their effects on isogeometric analysis are discussed.



(a) (b)

Figure 8.1: Modeling of a cube in a CAD program. It appears as a volumetric object (a), but the underlying description is defined as a set of boundary surfaces (b).

8.1 Integrating Isogeometric Shell Analysis into CAD

A general problem of using CAD geometries for analysis is that most CAD programs use the boundary representation (B-Rep) method for volumetric objects. This means that a volumetric object, e.g. a cube is not stored as a trivariate NURBS object, but as a set of surfaces, which are its boundary. This is illustrated in Figure 8.1. Obviously, this B-Rep representation cannot be used as geometry model for analysis. It is necessary to create a trivariate (volumetric) NURBS model from the boundary surfaces.
Thin-walled structures like, for example, car bodies or ship hulls are usually designed as surfaces with a certain thickness, rather than volumetric objects. This corresponds to the structural model in shell analysis. Of course, such structures could also be analyzed with

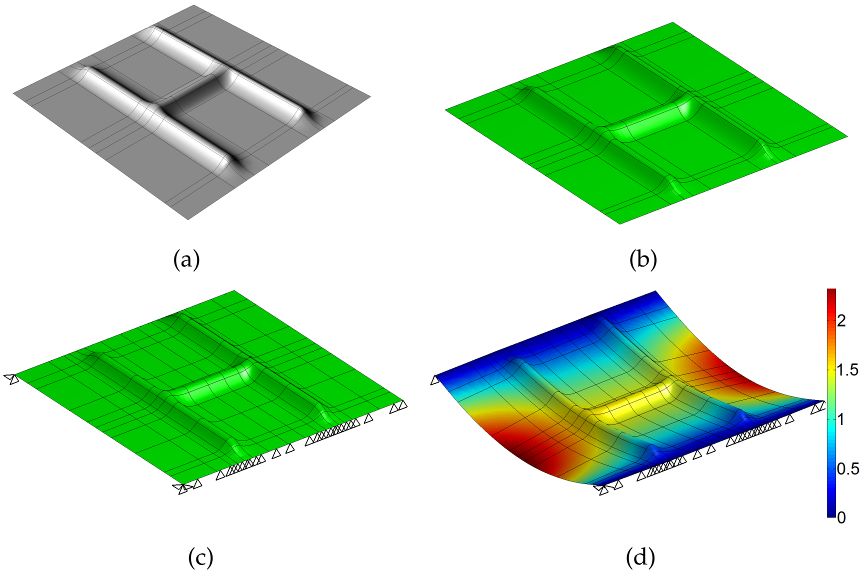

(a)
(b)
(c)
(d)

Figure 8.2: Isogeometric shell analysis. (a) CAD model. (b) The same model in the analysis code. (c) The analysis model is refined, boundary conditions and material properties are applied. (d) Displacement plot as result of the isogeometric analysis under self-weight.

solid elements, but then again, a volumetric geometry has to be created from the surface model first. For isogeometric shell analysis, the NURBS surface designed in a CAD program can be used as the shell's geometry. If a shell formulation using rotational degrees of freedom or nodal directors is used [BWBR04], these directors need to be assigned additionally in the analysis model, since they do not exist in the design model. Using a rotation-free shell formulation allows the use of exactly the same geometry description for design and analysis. This is demonstrated in Figure 8.2. It shows the single steps from the CAD geometry to the result of the isogeometric analysis. Figure (a) shows the geometry of a plate with beads in the CAD model. This geometry is exported without modification to the isogeometric analysis code (b). Here, refinement is performed and boundary conditions as well as material properties are applied (c). The plate is simply supported on two sides and subjected to self-weight. Figure (d) shows the color plot of the resulting deformation.

The geometry shown in Figure 8.2(a) has been created in *Rhino* [rhi], which is a powerful, NURBS-based CAD program used for ship design, industrial design, etc. For the next examples, a user-implemented plug-in [Sch09, Bre10] for *Rhino* has been employed. With this plug-in, a complete integration of isogeometric analysis into *Rhino* is realized, as illustrated in Figure 8.3. Figure (a) shows an oil barrel, modeled in *Rhino*. In (b), the same geometry is displayed with shaded environment to emphasize the CAD capabilities of this program. In (c), boundary conditions are applied (fixed support at the bottom and internal pressure) and material parameters are assigned. The boundary conditions are defined by surface parameters, so for analysis, different refinements can be performed without resetting the boundary conditions. As result of the analysis, the normal forces in circumferential direction are plotted onto the deformed geometry in (d).

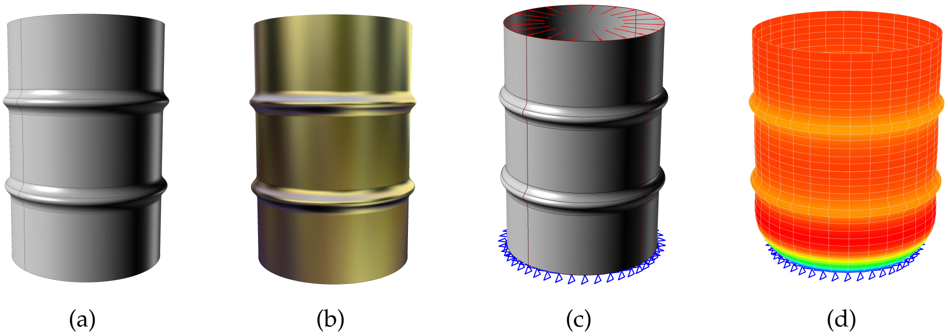


Figure 8.3: Integration of isogeometric analysis into a commercial CAD program. (a) Design model of an oil barrel. (b) Barrel with shaded environment mapping. (c) Boundary conditions and material properties are assigned in the CAD program through a user-implemented plug-in. (d) Isogeometric analysis is performed on a refined model. The normal forces in circumferential direction are plotted on the deformed geometry.

It is just important that the necessary continuity conditions for analysis are ensured. Especially for rotational surfaces, this is a point which might not be obvious immediately. Rotational surfaces such as a cylinder or the barrel in Figure 8.3 appear as closed, smooth surfaces. In the NURBS representation, however, there is first of all a location where the start and end of the NURBS patch meet, which is a location of C^{-1} continuity. Furthermore, there are three locations of C^0 continuity due to multiple knots in the knot vector. Additional constraints are needed to enforce the necessary continuity conditions for analysis. C^0 continuity is necessary to guarantee a closed surface, independent of the element type used in analysis, and is obtained by coupling the control points of the patch at its meeting ends. For the presented Kirchhoff-Love shell formulation, at least G^1 continuity is necessary everywhere on the patch. This means that the C^0 locations need to be identified and corrected by the methods presented in Chapter 7.

8.2 Analysis-Aware Modeling

In the previous section, the possibility of integrating isogeometric analysis into CAD with the developed shell formulation was shown. In this chapter, different aspects of geometric modeling and their effects on isogeometric analysis are investigated. Some of these aspects refer to this specific shell analysis, but most of them are generally valid for NURBS-based isogeometric analysis.

8.2.1 Alternative Parametrizations

There are typically several different parametrizations possible to describe the same geometry. While for a design model it might be of less importance how the geometry is parametrized, it has important consequences for the analysis model. This is exemplified on a circular disc [CMK+10, SKBW10]. Two possible parametrizations are considered. The first one is obtained by rotating a straight line around one of its endpoints, as illustrated in Figure 8.4. This can be considered as the easiest and most intuitive way to model a circular disc. However, it yields a rather bad parametrization. As can be seen, the disc consists of 18 control points, where nine of them coincide at the center of rotation. Obviously, this yields a singularity at the disc's center. Figure 8.5 shows this parametrization after some refinement. All the elements at the center are topologically quadrilaterals

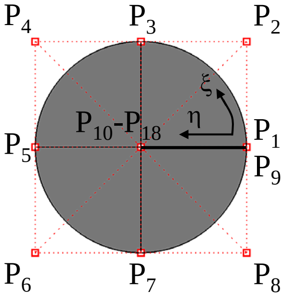


Figure 8.4: Circular disc, modeled by rotating a straight line.

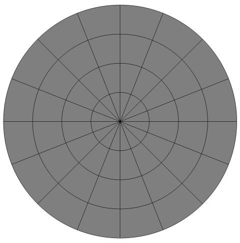


Figure 8.5: Refined model of the disc shown in Figure 8.4.

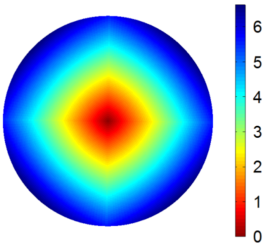


Figure 8.6: Color plot of the Jacobian for the disc model in Figure 8.4.

degenerated to triangles, and with increasing refinement they become very small. For investigating the parametrization quality, the Jacobian is plotted onto the disc in Figure 8.6. As can be seen, the Jacobian becomes zero at the center due to the singularity at this point. Furthermore, there are four locations with only C^0 continuity due to the rotational definition. For the presented shell formulation, at least G^1 continuity is necessary, which means that continuity constraints as presented in Chapter 7 have to be applied in order to use this geometric model for analysis.

A second alternative to model this circular disc is degenerating a square. This is illustrated in Figure 8.7. Starting from a biquadratic square with the edge length a, the midpoints of the edges (P_2, P_4, P_6, P_8) are pulled outwards by $a/2$ and their weights are adjusted to $\sqrt{0.5}$. In this case, only nine control points are needed and the parametrization is better than in the first model. Figure 8.8 shows this model after some refinement. As can be seen, the elements are less distorted than in the first model (Figure 8.5). In Figure 8.9 it can be seen that the variations in the Jacobian are smaller than in the first model (Figure 8.6). Furthermore, there is C^1 continuity or higher everywhere, and no additional continuity constraints are necessary. However, singularities appear also in this model. They are at the points P_1, P_3, P_7 and P_9, i.e. the corners of the former square. As can be seen in Figure 8.7, at point P_1, the two parametric directions ξ and η become collinear. The same problem occurs at the points P_3, P_7 and P_9. These singularities can

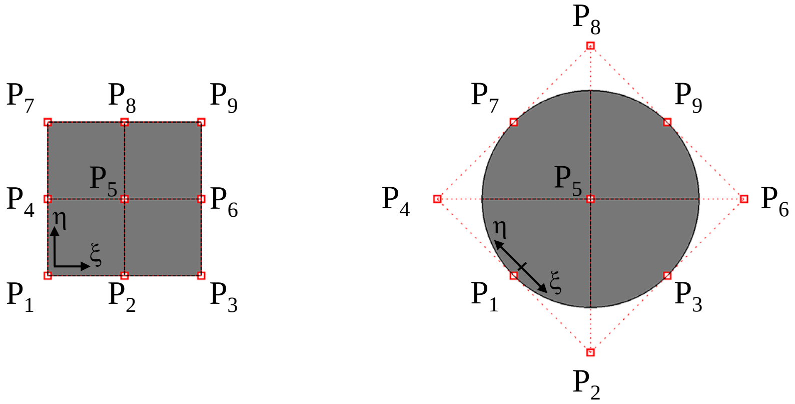


Figure 8.7: Circular disc, modeled by degenerating a square.

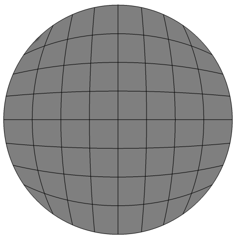


Figure 8.8: Refined model of the disc shown in Figure 8.7.

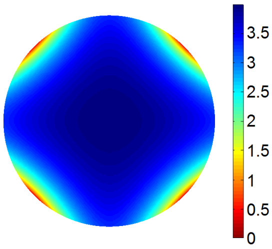


Figure 8.9: Color plot of the Jacobian for the disc model in Figure 8.7.

also be seen in the plot of the Jacobian in Figure 8.9. So, in both models geometric singularities occur, however they do not pose severe problems for analysis since they do not lie on integration points.

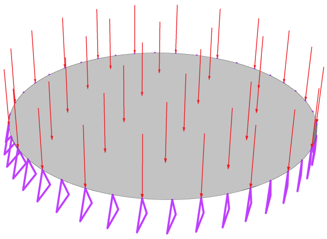


Figure 8.10: Simply supported circular plate under self-weight.

For evaluating the effect of the two different parametrizations on isogeometric analysis, a static analysis is performed for both cases. The disc acts as a plate, it is simply supported on its boundary and subjected to self-weight, see Figure 8.10. The displacement at the center is observed and compared to the analytical solution $w_{anal} = \frac{5}{64}\frac{pr^4}{K}$ where p is the load, r the radius and $K = \frac{Et^3}{12}$ the plate stiffness [Ble08]. For both parametrizations, biquadratic NURBS are used, knot refinement is applied and the results are compared to the analytical solution. The results are gathered in the convergence plot in Figure 8.11. It can be seen that Parametrization 2 converges faster, as it is to be expected due to the better parametrization. But it is also important to note that model 1, although highly distorted, converges to the correct result as well, with an acceptable convergence rate. This shows that NURBS-based elements are rather insensitive to mesh distortions [LEB+10].

8.2.2 Trimmed Surfaces

Trimming is a very important method in CAD modeling and a standard feature in CAD programs. It allows the modification of NURBS surfaces by "cutting off" parts along an arbitrary trimming curve. The Boolean operations unite, divide and intersection, which are essential operations for creating complex objects from geometric primitives, also make use of the trimming technique. The intersection curve between two objects is

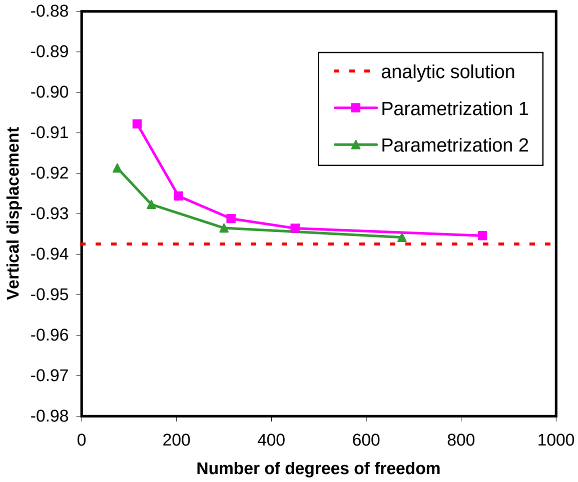


Figure 8.11: Simply supported circular plate under self-weight. Convergence plot of the center displacement.

then the trimming curve. However, trimming does not really divide the patch into two new NURBS patches of which one is then deleted. It rather maps the trimming curve onto the surface which divides the patch into a valid and an invalid part. The invalid part is made invisible in the program. However, the underlying NURBS description still contains the complete patch. This is illustrated in Figure 8.12. Figure (a) shows a square

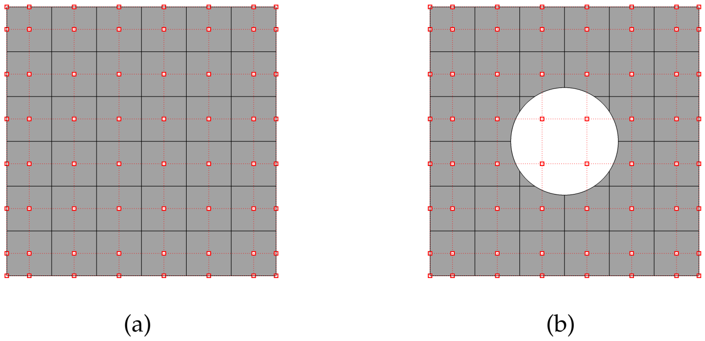


Figure 8.12: Trimming. (a) A square plate with the underlying control net. (b) A circular hole is cut into the plate by trimming. The underlying control net is still the same.

plate and its underlying control net. A circular hole is cut into this plate by trimming (b). In the control net of the trimmed surface, it can be seen that the underlying NURBS description is still the same. The "cut-off" part is only made invisible in the CAD program. It is obvious that this is not a valid representation for a NURBS-based analysis. The idea of disregarding the invalid elements during assembly for the analysis fails since the trimming curve can be arbitrary and is not confined to the existing knots which in fact define the element boundaries. An alternative representation is depicted in Figure 8.13. It is the closed patch between a circle as the inner boundary and a square as the outer boundary. In this representation, there are no trimmed parts and the model is analysis-suitable. However, it is more complicated to construct than the trimmed model.

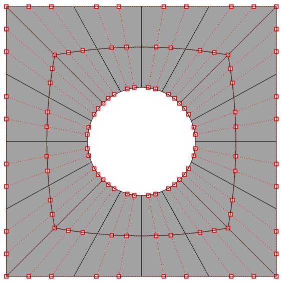


Figure 8.13: Analysis-suitable representation of the plate with hole.

The same problem can even appear without having performed a trimming operation. It is the case when modeling a simple triangle in a NURBS-based program. Due to their tensor product nature, NURBS always have a quadrilateral topology and a triangle with only three control points cannot be modeled by NURBS. Figure 8.14 shows what usually happens in a NURBS-based CAD program. A triangle is specified by three points (a). The underlying NURBS description can be seen by visualizing the control grid (b): instead of a triangle, a quadrilateral is constructed where half of the surface is trimmed. In (c), the analysis-suitable alternative is shown. The triangle is modeled as a degenerate quadrilateral, where the upper two control points coincide.

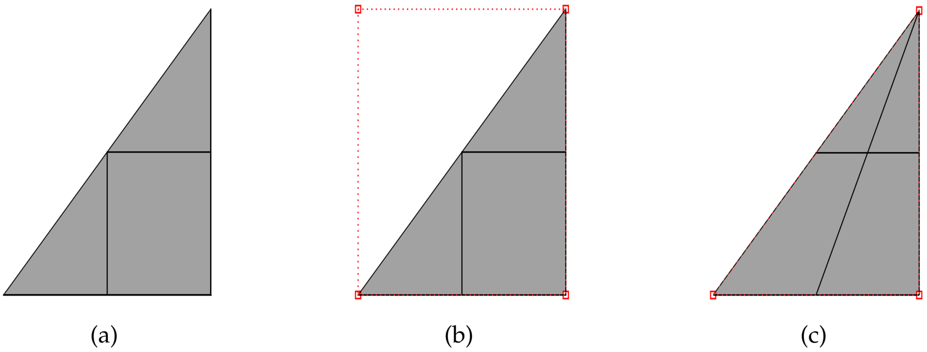


Figure 8.14: Modeling a triangle with NURBS. (a) A triangle is defined by three points. (b) Internally, a quadrilateral is created with half of it trimmed. (c) Analysis-suitable model: A square is degenerated to a triangle by coinciding two control points.

These examples show that if isogeometric analysis is to be performed on a CAD model, the designer has to be aware of the demands that the analysis has on the geometric model. If there are alternative parametrizations for the same shape, the one which is best suited for isogeometric analysis should be chosen. The biggest problem for NURBS-based isogeometric analysis is that trimmed surfaces, which are standard in CAD modeling, cannot be computed. It has been shown that trimmed surfaces can be avoided by using alternative, non-trimmed parametrizations. However, this is not a feasible solution for complex geometry models. Replacing all trimmed surfaces by untrimmed surfaces would result in a huge remodeling effort and the advantage of isogeometric analysis would get lost. An approach to perform analysis on trimmed surfaces is presented in [KSY09, KSY10] for plane 2D problems. It uses NURBS-enhanced finite elements (NEFEM) [SFMH08] for those elements which are divided by the trimming curve. A general problem of trimming is that the intersection of two NURBS surfaces cannot be represented exactly by a NURBS trimming curve [SFLL08]. This leads to small gaps between the patches which can be neglected for design but are crucial for analysis. This can be repaired by using T-Splines. T-Splines are a generalization of NURBS [SZBN03] that is not restricted to a tensor product structure. This means that T-junctions in the control grid are possible, which allows local refinement [SCF04]. Furthermore, it is possible to convert trimmed NURBS surfaces into untrimmed T-Spline surfaces [SFLL08] and then close the gaps between these patches. There are T-Spline plug-ins available for the NURBS-based CAD programs *Rhino* and *Maya*. Nevertheless, NURBS are still the predominant technology in CAD programs. Isogeometric analysis using T-Splines has been studied in [DJS10, BCC$^+$10, UY09, UKY08, BBDL$^+$10] and good results were obtained. However, there are still open questions about the linear independence of T-Splines, which is an indispensable prerequisite for analysis, as discussed in [BCS10] and [SLZ$^+$10].

Chapter 9

Isogeometric Shape Optimization

In this chapter, the isogeometric concept is extended to shape optimization. Shape optimization is a subcategory of structural optimization with the goal of finding the optimal shape of a structure with respect to a certain objective. The optimization process can be split into two main steps. The first step is formulating the optimization problem, the second step is choosing a proper optimization strategy. For setting up the optimization problem, the designer must define the objective which shall be optimized, and formulate it in the objective function. The next crucial step is defining the design model. The choice for a certain design model specifies the possible solutions of the optimization and therefore this step can be regarded as the most decisive step in the whole optimization process. In structural shape optimization, there are two different approaches concerning the design models, namely the CAD-based [Ble90, Ble93, BR01, BWDC05] and the FE-based [BFLW08, Dao05, DCB04] approach. In the CAD-based approach, the design model is specified by the CAD description of the structure, whereas in the FE-based approach, the finite element mesh is used as design model. These two approaches are reviewed, and isogeometric shape optimization is introduced as a possibility to merge the advantages of both approaches. The chapter starts with a short review of the fundamentals of structural optimization, and different solution strategies are presented. Due to the huge variety of optimization algorithms, only a few are presented. For an introduction to structural optimization, reference is made to [Aro89, BK85b].

9.1 Mathematical Formulation of a Structural Optimization Problem

A structural optimization problem can be defined mathematically in the following form:

$$
\begin{aligned}
&\text{Minimize} && f(\mathbf{s}) && \mathbf{s} \in \mathbb{R}^{n_s} \\
&\text{such that} && g_i(\mathbf{s}) \leq 0 && i = 1, \ldots, n_g \\
& && h_j(\mathbf{s}) = 0 && j = 1, \ldots, n_h \\
& && \mathbf{s}_l \leq \mathbf{s} \leq \mathbf{s}_u &&
\end{aligned}
\tag{9.1}
$$

where f is the objective function, $\mathbf{s}$ is the vector of the design variables, g_i is the set of inequality constraints and h_j are the equality constraints. $\mathbf{s}_l$ and $\mathbf{s}_u$ are the lower and upper bounds of the design variables. In fact, they are explicit inequality constraints on the design variables and therefore could be included in the set of inequality constraints, but in view of their simplicity they are treated separately.

9.1.1 Objective Function

The objective function describes the objective that has to be minimized by the optimization. Alternatively, optimization can also be formulated as maximization of the objective function, which in fact is equivalent to a minimization of its negative. It is important to be aware of the fact that complex and multidimensional objective functions usually have many local minima and it is not guaranteed that the global minimum is found by the optimization. Often, there are also more than one objectives. They can be transformed into one objective by the weighted sum of the single objective functions. Typical objectives for structural optimization are, for example, minimization of mass, minimization of compliance, maximization of eigenfrequency, maximization of buckling load, etc.

9.1.2 Design Variables

The design variables are the structural parameters which can be altered by the optimization in order to minimize the objective function. Depending on the type of design variables, structural optimization can be classified into four subcategories, namely sizing, material, topology and shape optimization. In the first two, the geometry of the structure remains unchanged. In sizing, only the sizes of the cross sections are to be optimized. This is applicable especially for truss structures. Material optimization aims at finding the best material parameters for non-isotropic materials, e.g. the fiber directions in composite laminated plates and shells. Topology optimization determines the best distribution of material in the design space. This means that no type of structure is specified a priori, but the whole design space is free to be filled with material. The result of topology optimization is usually not a practical and ready-to-build structure, but serves as a help to determine the appropriate type of structure and a coarse design for the given design space. In shape optimization, the type of structure is defined but its shape can be modified. For a geometric model defined by nodal points which are connected by certain basis functions, this means that the connectivity between the points remains fixed, but their position, i.e. their spacial coordinates, can be modified. If shape optimization is performed using finite element analysis, there are basically two types of design models, namely the CAD-geometry model and the finite element mesh. In both cases the design variables are nodal coordinates, for the first approach referring to control nodes in the second approach referring to FE-nodes. If the design model is described by NURBS, also the weights of the control points can be used as design variables. The different design models and corresponding design variables for shape optimization will be reviewed in more detail later in this chapter.

9.1.3 Constraints

The solution is usually restricted by some constraint functions which divide the design space into a feasible and an infeasible region. The constraint functions are classified into equality and inequality constraints. As mentioned above, the variable bounds $\mathbf{s}_l$ and $\mathbf{s}_u$ which describe the upper and lower bound of admissible values for the design variables, could also be regarded as inequality constraints, but due to their simplicity they are treated separately. For a certain design, constraints can be either fulfilled ($g_i \leq 0, h_j = 0$) or violated ($g_i > 0, h_j \neq 0$). Inequality constraints which are fulfilled can furthermore be classified into active and inactive. A constraint is active if it determines the optimum ($g_i = 0$), i.e. the optimum point lies on the border between feasible and infeasible design. An equality constraint can only be active or violated. If a constraint has no influence on the solution of the optimization problem, it is called redundant. Typical constraints in shape optimization are, for example, stress, mass, deformation, etc. Generally, all quantities which can be used as objective function could also serve as constraint function. Therefore, objective and constraint functions which describe responses of the structure to a variation in the design, are summarized in the term "response functions".

9.1.4 Lagrangian Function and Kuhn-Tucker conditions

For unconstrained optimization problems, the optimum is defined by the minimum of the objective function, i.e. the gradient of the objective function vanishes (assuming continuous first derivatives):

$$\nabla f(\mathbf{s}) = \mathbf{0} \tag{9.2}$$

For constrained problems, the auxiliary function L is introduced which includes the objective and the constraint functions:

$$L(\mathbf{s}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{s}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\mathbf{s}) + \boldsymbol{\mu}^T \boldsymbol{h}(\mathbf{s}) \tag{9.3}$$

This function is called the Lagrangian function and $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ are the Lagrangian multipliers. The optimum of the Lagrangian function is defined by the Kuhn-Tucker conditions:

$$\begin{aligned} \nabla_{\mathbf{s}} L(\mathbf{s}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \nabla_{\mathbf{s}} f(\mathbf{s}) + \boldsymbol{\lambda}^T \nabla_{\mathbf{s}} \boldsymbol{g}(\mathbf{s}) + \boldsymbol{\mu}^T \nabla_{\mathbf{s}} \boldsymbol{h}(\mathbf{s}) &= \mathbf{0} \\ \nabla_{\boldsymbol{\mu}} L(\mathbf{s}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \boldsymbol{h}(\mathbf{s}) &= \mathbf{0} \\ \lambda_i \nabla_{\lambda_i} L(\mathbf{s}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \lambda_i g_i(\mathbf{s}) &= 0 \\ \lambda_i &\geq 0 \end{aligned} \tag{9.4}$$

The Lagrangian function combines the objective function and the constraints into one function with additional variables $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. By this, the problem is transformed into the unconstrained problem of finding the stationary point of the Lagrangian function w.r.t. the primal variables $\mathbf{s}$ and the dual variables $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, and solution strategies from unconstrained optimization can be applied.

9.2 Optimization Algorithms

Optimization algorithms can be classified into different categories by the order of information they use. Therefore, the categories are zero, first and second order methods. For each of the categories, there exists a variety of algorithms which are not reviewed here, but only a general introduction is given.

Zero order methods evaluate only the function but not its derivatives. They are therefore the most simple but also the least efficient methods. Since they do not use any gradient information, they can be applied to very complex problems where evaluating the gradients is not possible or at least not effective, or to problems where the gradients cannot be evaluated due to discontinuous derivatives.

First order methods use the information of the gradient for determining the search direction. The simplest is the steepest descent method. It evaluates the gradient and uses its negative as search direction. A one-dimensional line search usually has to be performed to determine the optimal step size. The problem of steepest descent is that for badly conditioned problems it exhibits "zigzagging" and becomes inefficient. In these cases, conjugate gradient methods are better, which additionally use the gradient information of the last step to determine the search direction. Quasi-Newton methods, which work with an approximated inverse of the Hessian matrix, are classified as first order methods as well since they approximate the inverse Hessian by first order information. Very efficient algorithms like sequential quadratic programming (SQP) make use of Quasi-Newton methods.

Second order methods, also called Newton methods, make use of the second derivatives, the Hessian matrix, to obtain the search direction. Evaluating the Hessian matrix is usually very time-consuming. Therefore, Quasi-Newton methods are usually more efficient.

9.3 Sensitivity Analysis

Sensitivity analysis describes how much the objective changes due to a variation of a design variable [AH79]. Therefore, it is the gradient of the objective function w.r.t. the design variables. With this gradient, the search direction for the next step is determined. Sensitivity analysis can be performed using finite differences, analytically, or semi-analytically, which are explained in the following.

9.3.1 Global Finite Differences

The gradient can be obtained approximately by global finite differences, e.g. by forward differentiation:

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{s}} = \frac{f(\mathbf{s} + \Delta\mathbf{s}) - f(\mathbf{s})}{\Delta\mathbf{s}} \tag{9.5}$$

Obviously, this requires solving the whole equation system for each single variation Δs_i, which makes this method very inefficient for large systems. The advantage is that it is universally applicable, it does not need any information of the underlying analysis formulation. Thus, this method finds its application if, for example, a finite element code should be used as a black box inside the optimization algorithm.

9.3.2 Analytical Sensitivity Analysis

In analytical sensitivity analysis, the gradients are computed analytically. Obviously, this needs insight into the element formulation. For analytical sensitivity analysis, the objective function in the general optimization problem (9.1) is reformulated in the following way:

$$f = f(\mathbf{s}, \mathbf{u}(\mathbf{s})) \tag{9.6}$$

where $\mathbf{u}$ are called state variables, which in shape optimization are usually the displacements. The state variables, in turn, are functions of the design variables $\mathbf{u} = \mathbf{u}(\mathbf{s})$. This formulation accounts for the fact that most response functions are a function of the displacements and facilitates the formulation of the following derivations. The derivative of the objective function w.r.t. variations of the design variables can then be formulated using the chain rule:

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{s}} = \frac{\partial f}{\partial \mathbf{s}} + \left(\frac{\partial f}{\partial \mathbf{u}}\right)^T \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{s}} \tag{9.7}$$

The partial derivative of the objective w.r.t. to the design variable $\frac{\partial f}{\partial \mathbf{s}}$ is usually easy to compute and in many cases it is even zero. The most involved part is the derivative of the state variables w.r.t. the design variables $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{s}}$. For mechanically linear problems, these can be obtained by deriving the equilibrium condition w.r.t. $\mathbf{s}$:

$$\mathbf{K}\,\mathbf{u} = \mathbf{R} \tag{9.8}$$

$$\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}\mathbf{s}}\mathbf{u} + \mathbf{K}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{s}} = \frac{\mathrm{d}\mathbf{R}}{\mathrm{d}\mathbf{s}} \tag{9.9}$$

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{s}} = \mathbf{K}^{-1}\left(\frac{\mathrm{d}\mathbf{R}}{\mathrm{d}\mathbf{s}} - \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}\mathbf{s}}\mathbf{u}\right) \tag{9.10}$$

where $\mathbf{R}$ is the load vector. Due to the similarity of equation (9.10) with equation (9.8), the term in parenthesis is also called pseudo load vector $\mathbf{R}^*$:

$$\mathbf{R}^* = \frac{\mathrm{d}\mathbf{R}}{\mathrm{d}\mathbf{s}} - \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}\mathbf{s}}\mathbf{u} \tag{9.11}$$

Inserting (9.10) into equation (9.7) yields the equation for the analytical sensitivity analysis:

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{s}} = \frac{\partial f}{\partial \mathbf{s}} + \left(\frac{\partial f}{\partial \mathbf{u}}\right)^T \mathbf{K}^{-1} \left(\frac{\mathrm{d}\mathbf{R}}{\mathrm{d}\mathbf{s}} - \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}\mathbf{s}}\mathbf{u}\right) \tag{9.12}$$

$$= \frac{\partial f}{\partial \mathbf{s}} + \left(\frac{\partial f}{\partial \mathbf{u}}\right)^T \mathbf{K}^{-1}\, \mathbf{R}^* \tag{9.13}$$

As can be seen, the pseudo load vector requires the derivative of the stiffness matrix and the load vector w.r.t. the design variables. This is computationally very expensive, especially for complicated element formulations. And it means that access to the element formulation is necessary and the sensitivity analysis has to be reformulated for each new element type. By this, a general optimization environment which can be adopted for arbitrary element types is not possible.

9.3.3 Semi-Analytical Sensitivity Analysis

Semi-analytical sensitivity analysis [BFD08] is based on the analytical sensitivity as in equation (9.13). The difference is that the pseudo load vector is approximated by computing the sensitivities of the stiffness matrix and load vector by finite differences:

$$\mathbf{R}^* = \frac{\mathbf{R}(\mathbf{s}+\Delta\mathbf{s}) - \mathbf{R}(\mathbf{s})}{\Delta\mathbf{s}} - \frac{\mathbf{K}(\mathbf{s}+\Delta\mathbf{s}) - \mathbf{K}(\mathbf{s})}{\Delta\mathbf{s}}\mathbf{u} \tag{9.14}$$

By this, the derivation of the element formulation w.r.t. the design variables is circumvented, and an optimization algorithm using equation (9.14) can be applied to any new element type. Compared to global finite differences it is much more efficient because for every variation Δs_i only a modified stiffness matrix (and load vector) has to be computed but no additional equation system needs to be solved.

9.3.4 Direct vs. Adjoint Sensitivity Analysis

For analytical and semi-analytical sensitivity analysis, a distinction can be drawn between the direct and the adjoint method. The difference lies merely in the order in which matrix multiplications in equation (9.13) are executed. In the direct method, first the product $\mathbf{K}^{-1} \cdot \mathbf{R}^*$ is evaluated, whereas in the adjoint approach the product $\left(\frac{\partial f}{\partial \mathbf{u}}\right)^T \cdot \mathbf{K}^{-1}$ is evaluated first. This can have a significant effect on the numerical efficiency, depending on the number of design variables compared to the number of response functions. The adjoint method is more efficient if there are more design variables than response functions, which is usually the case in shape optimization problems. For more details, reference is made to [Ble90, BFD08].

9.4 Shape Parametrization

As stated above, shape optimization is a subcategory of structural optimization, where the design variables are geometrical parameters describing the shape of the structure. Generally, there are different possible representations to describe the same geometry and therefore different shape parameters can be chosen for optimization. For shape optimization using finite element analysis, there are even two different geometric models, namely the CAD model and the finite element model which is an approximation to the CAD model, created by meshing. Therefore, two different approaches for shape optimization have developed, the CAD-based approach on the one hand and the FE-based approach on the other hand.

9.4.1 CAD-based Shape Optimization

In the CAD-based approach, the CAD model represents the optimization model, which means that the parameters describing the CAD geometry, or a subset of them, are used as design variables for optimization. By this, the optimization works with a rather small number of design variables, which is numerically efficient and guarantees a certain regularity and smoothness in the solution due to the restricted solution space. Obviously, the latter is also a disadvantage. The optimization does not have much freedom, especially if the geometry is defined by very few parameters only. An extreme example of this is a sphere which is defined solely by its radius. For shape optimization on this model, it means that there is only one design variable and the solution can only be a sphere with a bigger or smaller radius, independent of the objective, the constraints, boundary conditions, etc. Then it is necessary to think of alternative parametrizations. This can be explained well with the example of the sphere. Instead of one radius R it could be defined by three radii R_1, R_2, R_3 which are equal at the beginning. With this parametrization, the solution would include elliptic shapes. A completely different parametrization could be a NURBS description of the sphere with the control point coordinates as design variables. This would allow for much more geometric flexibility in the solution. This example reveals two important aspects. Generally, it can be said that due to the relatively small number of shape variables, the variety of possible solutions is restricted and especially local effects can usually not be accounted for. It is furthermore obvious that the choice of parametrization strongly determines the solution. Although the latter generally holds for any kind of shape optimization (refer to the following subsections), it is much more pronounced in these cases where few variables define the whole shape. Therefore, this approach is, compared to the FE-based approach, applicable if the designer has already some predefined idea of what the solution should look like.

There is another drawback to this approach concerning the communication between different models. The optimization is performed on the CAD model, for evaluating the objective function and sensitivities, a finite element analysis has to be performed. This means that for each optimization step, actually for each design variation Δs_i, the finite element model has to be rebuilt.

9.4.2 FE-based Shape Optimization

In this approach, shape optimization is performed on the finite element model. This means that the transfer from the CAD to the FE model has to be done only once. The FE nodes, i.e. their coordinates, are the design variables, which provides a huge design space. This is advantageous in the sense that there is a big solution space and the solution is less predefined as in the CAD approach. At the same time, it is a problem because very irregular shapes and shapes that are infeasible in the sense of manufacturability can occur. To avoid this, filtering techniques and mesh regularization have to be applied [BDF06, DCB04]. By filtering, a filter with a certain radius is applied on the gradients which smoothens the gradients inside the filter radius. The size of the filter radius determines the wavelength of possible waves in the solution. If the filter radius is chosen too small, the result can still be a very irregular shape. If it is chosen too large, the solution might be constrained too much and be far from optimal. Therefore, the decision for a certain filter radius determines the possible solutions, similar to choosing a parametrization for the CAD approach. However, the finite element approach still has a significantly larger design space, and hence more different and locally adapted solutions are possible. Therefore, this approach is applicable to cases where no coarse shape shall be predefined for the solution and where local load-carrying behavior shall be encouraged, such as in bead optimization for thin metal sheets.

The fact that the finite element mesh has to be created only once and no further transfer between CAD model and FE model is necessary can be seen as a gain in efficiency compared to the CAD approach. On the other hand, it is obvious that this gain is compensated by the huge number of design variables and the necessity for additional operations (filtering, regularization).

An important aspect in choosing design variables for shape optimization is the question of prescribed directions. If nodal coordinates are used as design variables, it is reasonable to allow only variations which really vary the shape of the structure. This is explained best on the example of a flat plate. For any node on the inside, i.e. not on the plate's boundary, a variation tangential to the plate has no influence on the shape of the plate. A bad consequence is that these nodes can move "freely" on the plate during optimization and create a distorted mesh. To prevent this, only variations perpendicular to the surface are allowed. The variations that influence the shape depend on the topological position of the node. While for all inner nodes only variations perpendicular to the surface are of interest, for nodes on the edges only the variation parallel to the edge is to be prevented and for vertex nodes, all variations have influence on the shape.

For the inner nodes on curved surfaces, like shell structures, only a variation perpen-

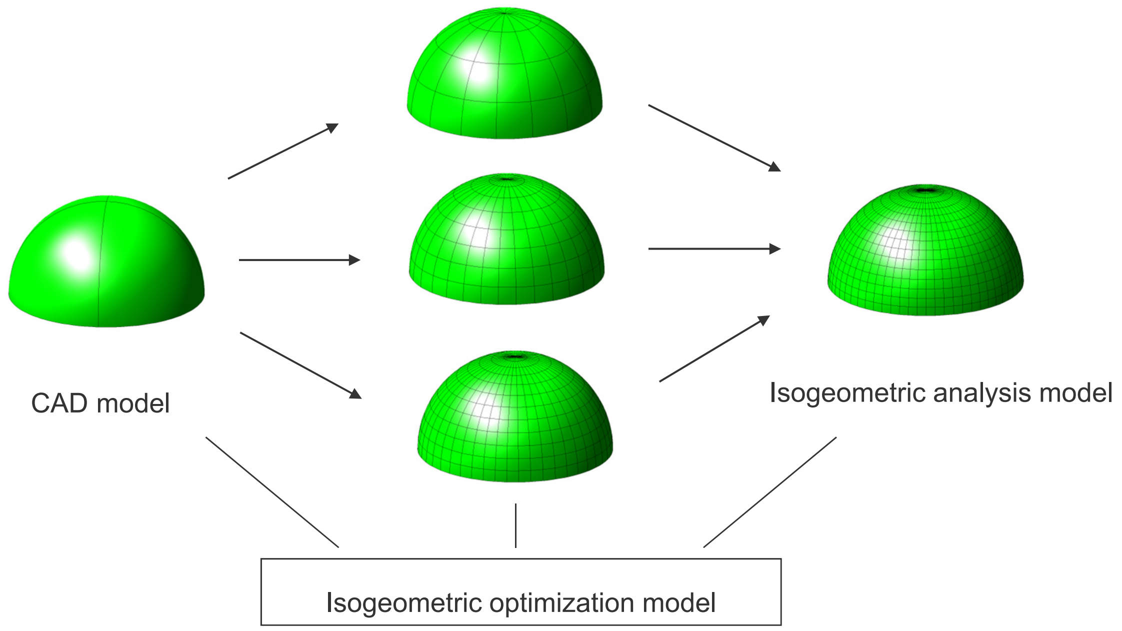


Figure 9.1: Isogeometric shape optimization. Optimization can be performed on the original CAD model, on the refined analysis model, or on any refinement level in between.

dicular to the tangent plane affects the shape. Being precise, also a variation inside the tangent plane has a small effect on the shape, due to the curvature. However, this contribution is negligible and variations are usually prescribed to be perpendicular to the surface. Obviously, this normal direction changes during the optimization procedure, i.e. the "prescribed" direction for each node has to be updated in each optimization step.
Allowing variations only in normal direction is not only applicable for FE nodes as design variables but also for the control nodes of a CAD optimization model. In CAD-based shape optimization, the problem of in-plane movements of nodes is less severe since the control nodes have a bigger region of influence and therefore also variations in the tangent plane have a considerable effect on the shape. Therefore, it is not generally necessary to restrict the variations to the direction normal to the plane. Nevertheless, variations in the tangent plane still have significantly less effect on the shape than perpendicular variations, and they can cause mesh distortion. Therefore, also for optimization on the CAD model, it is often advantageous to prescribe the variations to be in normal direction.

9.5 Isogeometric Shape Optimization

As has been shown in the previous chapters, in the isogeometric concept the CAD geometry and the analysis model rely on the same geometry description. As a consequence, the aforementioned distinction between CAD-based and FE-based shape optimization

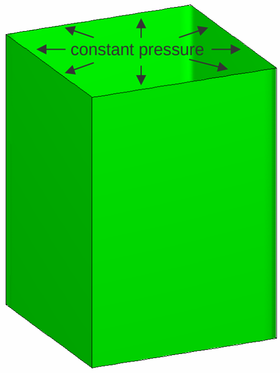


Figure 9.2: Tube with square cross section under constant internal pressure.

can be omitted using isogeometric analysis. It has also been shown in the previous chapters that for obtaining reliable results in isogeometric analysis, refinement of the original NURBS model is necessary. This means that it still can be distinguished between the design model, which is the geometry in a coarse NURBS representation, and the analysis model, which is the same geometry in a refined NURBS representation. For NURBS-based shape optimization this means that the optimization can be carried out both on the design model and on the analysis model, since both are NURBS geometries. Furthermore, an additional model with a refinement level between the design and the analysis model can be introduced. So there are three NURBS models of the same geometry with different refinement levels, namely the design model, the optimization model and the analysis model. In the first step, the design model is refined yielding the optimization model and in the second step the latter is refined yielding the analysis model. The two refinement steps are independent of each other, which means that a certain refinement for the analysis model can be chosen independently of the optimization model and vice versa. Figure 9.1 illustrates the model hierarchy for isogeometric shape optimization on the example of a half-sphere. The CAD model is represented by a very coarse NURBS description. In a first refinement step, the optimization model is defined, where the level of refinement can be chosen freely. In a second refinement step, the analysis model is obtained. Since, in both steps, it is also possible to perform no refinement, the two classical approaches of optimizing on the CAD model and optimizing on the analysis model are included in this approach as well. Thus, this approach is an enhancement to the existing approaches, including both of them as special cases.

This approach also allows for an adaptive optimization. A coarse optimization model can be used to determine the global shape of the structure accounting for global loads. Further optimization on refined models can then be performed to account for local loads.

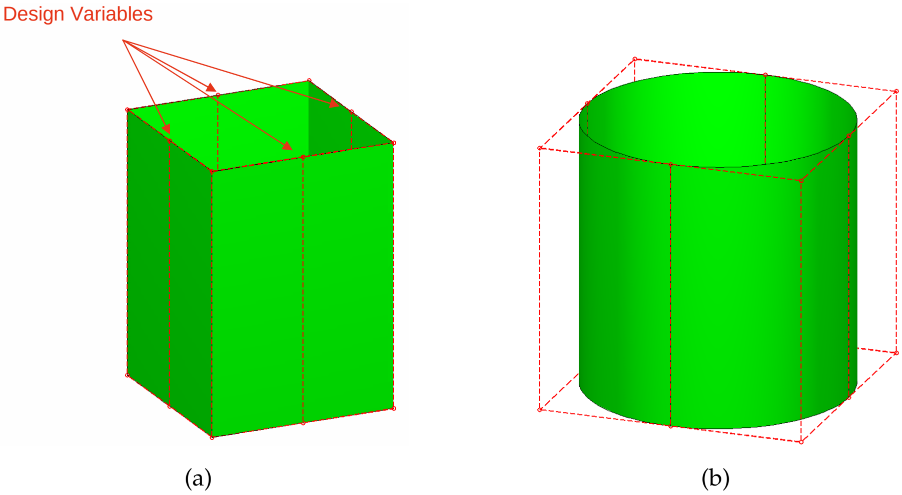


Figure 9.3: Tube under internal pressure. (a) Design variables: x-, y-coordinates and the weights of the indicated control points. (b) Optimal geometry after shape optimization.

9.5.1 Example: Tube under internal pressure

In the following, an example is presented which demonstrates the abilities of isogeometric shape optimization. A tube with a square cross section under constant internal pressure is given, Figure 9.2. The cross-sectional shape shall be optimized with respect to linear compliance, i.e. internal energy. In Figure 9.3(a), the NURBS parametrization of the tube is shown and the design variables are indicated. The cross section is modeled by a quadratic NURBS with nine control points. In the figure, only eight control points can be seen, since the first and the last control point coincide. As indicated, the design variables are the control points at the middle of the edges. In detail, it is the x- and y-coordinates (the cross section lies in the x-y-plane) of these control points, and also their weights. The cross-sectional shape shall remain constant along the tube, therefore the design variables apply to both the top and the bottom of the tube. Symmetry inside the cross-sectional plane is not used, i.e. the control points on all four sides are independent design variables.

The result of the shape optimization is depicted in Figure 9.3(b). As to be expected, the optimal shape of the cross section is a circle, because a circle carries the constant internal pressure by pure membrane stresses without bending moments. It is pointed out that the resulting shape is an exact circle which is possible by using NURBS for the geometry description. This result can be obtained only by including the weights of the control points as design variables. However, it must also be stated that this example is an academic example where the optimal solution is known in advance, and this optimal shape needs the adjustment of the weights. For the shape optimization of free form shapes it can be reasonable to use only control point coordinates and not the weights, since the influence of weights on the shape is less compared to the influence of the coordinates. This is es-

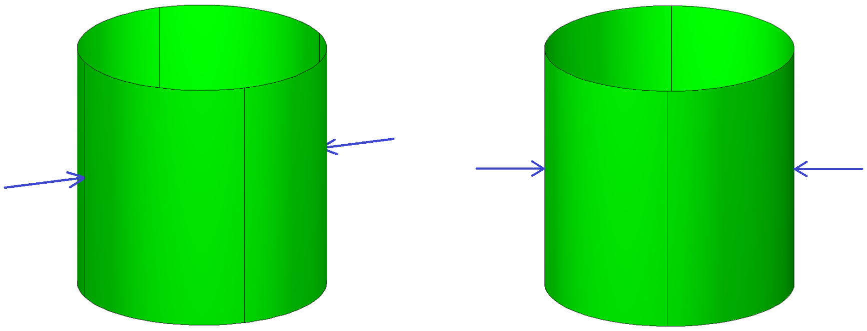


Figure 9.4: Cylindrical tube with local loads.

pecially the case if the optimization is performed on a rather fine NURBS model with a dense control grid.

Remark: As has been explained in Chapter 4, the weights of the control points are not used as degrees of freedom for isogeometric analysis. Nevertheless, they can be used as design variables for isogeometric shape optimization since parameters of optimization and analysis model are not identical. The weights can be varied for a design update, while inside each optimization step an isogeometric analysis is performed where weights remain constant.

The shape optimization shown above was performed on a very coarse NURBS description which was adequate to account for the globally constant pressure load. In the next step, local loads are applied and optimization is performed on refined optimization models where different levels of refinement are employed. Figure 9.4 shows the setup. The starting geometry is now the geometry obtained by the first shape optimization, i.e. a cylinder. Two opposite point loads are applied at the middle of the cylinder's height. The NURBS model of the cylinder is refined in both circular and height direction. The control points over the height are now independent design variables as well. The direction of the design variation is prescribed to be perpendicular to the surface and weights are not used as design variables.

Remark: Since NURBS control points do not lie on the surface in general, the question is how to assign them a direction which is normal to the surface. In the present example, this normal direction is determined by vector products of the neighboring control points. An alternative would be to find the closest point on the surface by point projection, evaluate the normal there and assign it to the control point. This, however, would be much more time-consuming since for every point projection a Newton iteration has to be solved.

Two different refinements are used as optimization model. The first case is depicted in Figure 9.5. Subfigure (a) shows the control grid of the optimization model indicated by dashed lines. For analysis, additional refinement is performed, as indicated in Figure 9.5(b). It shows the resulting geometry after shape optimization. Buckles have formed under the point loads to carry the load primarily by membrane forces. Due to the rather

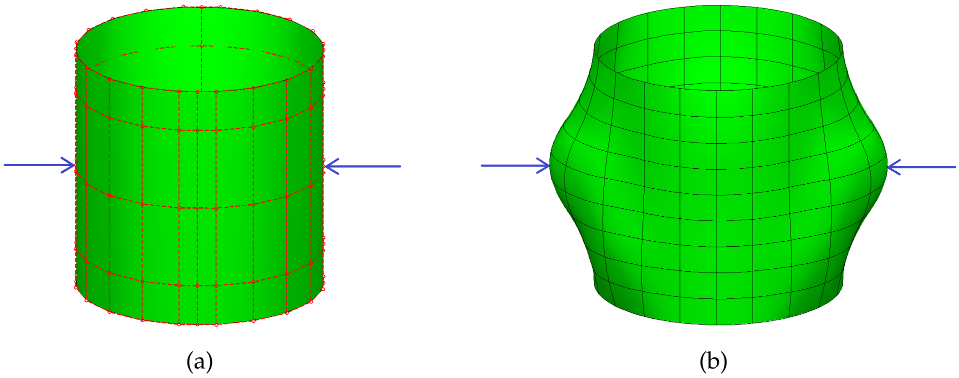


Figure 9.5: Cylindrical tube with local loads, coarse refinement for optimization model. (a) Control grid of the optimization model. (b) Result of shape optimization. Buckles form under the point loads, which affects the global shape.

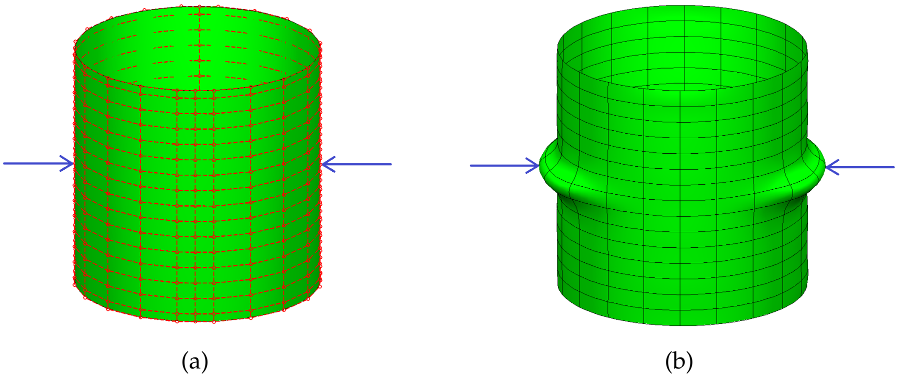


Figure 9.6: Cylindrical tube with local loads, fine refinement for optimization model. (a) Control grid of the optimization model. (b) Result of shape optimization. The load is carried locally by small beads.

coarse optimization model, the buckles are relatively big and affect the whole shape of the cylinder.
In the next case, a very fine optimization model is chosen, see Figure 9.6(a). The point loads can now be captured more locally by small beads which affect only a small region around the point loads, as shown in Figure 9.6(b). Here, the same refinement has been used for optimization and analysis.

In a classical FE-based shape optimization, the optimized geometry is a faceted finite element geometry. Such a geometry cannot be used as design model for manufacturing and therefore this optimized geometry must be approximated by a continuous surface, e.g. by NURBS. Obviously, this is another challenging step. An appropriate model has to be chosen to accurately reproduce the shape and in any case geometric information is lost due to the approximation. Using isogeometric shape optimization, the result of the

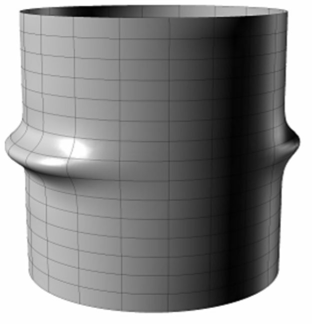


Figure 9.7: The optimized geometry from Figure 9.6(b) as design geometry in a CAD environment.

optimization is a NURBS geometry and therefore it can be used as the design model for production without model conversion. This is illustrated in Figure 9.7 which shows the optimized geometry from Figure 9.6(b) as design geometry in a CAD environment. At this point it is repeated that in isogeometric shape optimization all involved models are based on a common geometry description. These models are: the initial design model, the optimization model, the analysis model and the final design model.

Chapter 10

FSI Simulation of a Wind Turbine Blade

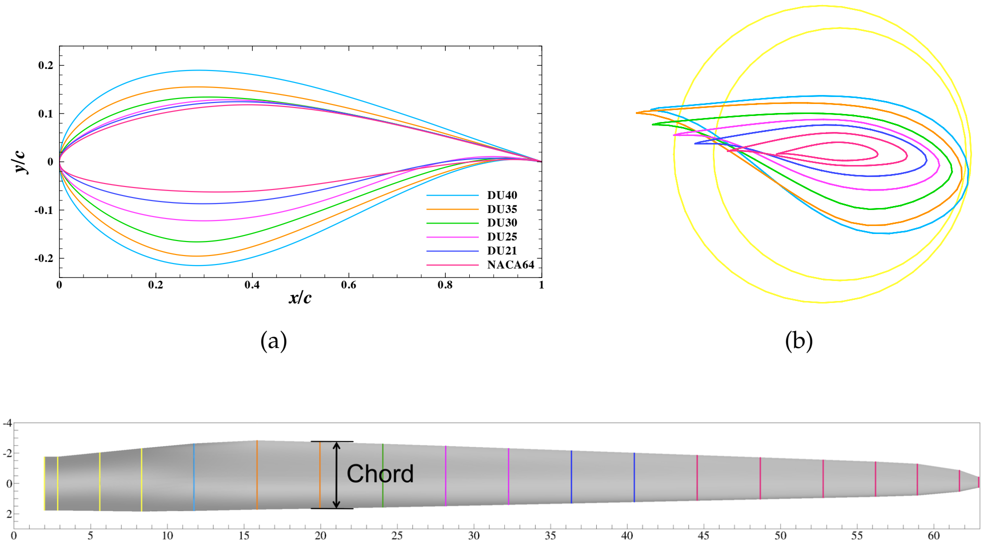
In this chapter, the applicability of the isogeometric Kirchhoff-Love shell for coupled Fluid-Structure Interaction (FSI) computations is shown and at the same time it is an application of the method to a realistic large-scale structure. It is demonstrated on the fully coupled FSI analysis of a 63 m long offshore wind turbine blade which is rotating and deflecting under the wind loading. This work has been done in cooperation with Professor Bazilevs at the Department of Structural Engineering at the University of California, San Diego. Professor Bazilevs and his group use NURBS-based isogeometric fluid analysis for a three-dimensional simulation of the air flow around rotating turbine blades.

At the moment, the common practice for the design of wind turbine blades is to use the steady-state, two-dimensional aerodynamic models for airfoil sections on a set of cross sections along the blade. With these models, the air pressure on the blades is determined, which defines the aerodynamic torque on the one hand, as well as the forces acting on the blade for structural analysis on the other hand. Obviously, these simplified models cannot capture three-dimensional and time-dependent effects of the fluid or the interaction between blade deformation and the air flow. For a better prediction of both the aerodynamic efficiency and the structural behavior of the blades, these effects can be studied precisely by a numerical 3D FSI-simulation.

For this joint project, the isogeometric Kirchhoff-Love shell and the bending strips, as presented in this thesis, have been implemented into the FSI environment of Professor Bazilevs. This allows a detailed structural analysis of the wind turbine blade inside the FSI analysis of the rotating and deflecting turbine blade. The results are presented in detail in [BHK$^+$10].

10.1 Geometry Description

The turbine blade under consideration is related to a NREL 5MW offshore baseline wind turbine described in detail in [JBMS09]. It is described by a set of airfoil sections which vary along the blade's length and a cylindrical cross section at the hub. The blade is 61 m long and attached to a hub with a radius of 2 m so the total rotor radius is 63 m. The cross sections switch from circular at the hub into to a series of DU (Delft University) profiles and then to the NACA64 profile. Figure 10.1(a) shows the definition of the different airfoil profiles, in Figure 10.1(b) the profiles are depicted in their actual size and orientation, and 10.1(c) indicates their position on the blade. These cross sections are modeled by NURBS curves and from these curves a NURBS surface is created by lofting. The surface is partitioned into 20 NURBS patches which is due to the decomposition of the fluid domain for parallel computing. The wind turbine consists of three blades, but due to symmetry only one blade in conjunction with appropriate boundary conditions is considered in the simulation.



(a)

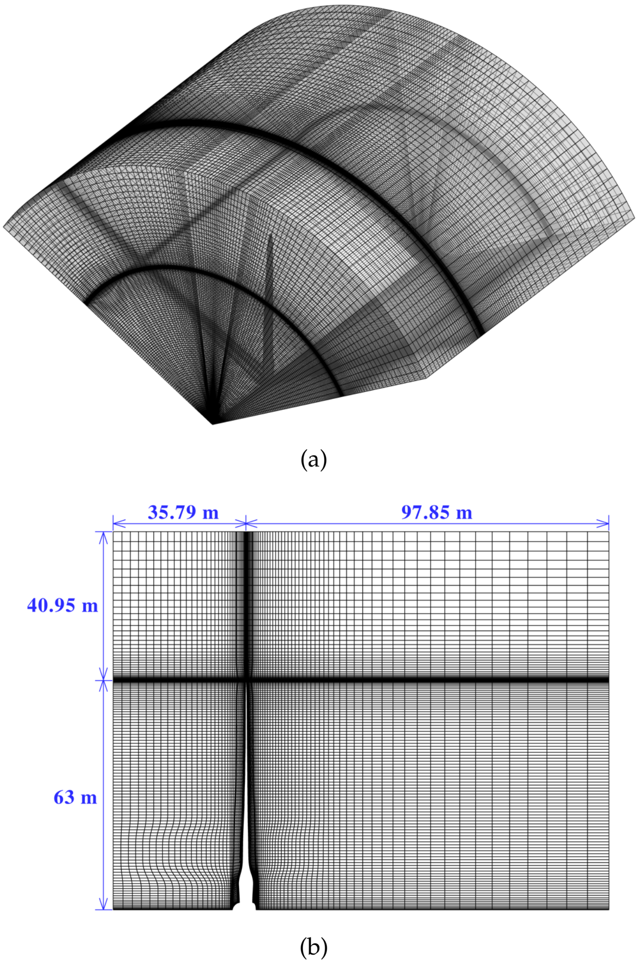
(b)

(c)

Figure 10.1: Airfoil cross-sections. (a) Definition of different airfoil profiles. (b) Airfoil cross sections of the wind turbine blade. (c) Distribution of the profiles along the blade.

10.2 Fluid Mechanics and Mesh Motion Part

The details of the fluid computations are presented in [BHA$^+$10]. Its basics are reviewed here shortly for completeness but not in detail since this is not in the scope of this thesis. For the fluid computations in the moving domain, the Arbitrary-Lagrangian-Eulerian (ALE) description is employed. Turbulence is modeled by the residual-based variational multiscale method [BCC$^+$07, BMCH10] which has proved to be very accurate for turbulence modeling, especially for rotating flows [BA10, BHA$^+$10]. The large dimensions and high flow velocities with Reynolds numbers in the Range of $10^7 - 10^8$ in this example are very challenging for the fluid computations and fine grids are necessary. NURBS-based



(a)

(b)

Figure 10.2: (a) Fluid mesh around the blade, consisting of 1,449,000 volumetric NURBS elements. (b) The mesh is refined at the blade's surface.

fluid analysis has proved to be more accurate and efficient compared to low-order finite elements for these computations [BHA$^+$10].

The mesh motion is divided into the parts due to rotation and blade deflection. The rotational part is handled exactly while the mesh motion due to the deflection of the blade is computed using the equations of linear elasticity.

Due to symmetry, only one third (i.e. 120°) of the whole cylindrical domain is modeled, see Figure 10.2. The fluid mesh is constructed around the blade by volumetric NURBS and consists of 1,449,000 quadratic NURBS elements.

10.3 Structural Mechanics Part

The NURBS-based Kirchhoff-Love shell turns out to be a very appropriate choice for the structural analysis in this coupled problem for several reasons:
The turbine blade represents a thin and slender structure which undergoes large deformations. Therefore, a locking-free shell with a geometrically nonlinear formulation is needed. The smooth, curved geometry of the blade can be exactly represented by the NURBS-based elements which is important for a correct buckling behavior of the shell.

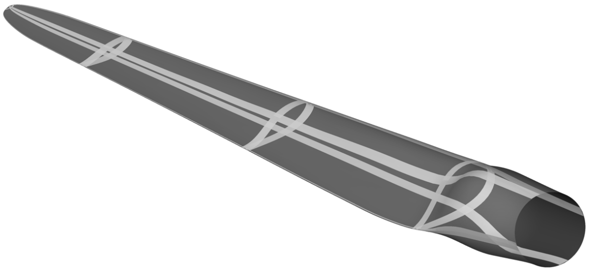


Figure 10.3: Structural model of the blade.

As shown in Chapter 8, the NURBS surface description of the blade can be used as structural model for the isogeometric shell without further modification and since the fluid mesh is built from this surface model, the meshes at the interface match and no mapping between meshes is necessary. For a coupling without any mapping, it is also necessary that the control points on the fluid and the structural side have the same degrees of freedom, which is the case here, since the shell formulation is rotation-free. Thus, a clear advantage of the presented shell formulation for this coupled computation is the fact that it matches to the discretization of the fluid field which in turn has proved to be very efficient and accurate for rotating and highly turbulent flows as appearing in this problem.
In [HCB05], it is described that flow simulations around airfoil shapes are very sensitive to geometrical imperfections of the airfoil, i.e. small kinks due to a piecewise linear approximation of the geometry. According to [Bar98], spurious entropy layers occur in the case of linear geometry approximation and they disappear for smooth geometry descriptions, even if the flow field still is discretized by linear polynomials. Thus, for the FSI simulation of the blade, which in fact is composed by airfoil cross sections, a smooth description of the structure by NURBS-based elements is crucial for a good behavior of the flow around the blade.
The blade's surface model as described above is used as the structural shell model of the blade. The "inner parts" of the blade, such as spar webs or spar boxes, have been neglected in this first model. For a comprehensive overview of different structural types for rotor blades, reference is made to [Hau06]. According to the geometry model, the shell consists of 20 quadratic NURBS patches which are connected by a total of eight bending

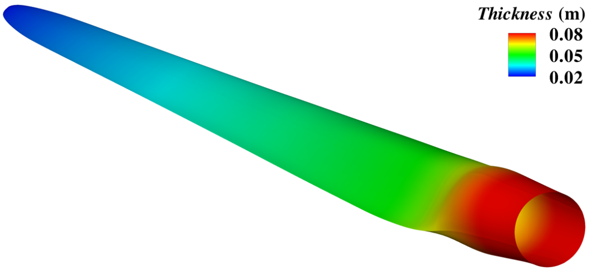


Figure 10.4: Blade thickness

strips: Four bending strips in the longitudinal direction and four bending strips in the tangential direction, see Figure 10.3. It can be seen that this creates overlaps between the longitudinal and tangential bending strips which indeed does not impose problems, as has been described in Chapter 7. It must be noted that the last four patches and the corresponding bending strip are very small and close to the tip so they are hard to capture in the figure. As mentioned above, the division of the patches is governed by the division of the fluid mesh.

For the material, a multilayer fiberglass-epoxy composite with a symmetric lay-up of 16 plies with equal thickness and the orientation $[\pm 45/0/90_2/0_3]_s$ is employed. The lay-up description gives the orientation angle of the fibers for each ply, starting with the outermost ply. The notation ± 45 stands for two plies, one with $+45^\circ$ and one with -45°. A subscript number, as in 90_2 and 0_3 represents two and three layers, respectively, with the same orientation, and the subscript "s" at the end is for symmetric. The angles of the fibers are referred to the circumferential direction of the blade. The material is modeled by laminated plate theory as described in Section 3.3, i.e. the composite is represented by a homogenized material. All plies use the same material properties and differ only in their orientation. The orthotropic material properties for a ply are gathered in Table 10.1. The blade's thickness distribution is illustrated in Figure 10.4. For simplicity, the number of plies is assumed to be constant along the blade, and their thickness decreases with decreasing shell thickness.

E_1 (GPa)	E_2 (GPa)	G_{12} (GPa)	ν_{12}	ρ (kg/m^3)
39	8.6	3.8	0.28	2.1

Table 10.1: Material properties of a unidirectional E-glass/epoxy composite [DI94].

10.4 Results

The computation has been performed for a wind speed of 11.4 m/s and a rotor angular velocity of 12.1 rpm (rotations per minute), as illustrated in Figure 10.5. These values correspond to one of the cases reported in [JBMS09].

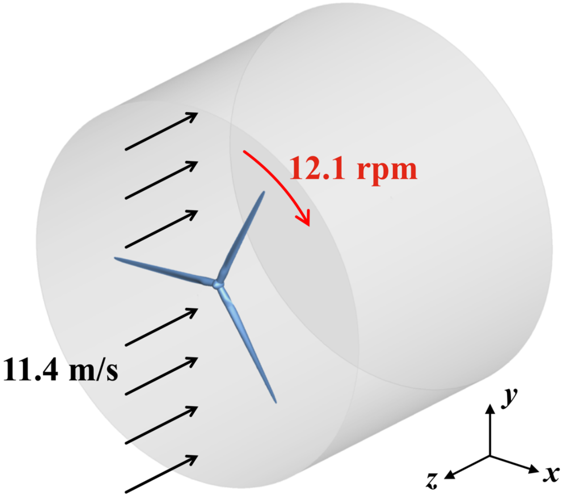


Figure 10.5: Setup of the problem.

In Figure 10.6, one can see the deforming blade in the flow field at a time $t = 0.7$ s. The airflow is visualized by the isocontours of air speed at a planar cut through the turbine's center and the deformation of the blade can clearly be seen. Figure 10.7 shows the deforming blade at four different time steps with isocontours visualized on a plane at a radial distance of 30 m. The maximum displacement appears at a time $t = 0.7$ s. Therefore, the results at this time step are used for a detailed structural analysis in the sequel.
The maximum displacement of the blade tip is 5.9 m in flap-wise direction, see Figure 10.8, which is consistent with the data reported in [JBMS09]. The maximum stresses appear in the blade's longitudinal direction. Figure 10.9 shows the stress resultant n_{22}. This is the normal force in longitudinal direction and it shows the main load carrying behavior of the blade: There is tension at the front side and compression at the back side of the blade, as to be expected. On both sides, the highest forces (in absolute values) appear close to the leading edge and the normal force decreases towards the tip.
The normal forces are useful to study the structural behavior of the blade. In order to check the material for failure, the stresses in the laminate are evaluated and compared with the maximum allowable values for each ply. For a fiber-reinforced composite material, the material strength is different in fiber direction and transverse direction, and it is also different with respect to tension and compression. Consequently, there are five maximum values to check for each ply: the maximum tension and compression in fiber direction, σ_1^t and σ_1^c, as well as the maximum tension and compression in transverse di-

Ply no.	Position	σ_1^t	σ_1^c	σ_2^t	σ_2^c	$\lvert\sigma_{12}\rvert$
16 (45°)	Top	60.774	-61.631	17.312	-19.949	14.344
	Bottom	57.601	-58.019	15.857	-18.663	13.930
15 ($-45°$)	Top	70.223	-75.571	14.823	-15.644	13.949
	Bottom	63.786	-70.405	14.097	-14.712	13.553
14 (0°)	Top	40.400	-52.065	22.629	-19.840	5.490
	Bottom	34.075	-43.919	22.376	-19.744	5.045
13 (90°)	Top	105.350	-91.728	9.242	-11.688	5.061
	Bottom	104.635	-92.211	7.842	-9.737	4.616
12 (90°)	Top	104.635	-92.211	7.842	-9.737	4.616
	Bottom	103.968	-92.693	6.443	-9.028	4.171
11 (0°)	Top	21.426	-36.385	21.982	-19.717	4.157
	Bottom	19.018	-36.384	21.813	-19.746	3.771
10 (0°)	Top	19.018	-36.384	21.813	-19.746	3.771
	Bottom	16.995	-36.384	21.750	-19.790	3.851
9 (0°)	Top	16.995	-36.384	21.750	-19.790	3.851
	Bottom	15.047	-36.384	21.686	-19.834	3.931
8 (0°)	Top	15.047	-36.384	21.686	-19.834	3.931
	Bottom	15.070	-36.384	21.623	-19.877	4.193
7 (0°)	Top	15.070	-36.384	21.623	-19.877	4.193
	Bottom	23.381	-36.383	21.652	-19.937	4.490
6 (0°)	Top	23.381	-36.383	21.652	-19.937	4.490
	Bottom	32.250	-36.383	21.682	-20.024	4.821
5 (90°)	Top	100.120	-95.590	7.235	-8.837	4.826
	Bottom	99.655	-96.072	9.065	-8.837	5.304
4 (90°)	Top	99.655	-96.072	9.065	-8.837	5.304
	Bottom	99.189	-96.555	10.895	-8.837	5.781
3 (0°)	Top	49.988	-36.383	21.740	-20.198	5.776
	Bottom	58.856	-36.757	21.792	-20.453	6.253
2 ($-45°$)	Top	60.694	-52.822	16.683	-12.365	12.956
	Bottom	64.454	-55.162	17.496	-12.712	13.717
1 (45°)	Top	70.418	-47.956	15.694	-13.656	13.731
	Bottom	73.840	-49.168	16.658	-14.237	14.511

Table 10.2: Maximum stresses in [MPa] for each ply, at the time step of maximum tip displacement.

$\sigma_1^{t,u}$ (MPa)	$\sigma_1^{c,u}$ (MPa)	$\sigma_2^{t,u}$ (MPa)	$\sigma_2^{c,u}$ (MPa)	σ_{12}^{u} (MPa)
1080	620	39	128	89

Table 10.3: Tensile, compressive, and shear strengths of a unidirectional E-glass/epoxy composite lamina [DI94].

Ply no.	Position	$\frac{\sigma_1^t}{\sigma_1^{t,u}}$	$\frac{\sigma_1^c}{\sigma_1^{c,u}}$	$\frac{\sigma_2^t}{\sigma_2^{t,u}}$	$\frac{\sigma_2^c}{\sigma_2^{c,u}}$	$\frac{\lvert\sigma_{12}\rvert}{\sigma_{12}^u}$	Tsai-Wu
16 (45°)	Top	0.056	0.099	0.444	0.156	0.161	0.360
	Bottom	0.053	0.094	0.407	0.146	0.157	0.325
15 (−45°)	Top	0.065	0.122	0.380	0.122	0.157	0.287
	Bottom	0.059	0.114	0.361	0.115	0.152	0.270
14 (0°)	Top	0.037	0.084	0.580	0.155	0.062	0.513
	Bottom	0.032	0.071	0.574	0.154	0.057	0.508
13 (90°)	Top	0.098	0.148	0.237	0.091	0.057	0.160
	Bottom	0.097	0.149	0.201	0.076	0.052	0.130
12 (90°)	Top	0.097	0.149	0.201	0.076	0.052	0.130
	Bottom	0.096	0.150	0.165	0.071	0.047	0.101
11 (0°)	Top	0.020	0.059	0.564	0.154	0.047	0.497
	Bottom	0.018	0.059	0.559	0.154	0.042	0.494
10 (0°)	Top	0.018	0.059	0.559	0.154	0.042	0.494
	Bottom	0.016	0.059	0.558	0.155	0.043	0.491
9 (0°)	Top	0.016	0.059	0.558	0.155	0.043	0.491
	Bottom	0.014	0.059	0.556	0.155	0.044	0.488
8 (0°)	Top	0.014	0.059	0.556	0.155	0.044	0.488
	Bottom	0.014	0.059	0.554	0.155	0.047	0.485
7 (0°)	Top	0.014	0.059	0.554	0.155	0.047	0.485
	Bottom	0.022	0.059	0.555	0.156	0.050	0.483
6 (0°)	Top	0.022	0.059	0.555	0.156	0.050	0.483
	Bottom	0.030	0.059	0.556	0.156	0.054	0.482
5 (90°)	Top	0.093	0.154	0.186	0.069	0.054	0.124
	Bottom	0.092	0.155	0.232	0.069	0.060	0.162
4 (90°)	Top	0.092	0.155	0.232	0.069	0.060	0.162
	Bottom	0.092	0.156	0.279	0.069	0.065	0.202
3 (0°)	Top	0.046	0.059	0.557	0.158	0.065	0.480
	Bottom	0.054	0.059	0.559	0.160	0.070	0.479
2 (−45°)	Top	0.056	0.085	0.428	0.097	0.146	0.334
	Bottom	0.060	0.089	0.449	0.099	0.154	0.351
1 (45°)	Top	0.065	0.077	0.402	0.107	0.154	0.308
	Bottom	0.068	0.079	0.427	0.111	0.163	0.329

Table 10.4: Maximum stress criterion and Tsai-Wu criterion for each ply, at the time step of maximum tip displacement.

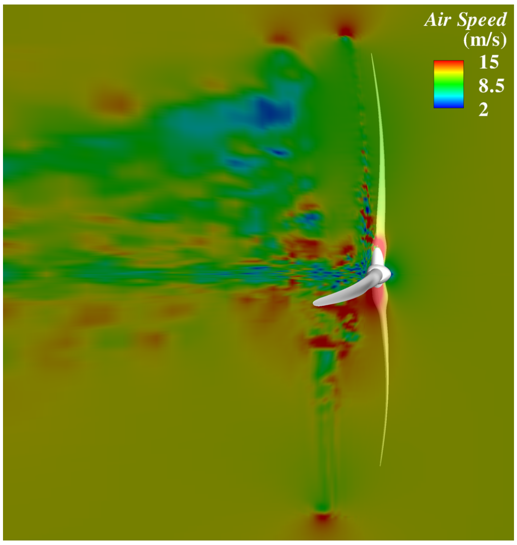
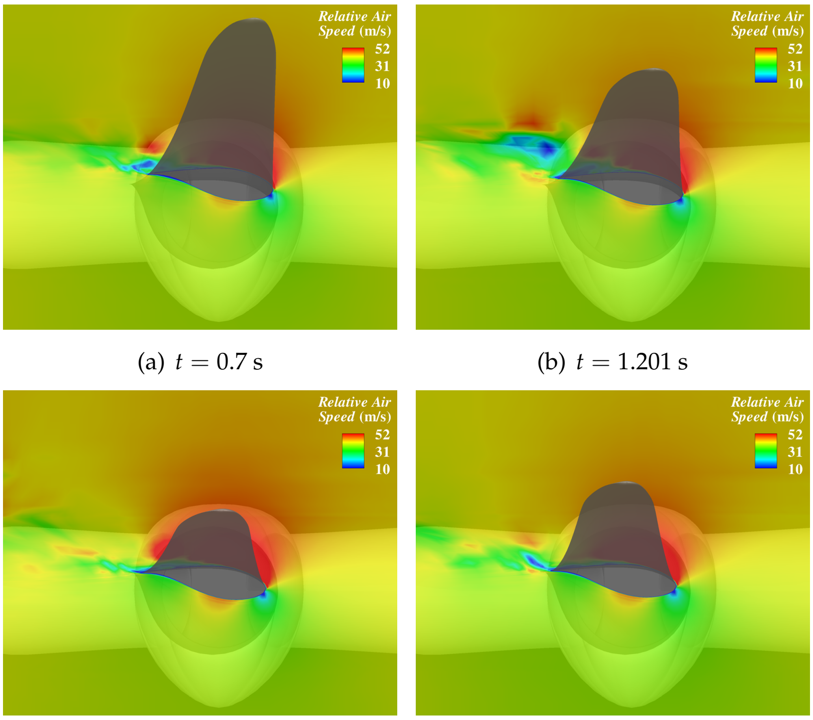


Figure 10.6: The deforming blade in the air flow, visualized by isocontours of air velocity, at $t = 0.7$ s. The single-blade results are assembled to the whole turbine consisting of three blades.

rection, σ_2^t and σ_2^c, and the maximum in-plane shear stress σ_{12}. In Table 10.2, these results are listed for each ply, evaluated at the bottom and the top of each ply, since it is not clear in advance where the maximum stress appears. The material strength values are given in Table 10.3, where $\sigma_1^{t,u}$ and $\sigma_1^{c,u}$ are the longitudinal tensile and compressive strength, respectively, $\sigma_2^{t,u}$ and $\sigma_2^{c,u}$ are the transverse tensile and compressive strength, respectively, and σ_{12}^u is the in-plane shear strength. The ratios of maximum stresses over respective strengths for all plies are listed in Table 10.4. Furthermore, the Tsai-Wu criterion is given. This widely used failure criterion computes one value that takes into account the interaction of the different stresses, but it does not provide information about the failure mode. For fulfilling the Tsai-Wu criterion, this value has to be between minus one and one. For more details, reference is made to [DI94]. As can be seen in Table 10.4, all stress criteria are fulfilled in all plies. The maximum Tsai-Wu value appears in ply no.14. The highest value in the Maximum Stress criterion appears for σ_2^t, also in ply no.14. Figure 10.10 shows the stress distribution for σ_{22} in ply no.14.

For investigating the aerodynamic efficiency of the blade, the aerodynamic torque is plotted over time in Figure 10.11. For validation, the torque obtained by the FSI simulation is compared with the reference value reported in [JBMS09]. Furthermore, the result from the computation with the blade modeled as a rigid structure is depicted. It can be seen that both cases are in good accordance with the reference value. However, there are also clear differences between the rigid and the flexible case, which shows that also in terms of design for aerodynamic efficiency, it is important to take the blade's structural behavior into account. It can be seen that the torque undergoes oscillations for the flexible blade



(a) $t = 0.7$ s

(b) $t = 1.201$ s

(c) $t = 1.7$ s

(d) $t = 2.4165$ s

Figure 10.7: Deformation of the blade and the air flow at 30 m radial distance, at four different time steps.

that stem from vortex shedding at the trailing edge, which in turn causes twisting deformations of the blade. For a more detailed explanation, reference is made to [BHK$^+$10].

For the computations presented here, a simplified structural model has been used, which considers only the shell of the rotor blade. For more realistic results, the inner parts of the blade, such as spars, need to be included in the model. Nevertheless, the results obtained are in good accordance with the results reported in the literature [JBMS09].

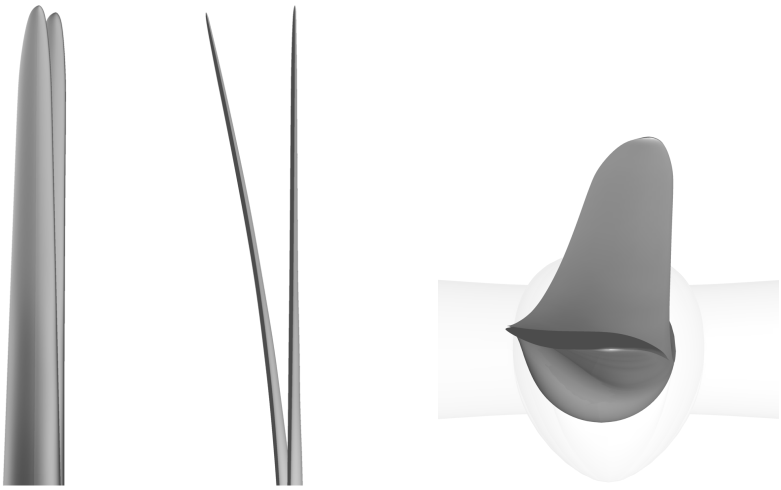
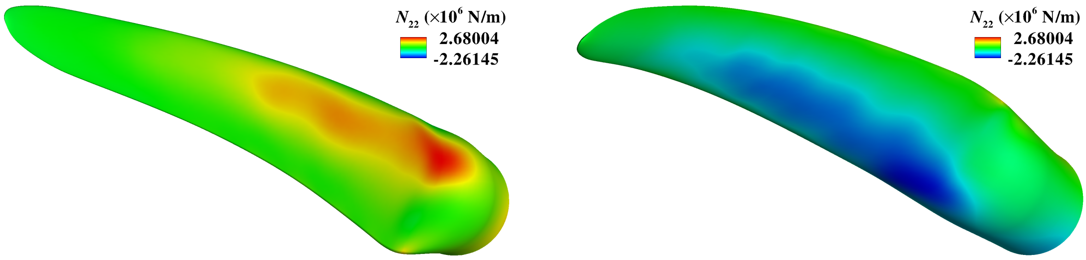
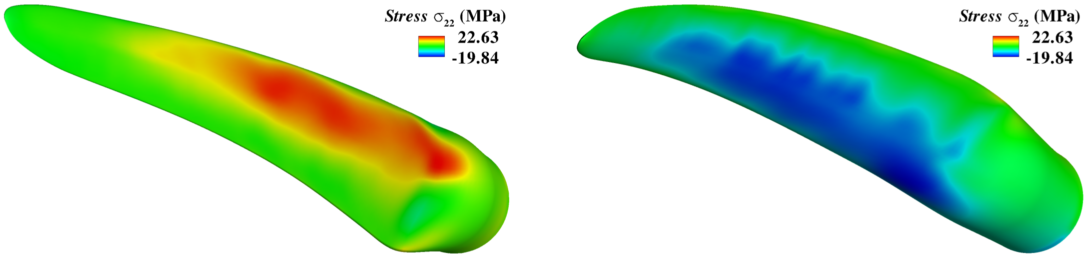


Figure 10.8: Front, side and top view of the blade deflection at the point of maximum tip displacement.



(a) (b)

Figure 10.9: Normal force in longitudinal direction of the blade at the time step of maximum tip displacement. (a) Front view. (b) Rear view.



(a) (b)

Figure 10.10: Normal stress in ply no.14, in longitudinal direction of the blade at the time step of maximum tip displacement. (a) Front view. (b) Rear view.

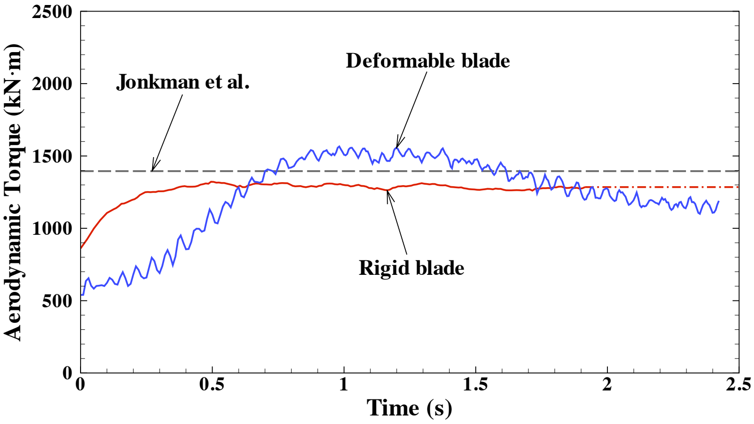


Figure 10.11: Aerodynamic torque

Chapter 11

Conclusions and Outlook

In this thesis, the isogeometric concept has been applied to shell analysis and shape optimization. A new shell element has been developed based on the Kirchhoff-Love shell theory, with NURBS as basis functions. Hereby, the Kirchhoff-Love theory and NURBS as basis functions have proved as an excellent combination. On the one hand, NURBS are the ideal basis functions for a Kirchhoff-Love shell, since they provide the necessary C^1 continuity between elements. They allow a straightforward implementation of the Kirchhoff-Love theory without additional modifications, which is not possible with standard finite elements using low-order polynomials. Due to the exclusion of transverse shear and due to the higher order nature of NURBS, this element is free from the typical shell locking phenomena. Various benchmark tests have shown its good performance, even in the case of very large deformations.

In order to ensure the necessary continuities over patch boundaries, two different methods have been developed. For smooth patches, i.e. patches that are G^1 continuous at the interface, this continuity can be maintained by a direct coupling of the respective degrees of freedom via master-slave constraints. For patches that meet at a certain angle, i.e. there is a kink at the interface, the necessary constraint equations describe nonlinear relations between the degrees of freedom and therefore an exact compliance by direct coupling is not possible. The bending strip method has been developed which fulfills the angle constraint in an approximate sense. It has been tested on a series of examples and proven to be reliable, and it can handle both smooth patches and patches forming a kink. The advantages of this method are flexibility and simplicity in both implementation and application. Due to the regular topology of NURBS, the bending strips can be created automatically. For the integration into CAD, this means that they do not need to be added in the design model, but are created automatically during analysis. Furthermore, this method can be used for the coupling of shell and solid elements.

The work for this thesis focused on structural analysis, but within a cooperation project with Prof. Bazilevs (University of California, San Diego) it was integrated into a fluid-structure interaction environment and was successfully applied to the three-dimensional FSI simulation of an offshore wind turbine blade. This project also shows the relevance of this work for industrial applications.

The isogeometric concept furthermore has been extended to shape optimization. Due to the common geometry description of design and analysis, the traditional distinction between CAD-based and FE-based shape optimization is superfluous. Therefore, isogeometric shape optimization combines the advantages of both approaches and gives more possibilities in defining the design space. In this thesis, the principle of isogeometric shape optimization has been shown and it has been successfully applied to examples with different refinement levels for the optimization model. Nevertheless, there are still interesting aspects open to be studied in future research. Different methods known from FE-based optimization could be applicable to the isogeometric approach as well, such as filtering and weighted sensitivities. The question of weighted sensitivities is more complicated for NURBS than for standard finite elements. Control points have different influence on the shape, depending on the corresponding shape function. An interpolated control point, e.g. a vertex point, has more influence on the shape than a non-interpolated point. On the other hand, a vertex node has a smaller area of influence. The correct combination of these effects for an appropriate weighting of sensitivities is a complicated task and should be part of further research.

As mentioned in the first paragraph, NURBS have proved to be an ideal basis for a Kirchhoff-Love shell formulation. On the other hand, the Kirchhoff-Love shell has also emerged as very well-suited for effectively integrating analysis into CAD. CAD programs usually work with surface description, which means that for using isogeometric solid elements, the surface model needs to be transformed into a volumetric model, which is a complicated task and subject of actual international research. While for volumetric objects this is indispensable, for thin-walled structures a shell formulation enables the analysis on the CAD surface model. With a rotation-free shell formulation as presented in this thesis, design and analysis actually use the same geometry description. This makes it easy to effectively integrate isogeometric analysis into a CAD environment, which has been demonstrated by integrating the developed method into *Rhino*, a commercial CAD program. Nevertheless, the applicability of this approach is still very limited, due to the fact that it is restricted to untrimmed NURBS surfaces. Trimming however, is a basic method in CAD modeling and complex geometries can hardly be constructed without it. There is actual research on analysis of trimmed NURBS surfaces [KSY09, KSY10], which at the moment is restricted to plane 2D geometries. Alternatively, trimmed NURBS can be converted into untrimmed T-Splines [SFLL08] which also allow for local refinement. Isogeometric analysis using T-Splines has been successfully performed in [DJS10, BCC$^+$10, UY09, UKY08, BBDL$^+$10]. However, there are still open questions about the linear independence of T-Splines, as discussed in [BCS10] and [SLZ$^+$10]. Concluding, it can be said that using NURBS as basis for analysis is a first step towards the integration of design and analysis. And it is a step towards this goal from the FE community. Significant changes are also necessary in CAD modeling which concern the creation of analysis-suitable geometries. For a successful integration of isogeometric analysis, the CAD and FE communities will have to work together more closely in order to understand each other's demands on a geometric model and to agree on a common isogeometric representation.

List of Figures

List of Tables

Bibliography

[AB97] D.N. Arnold and F. Brezzi. Locking-free finite element methods for shells. *Mathematics of Computations*, 66(217):1–14, 1997.

[aba] www.simulia.com/.

[AH79] J. S. Arora and E. J. Haug. Methods of design sensitivity analysis in structural optimization. *AIAA Journal*, 17(9):970–974, 1979.

[Aro89] J. S. Arora. *Introduction to Optimum Design*. McGraw-Hill Series in Mechanical Engineering. McGraw-Hill, New York, NY, USA, 1989.

[BA10] Y. Bazilevs and I. Akkerman. Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual–based variational multiscale method. *Journal of Computational Physics*, 229:3402–3414, 2010.

[Bai07] H. Baier. Faserverbundwerkstoffe. Lecture notes, 2007.

[Bar98] T. J. Barth. *Simplified Numerical Methods for Gas Dynamics Systems on Triangulated Domains*. Ph.D. Thesis, Department of Aeronautics and Astronautics, Stanford University, 1998.

[BBDL$^+$10] D. J. Benson, Y. Bazilevs, E. De Luycker, M.-C. Hsu, M. Scott, T. J. R. Hughes, and T. Belytschko. A generalized finite element formulation for arbitrary basis functions: From isogeometric analysis to xfem. *International Journal for Numerical Methods in Engineering*, 83:765–785, 2010.

[BBHH10] D.J. Benson, Y. Bazilevs, M.C. Hsu, and T.J.R. Hughes. Isogeometric shell analysis: The Reissner–Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199:276–289, 2010.

[BBR00] K.-U. Bletzinger, M. Bischoff, and E. Ramm. A unified approach for shear-locking free triangular and rectangular shell finite elements. *Computers and Structures*, 75:321–334, 2000.

[BCC$^+$07] Y. Bazilevs, V.M. Calo, J.A. Cottrel, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.

[BCC$^+$10] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:229–263, 2010.

[BCHZ08] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computatonal Mechanics*, 43:3–37, 2008.

[BCS10] A. Buffa, D. Cho, and G. Sangalli. Linear independence of the T-spline blending functions associated with some particular T-meshes. *Comput. Methods Appl. Mech. Engrg.*, 199:1437–1445, 2010.

[BCZH06] Y. Bazilevs, V.M. Calo, Y. Zhang, and T.J.R. Hughes. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38:310–322, 2006.

[BDF06] K.-U. Bletzinger, F. Daoud, and M. Firl. Filter techniques in shape optimization with CAD-free parametrization. In *III European Conference on Computational Mechanics Solids, Structures andCoupled Problems in Engineering*, 2006.

[BFD08] K.-U. Bletzinger, M. Firl, and F. Daoud. Approximation of derivatives in semi-analytical structural optimization. *Computers & Structures*, 86:1404–1416, 2008.

[BFLW08] K.-U. Bletzinger, M. Firl, J. Linhard, and R. Wüchner. Optimal shapes of mechanically motivated surfaces. *Computer Methods in Applied Mechanics and Engineering*, 2008.

[BH08] Y. Bazilevs and T.J.R. Hughes. NURBS-based isogeometric analysis for the computation of flows about rotating components. *Computatonal Mechanics*, 43:143–150, 2008.

[BHA$^+$10] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T.E. Tezduyar. 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids*, 2010. Published online, doi:10.1002/fld.2400.

[BHK$^+$10] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid-structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 2010. Published online, doi:10.1002/fld.2400.

[Bis99] Manfred Bischoff. *Theorie und Numerik einer dreidimensionalen Schalenformulierung*. PhD thesis, Institut für Baustatik, Universität Stuttgart, 1999.

[BK85a] Y. Basar and W. Krätzig. *Mechanik der Flächentragwerke*. Vieweg, Braunschweig, 1985.

[BK85b] K.-U. Bletzinger and S. Kimmich. *Strukturoptimierung*. Sonderforschungsbereich 230. Natürliche Konstruktionen. Universität Stuttgart und Tübingen, 1985.

[Ble90] K.-U. Bletzinger. *Formoptimierung von Flächentragwerken*. PhD thesis, Universität Stuttgart, 1990.

[Ble93] K.-U. Bletzinger. Form finding of shells by structural optimization. *Engineering with Computers*, 9:27–35, 1993.

[Ble08] K.-U. Bletzinger. Theory of plates. Lecture notes, 2008.

[BMCH10] Y. Bazilevs, C. Michler, V.M. Calo, and T.J.R. Hughes. Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly-enforced boundary conditions on unstretched meshes. *Computer Methods in Applied Mechanics and Engineering*, 2010. doi:10.1016/j.cma.2008.11.020.

[BR01] K.-U. Bletzinger and E. Ramm. Structural optimization and form finding of light weight structures. *Computers and Structures*, 79:2053–2062, 2001.

[Bra98] D. Braess. Enhanced assumed strain elements and locking in membrane problems. *Computer Methods in Applied Mechanics and Engineering*, 165:155–174, 1998.

[Bre10] M. Breitenberger. *Isogeometrische Analyse in Carat++*. Master's thesis, Technische Universität München, 2010.

[BS92a] I. Babuska and M. Suri. Locking effects in the finite element approximation of elasticity problems. *Numerische Mathematik*, 62:439–463, 1992.

[BS92b] I. Babuska and M. Suri. On locking and robustness in the finite element method. *SIAM J. Numer. Anal.*, 29:1261–1293, 1992.

[BSL$^+$85] T. Belytschko, H. Stolarski, W.K. Liu, N. Carpenter, and J.S.-J. Ong. Stress projection for membrane and shear locking in shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 51:221–258, 1985.

[BSMM00] I.N Bronstein, K.A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 2000.

[BW00] Y. Basar and D. Weichert. *Nonlinear continuum mechanics of solids*. Springer Verlag, Berlin, Heidelberg, New York, 2000.

[BWBR04] M. Bischoff, W.A. Wall, K.-U. Bletzinger, and E. Ramm. Models and finite elements for thin-walled structures. In *Encyclopedia of Computational Mechanics*, volume 2, Solids, Structures and Coupled Problems, chapter 3. Wiley, 2004.

[BWDC05] K.-U. Bletzinger, R. Wüchner, F. Daoud, and N. Camprubí. Computational methods for form finding and optimization of shells and membranes. *Computer Methods in Applied Mechanics and Engineering*, 194:3438–3452, 2005.

[Béz66] P. Bézier. Définition numérique de courbes et surfaces. *Automatisme*, 11:625–632, 1966.

[CdSNJFVAA02] J. M. A. César de Sá, R. M. Natal Jorge, R. A. Fontes Valente, and P. M. Almeida Areias. Development of shear locking-free shell elements using an enhanced assumed strain formulation. *International Journal for Numerical Methods in Engineering*, 53:1721–1750, 2002.

[CH09] S. Cho and S-H. Ha. Isogeometric shape design optimization: exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization*, 38:53–70, 2009.

[Cha04] K.-J. Chapelle, D. & Bathe. *The Finite Element Analysis of Shells - Fundamentals*, volume 47. Springer, 2004.

[CHB09] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009.

[CHR07] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196:4160–4183, 2007.

[CMK$^+$10] E. Cohen, T. Martin, R. M. Kirby, T. Lyche, and R.F. Riesenfeld. Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 199:334–356, 2010.

[CO01] F. Cirak and M. Ortiz. Fully C^1-conforming subdivision elements for finite deformation thin–shell analysis. *International Journal for Numerical Methods in Engineering*, 51:813–833, 2001.

[COS00] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin shell analysis. *International Journal for Numerical Methods in Engineering*, 47:2039–2072, 2000.

[CRBH06] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5297, 2006.

[Dao05] F. Daoud. *Formoptimierung von Freiformschalen*. PhD thesis, Techische Universität München, 2005.

[DCB04] F. Daoud, N. Camprubí, and K.-U. Bletzinger. Filtering and regularization techniques in shape optimization with cad-free parametrization. In *NATO Advanced Research Workshop on Multiphysics and Multiscale Computer Models in Nonlinear Analysis and Optimal Design of Engineering Structures under Extreme Conditions*, Bled, Slovenia, 2004.

[DI94] I.M. Daniel and O. Ishai. *Engineering Mechanics of Composite Materials*. Oxford University Press, New York, NY, 1994.

[DJS10] M. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:264–275, 2010.

[EBCH08] T. Elguedj, Y. Bazilevs, V.M. Calo, and T.J.R. Hughes. Bbar and Fbar projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer Methods in Applied Mechanics and Engineering*, 197:2732–2762, 2008.

[Fel04] C.A. Felippa. Introduction to finite element methods. Course material. Department of Aerospace Engineering Sciences, University of Colorado at Boulder, 2004.

[FHK02] G. Farin, J. Hoschek, and M.S. Kim. *Handbook of Computer Aided Geometric Design*. Elsevier Science Ltd, 2002.

[Gel88] M. Gellert. A new method for derivation of locking-free plate bending finite elementsvia mixed/hybrid formulation. *International Journal for Numerical Methods in Engineering*, 26:1185–1200, 1988.

[Hau06] E. Hau. *Wind Turbines: Fundamentals, Technologies, Application, Economics. 2nd Edition*. Springer, Berlin, 2006.

[HBC$^+$10] M.C. Hsu, Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes. Improving stability of multiscale formulations of fluid flow at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199:828–840, 2010.

[HCB05] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[HCC10] S-H. Ha, K.K. Choi, and S. Cho. Numerical method for shape optimization using T-spline based isogeometric method. *Structural and Multidisciplinary Optimization*, 42:417–428, 2010.

[Hol04] G.A. Holzapfel. *Nonlinear solid mechanics*. John Wiley & Sons, Chichester, 2004.

[HRS10] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:301–313, 2010.

[Hug00] T.J.R. Hughes. *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Dover Publications Inc., 2000.

[JBMS09] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, Golden, CO, 2009.

[Kab92] H.R.H. Kabir. A shear locking free isoparametric three-node triangular finite element for moderately-thick and thin plates. *International Journal for Numerical Methods in Engineering*, 35:503–519, 1992.

[KBCB04] F. Koschnick, M. Bischoff, N. Camprubí, and K.-U. Bletzinger. The discrete strain gap method and membrane locking. *Computer Methods in Applied Mechanics and Engineering*, 194:2444–2463, 2004.

[KBH$^+$10] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199:2403–2416, 2010.

[KBLW09] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.

[Kir50] G. Kirchhoff. Über das gleichgewicht und die bewegung einer elastischen scheibe. *Journal für reine angewandte Mathematik*, 40:51–58, 1850.

[KJ03] W.B. Krätzig and D. Jun. On 'best' shell models – from classical shells, degenerated and multi-layeredconcepts to 3d. *Archive of Applied Mechanics*, 73:1–25, 2003.

[Kli89] E. Klingbeil. *Tensorrechnung fur Ingenieure*. Bibliographisches Institut & F.A Brockhaus AG, Zürich, 2. edition, 1989.

[Kos04] F. Koschnik. *Geometrische Lockingeffekte bei Finiten Elementen und ein allgemeines Konzept zu ihrer Vermeidung*. PhD thesis, Techische Universität München, 2004.

[KSY09] H.-J. Kim, Y.-D. Seo, and S.-K. Youn. Isogeometric analysis for trimmed CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 198:2982–2995, 2009.

[KSY10] H.-J. Kim, Y.-D. Seo, and S.-K. Youn. Isogeometric analysis with trimming technique for problems of arbitrary complex topology. *Computer Methods in Applied Mechanics and Engineering*, published online, 2010.

[LEB$^+$10] S. Lipton, J. A. Evans, Y. Bazilevs, T. Elguedj, and T. J. R. Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Comput. Methods Appl. Mech. Engrg.*, 199:357–373, 2010.

[Lee05] K.-J. Lee, P.-S. & Bathe. Insight into finite element shell discretizations by use of the 'basic shell mathematical model'. *Computers & Structures*, 83:69–90, 2005.

[Lov88] A.E.H. Love. On the small vibrations and deformations of thin elastic shells. *Philosophical Transactions of the royal Society*, 179:491 ff., 1888.

[LWB07] J. Linhard, R. Wüchner, and K.-U. Bletzinger. "Upgrading" membranes to shells–The CEG rotation free shell element and its application in structural analysis. *Finite Elements in Analysis and Design*, 44:63–74, 2007.

[L.Y98] L.Yunhua. Explanation and eliminatin of shear locking and membrane locking with fieldconsistence approach. *Computer Methods in Applied Mechnics and Engineering*, 162:249–269, 1998.

[MH85] R.H. MacNeal and R.L. Harder. A proposed standard set of problems to test finite element accuracy. *Finite Elements in Analysis and Design*, 1:3–20, 1985.

[OnF05] E. Oñate and F. G. Flores. Advances in the formulation of the rotation-free basic shell triangle. *Comput. Methods Appl. Mech. Engrg.*, 194:2406–2443, 2005.

[OnZ00] E. Oñate and F. Zárate. Rotation-free triangular plate and shell elements. *International Journal for Numerical Methods in Engineering*, 47:557–603, 2000.

[PT97] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, 2nd edition, 1997.

[RDN$^+$04] E. Rank, A. Düster, V. Nübel, K. Preusch, and O.T. Bruhns. High order finite elements for shells. *Computer Methods in Applied Mechanics and Engineering*, 194:2494–2512, 2004.

[Red04] J.N. Reddy. *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis, 2nd ed.* CRC Press, Boca Raton, FL, 2004.

[Rei45] E. Reissner. The effect of transverse shear deformation on the bending on of elastic plates. *Journal of Applied Mechanics*, pages A–69 – A–77, 1945.

[rhi] www.rhino3d.com.

[Rog01] D.F. Rogers. *An Introduction to NURBS With Historical Perspective*. Academic Press, San Diego, CA, 2001.

[SB82] H. Stolarski and T. Belytschko. Membrane locking and reduced integration for curved elements. *Journal of Applied Mechanics*, 49:172–176, 1982.

[SCF04] T. W. Sederberg, D.L. Cardon, and G.T. Finnigan. T-Spline simplification and local refinement. *ACM Transactions on Graphics*, 23:276–283, 2004.

[Sch05] H. Schürmann. *Konstruieren mit Faser–Kunststoff–Verbunden*. Springer-Verlag, Berlin, 2005.

[Sch09] R. Schmidt. *Development and Implementation of a Rhino Preprocessor for the Integration of CAD and Isogeometric Analysis*. Master's thesis, Technische Universität München, 2009.

[SDR04] B. Szabó, A. Düster, and E. Rank. The p-version of the finite element method. In *Encyclopedia of Computational Mechanics*, pages 119–140. John Wiley & Sons, Ltd., Chichester, 2004.

[SFLL08] T.W. Sederberg, G.T. Finnigan, X. Li, and H. Lin. Watertight trimmed NURBS. In *Proceedings of SIGGRAPH 08*, 2008.

[SFMH08] R. Sevilla, S. Fernández-Méndez, and A. Huerta. NURBS-enhanced finite element method (NEFEM). *International Journal for Numerical Methods in Engineering*, 76:56–83, 2008.

[SKBW10] R. Schmidt, J. Kiendl, K.-U. Bletzinger, and R. Wüchner. Realization of an integrated structural design process: analysis–suitable geometric modelling and isogeometric analysis. *Computing and Visualization in Science*, accepted, 2010.

[SLZ$^+$10] T.W. Sederberg, X. Li, J. Zheng, T.J.R. Hughes, and M.A. Scott. On linear independence of T-Splines. Technical report, ICES, University of Texas at Austin, 2010.

[SR84] K. Schweizerhof and E. Ramm. Displacement dependent pressure loads in nonlinear finite element analyses. *Computers & Structures*, 18:1099–1114, 1984.

[SZBN03] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-Splines and T-NURCCS. *ACM Transactions on Graphics*, 22:477–484, 2003.

[UKY08] T.-K. Uhm, Y.D. Kim, and S.-K. Youn. A locally refinable T-spline finite element method for cad/cae integration. *Structural Engineering and Mechanics*, 80:225–245, 2008.

[UY09] T.-K. Uhm and S.-K. Youn. T-spline finite element method for the analysis of shell structures. *International Journal for Numerical Methods in Engineering*, 80:507–536, 2009.

[WFC08] W.A. Wall, M.A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197:2976–2988, 2008.

[WP03] W. Wunderlich and W. D. Pilkey. *Mechanics of Structures, Variational and Computational Methods*. CRC Press, 2nd edition, 2003.

[YSMK00] H.T.Y. Yang, S. Saigal, A. Masud, and R. K. Kapania. A survey of recent shell finite elements. *International Journal for Numerical Methods in Engineering*, 47:101–127, 2000.

[ZTZ05] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method, Its Basis and Fundamentals*. Elsevier, Oxford, 6th edition, 2005.