Technische Universität München

Ingenieurfakultät Bau Geo Umwelt

Lehrstuhl für Statik

CAD-integrated Design and Analysis of Shell Structures

Michael Breitenberger

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:

Univ.-Prof. Dr. rer. nat. Ernst Rank

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Kai-Uwe Bletzinger
2. Prof. Dr. Thomas J. R. Hughes
3. Univ.-Prof. Dr.-Ing. Josef Kiendl

Die Dissertation wurde am 22.06.2016 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 16.11.2016 angenommen.

Schriftenreihe des Lehrstuhls für Statik
TU München

Band 34

Michael Breitenberger

CAD-INTEGRATED DESIGN AND ANALYSIS OF SHELL STRUCTURES

München 2016

Veröffentlicht durch

Kai-Uwe Bletzinger
Lehrstuhl für Statik
Technische Universität München
Arcisstr. 21
80333 München

Telefon:	+49(0)89 289 22422
Telefax:	+49(0)89 289 22421
E-Mail:	kub@tum.de
Internet:	www.st.bgu.tum.de

ISBN: 978-3-943683-41-7

Abstract

A new CAD-integrated design-through-analysis workflow for shell structures, named *Analysis in Computer Aided Design* (AiCAD), is presented. In contrast to existing workflows AiCAD uses the CAD geometry description throughout the entire workflow. Contemporary CAD systems mainly use a *non-uniform rational B-Splines* (NURBS)-based *boundary representation* (B-Rep) for the description of geometries. The usage of such models within AiCAD requires an analysis technique which is able to deal with arbitrarily complex NURBS-based B-Rep models. For this purpose a new finite element technique is developed with the name *isogeometric B-Rep analysis* (IBRA). IBRA provides the framework for creating a direct and complete analysis model from CAD in a consistent finite-element-like manner. The newly developed B-Rep elements are used to handle the discontinuous and trimmed geometries incl. gaps and overlaps for structural analysis. Thus, IBRA allows analyzing surface CAD models without remodeling and meshing, even for highly complex geometries. Various numerical examples including real industrial problems confirm the accuracy, flexibility, and robustness of IBRA and thus highlight the advantages of the realization of a design-through-analysis workflow with a uniform geometry representation.

The proposed AiCAD concept allows bridging the gap between CAD systems and FE programs efficiently by using a new analysis approach which is able to handle CAD models. AiCAD is realized within several commercial CAD systems and can compete with established analysis techniques used in industry.

Zusammenfassung

Ein neuer CAD-integrierter Prozess für den Entwurf und die Berechnung von Schalenstrukturen mit dem Namen *Analysis in Computer Aided Design* (AiCAD) wird vorgestellt. Im Gegensatz zu anderen Verfahren verwendet AiCAD ausschließlich CAD Modelle für die Geometriebeschreibung. Um CAD Modelle für Strukturanalysen verwenden zu können, wurde die *isogeometrische B-Rep Analyse* (IBRA), eine neue Finite-Elemente-Methode (FEM), entwickelt. IBRA liefert den Rahmen für die Erstellung eines kompletten Berechnungsmodells auf Basis eines *Non-Uniform Rational B-Splines* (NURBS)-basierten *Boundary Representation* (B-Rep) Modells. Solche Modelle werden in modernen CAD Systemen am häufigsten verwendet. IBRA verwendet neu entwickelte B-Rep Elemente, um die diskontinuierliche und getrimmte Geometrie unter Berücksichtigung von Lücken und Überlappungen für Strukturanalysen zugänglich zu machen. Damit ermöglicht IBRA die direkte Berechnung von komplexen CAD Flächenmodellen in einem FE Programm ohne Vernetzung und ohne zusätzlichen geometrischen Modellierungsaufwand. Eine Vielzahl an numerischen Beispielen inkl. industrieller Problemstellungen demonstrieren die Genauigkeit, Flexibilität und Robustheit von IBRA und unterstreichen die Vorteile eines CAD-basierten Berechnungsprozesses.

Das AiCAD Konzept erlaubt eine effiziente Vereinigung von CAD und FEM durch die Verwendung einer neuen FEM Berechnungsmethode, welche CAD Modelle für die Geometriebeschreibung verwendet. AiCAD wurde in unterschiedlichen CAD Systemen implementiert und erweist sich gegenüber in der Industrie etablierten Berechnungsmethoden als konkurrenzfähig.

Acknowledgments

This thesis was written from 2011 to 2016 during my time at the *Chair of Structural Analysis* (Lehrstuhl für Statik) at the Technische Universität München, Munich, Germany.

First of all, I would like to thank sincerely Prof. Dr.-Ing. Kai-Uwe Bletzinger for proving me the possibility to work in this very interesting field of research and guiding me successfully through my time as a doctoral candidate. I also want to express my thanks to Dr.-Ing. Roland Wüchner for his great support and for initiating my interest on isogeometric analysis after the exam of "Introduction to Finite Element Methods" in 2009.

Furthermore, I would like to address my thanks to Prof. Dr. Thomas J. R. Hughes, Prof. Dr.-Ing. Josef Kiendl, and Prof. Dr. rer. nat. Ernst Rank for completing my board of examiners. Their interest in my research are gratefully appreciated.

Very special thanks goes also to my coworkers at the Chair of Structural Analysis for the friendly cooperation and the pleasant time that I had working with them.

At this point I would like to express my deepest gratitude to my parents Anna Maria and Leo. Without their absolute support it would not have been possible for me to successfully finish this chapter in my life.

Finally, I want to thank my entire family especially Anna, Johannes, and Suzy for their support and patience.

Michael Breitenberger
Technische Universität München
November, 2016

List of Symbols

Calligraphic letters

$\mathcal{I}_\mathrm{H}$	Height of integration domain
$\mathcal{I}_\mathrm{W}$	Width of integration domain
$\mathcal{D}$	Trimmed domain of a geometry
$\tilde{\mathcal{D}}$	Trimmed domain of a trimming curve within the parameter space of a surface
$\partial\mathcal{D}$	Boundary of a trimmed domain
$\mathcal{H}$	Parameter domain of a geometry
$\mathcal{I}$	Integration domain

Greek letters

α_disp	Penalty factor for the displacement
α_rot	Penalty factor for the rotation
$\delta\boldsymbol{\epsilon}$	Virtual normal strain derived from Green-Lagrange strain tensor
$\delta\boldsymbol{\kappa}$	Virtual change in curvature derived from Green-Lagrange strain tensor
η	Parameter for B-Splines resp. NURBS geometries
$\mathit{\Gamma}$	Boundary of the physical (shell) domain in the reference configuration

$\mathit{\Gamma}_i$	$(i)^{\text{th}}$ isogeometric B-Rep edge element
$\boldsymbol{\omega}$	Rotation vector
$\boldsymbol{\Xi}$	Knot vector in ξ direction
$\mathbf{H}$	Knot vector in η direction
$\mathit{\Omega}$	Physical (shell) domain in the reference configuration
$\omega_{\mathbf{T}_2}$	Rotation around boundary vector $\mathbf{T}_2$
$\mathit{\Omega}_{ij}$	$(i,j)^{\text{th}}$ isogeometric (trimmed) element of a patch
θ^i	Contravariant coordinates
θ_i	Covariant coordinates
$\tilde{\mathit{\Gamma}}$	Boundary of the shell domain within parameter space
$\tilde{\mathit{\Omega}}$	Shell domain within parameter space
ξ	Parameter for B-Splines resp. NURBS geometries
$\tilde{\xi}$	Parameter for B-Splines resp. NURBS curves within parameter space

Mathematical symbols and operators

$\mathbb{N}$	Set of natural numbers
$\square \otimes \square$	Dyadic product
$\mathbb{R}$	Set of real numbers
$(\square \cdot \square)$	Inner product
$(\cdot)_{,i}$	Partial derivatives w.r.t. to a quantity i
$\left\lVert(\cdot)\right\rVert_2$	L^2-norm

Latin letters

$\bar{T}_i$	Length of vector $\check{\mathbf{T}}_i$

$\check{\mathbf{T}}_i$	Orthogonal local coordinate system aligned to a boundary
$\Delta\mathbf{u}$	Displacement increment
$\delta\mathbf{u}_\mathrm{h}$	Virtual discretized displacement vector
δW	Virtual work
δW_ext	External virtual work
δW_int	Internal virtual work
$\delta\mathbf{u}$	Virtual displacement field
$\hat{\mathbf{u}}^r$	r^th displacement discretization parameter
$\hat{\mathbf{X}}^r$	r^th geometry discretization parameter in reference configuration
$\hat{\mathbf{x}}^r$	r^th geometry discretization parameter in current configuration
$\mathbf{A}$	Metric tensor on surface
$\mathbf{A}^i$	Contravariant basis on surface
$\mathbf{A}_i$	Covariant basis on surface
$\mathbf{C}$	B-Spline resp. NURBS curve
$\mathbf{C}_\mathrm{visible}$	Set of all visible points of a trimmed curve
$\mathbf{e}_i$	Global orthonormal coordinate system
$\mathbf{K}$	Tangential stiffness matrix
$\mathbf{M}$	Moment derived from 2$^\mathrm{nd}$ Piola Kirchhoff (PK2) stress tensor
$\mathbf{N}$	Normal force derived from 2$^\mathrm{nd}$ Piola Kirchhoff (PK2) stress tensor
$\mathbf{P}$	Control point
$\mathbf{p}$	External force
$\mathbf{R}$	Residual force vector
$\mathbf{S}$	B-Spline resp. NURBS surface or 2$^\mathrm{nd}$ Piola Kirchhoff (PK2) tensor

$\mathbf{S}_{\text{visible}}$	Set of all visible points of a trimmed surface
$\mathbf{T}_i$	Orthonormal local coordinate system aligned to a boundary
$\mathbf{u}$	Displacement field
$\mathbf{w}$	Displacement of the tip of the local base vector $\mathbf{t}_3$
$\mathbf{X}_{\text{B-Rep}}$	Spatial point on a surface boundary
$\mathbf{X}_{\text{surf}}$	Spatial point on a surface
$\partial\,\mathbf{S}_{\text{E}}$	Set of points of an edge
$\partial\,\mathbf{S}_{\text{visible}}$	Boundary of trimmed B-Spline resp. NURBS surface
$\mathbf{n}_{\Gamma}$	Traction force at the boundary derived from 1st Piola Kirchhoff (PK1) stress tensor
$\tilde{\mathbf{C}}$	Trimming curve in parameter space
$\tilde{\mathbf{e}}_i$	Local orthonormal coordinate system aligned to a trimming curve within the parameter space of a surface
$\tilde{\mathbf{P}}$	Control point in parameter space
$\tilde{\mathbf{T}}$	Tangent vector of a trimming curve $\tilde{\mathbf{C}}$ within the parameter space of a surface
$\tilde{w}$	Weighting factor used for numerical integration
A^{ij}	Contravariant metric coefficients on surface
A_{ij}	Covariant metric coefficients on surface
$B_{\alpha\beta}$	Covariant coefficients of curvature tensor
c	Coupling
d	Dirichlet
J	Jacobian used for different mapping
K_{rs}	$(r,s)^{\text{th}}$ component of the stiffness matrix

$m_{\mathbf{T}_2}$	Moment around the boundary tangent $\mathbf{T}_2$ derived from 1st Piola Kirchhoff (PK1) stress tensor
n	Neumann
$n_{\square}$	Number of entity $\square$
p	Polynomial degree in ξ-direction
q	Polynomial degree in η-direction
R_r	r^{th} component of the residual force vector
t	Thickness of a shell
w	Control point weight or quadrature weight
M	(B-Spline) basis function in η direction
N	(B-Spline) basis function in ξ direction
R	NURBS basis function

Abbreviations:

E	Edge
F	Face
V	Vertex
AGIP	Adaptive Gaussian integration procedure
AiCAD	Analysis in Computer Aided Design
API	Application programming interface
B-Rep	Boundary Representation
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing

CSG	Constructive Solid Geometry
Ctrlpts	Control points
FCM	Finite Cell Method
FE	Finite element
FEA	Finite element analysis
FEM	Finite Element Method
IBRA	Isogeometric B-Rep analysis
IGA	Isogeometric analysis
IGES	Initial Graphics Exchange Specification
ISO	International Organization for Standardization
KL	*Kirchhoff-Love* shell theory
NURBS	Non-Uniform Rational B-Splines
RM	*Reissner-Mindlin* shell theory
STEP	Standard for the exchange of product model data
TeDA	Tool to enhance Design by Analysis
TUM	Technische Universität München

Contents

We can't solve problems by using the same kind of thinking we used when we created them.

Albert Einstein

Chapter 1

Introduction

The usage of different geometry descriptions for design and analysis is one of the bottlenecks in a virtual product development process. Historically, the independent developments in the fields *computer-aided design* (CAD) and *computer-aided engineering* (CAE) caused this "gap". The predominant method in CAE for solving structural problems is the *finite element method* (FEM). Since its origins in the 1950s FEM was developed primarily with linear respectively quadratic polynomials defined over non-overlapping *parametric domains* (the elements). The standard for geometry description in contemporary CAD systems, however, is based on a *non-uniform rational B-Splines* (NURBS)-based *boundary representation* (B-Rep) which was mainly developed in the 1970s and 1980s (see also Boehm [14], Boor [15], Cohen et al. [22], Gordon et al. [35], Patrikalakis [62], Riesenfeld et al. [71], and Sederberg et al. [81]). A *design-through-analysis workflow* therefore requires a geometry conversion called *meshing* which is a highly complex task in itself because the meshes derived from CAD models have to fulfill strict criteria (see also Abel Coll Sans [1], Hansen et al. [37], Knupp et al. [49], and Topping [89]). In addition, *meshing* leads to a geometry discretization error because usually a finite element mesh represents just an approximation of the CAD model. The biggest disadvantage, however, is that *meshing* often requires manual interactions and thus it can not be fully automatized.

To avoid the *meshing*, a few alternatives for the integration of design and analysis have been suggested in the past few years. One of these approaches is the *finite cell method* (FCM) (see Düster et al. [32], Parvizian et al. [61], Rank et al. [67], and Schillinger et al. [76, 77]). This approach has high potential, especially for three-dimensional structures, and it is being developed continuously by Rank and his co-workers. Other approaches include the Kantorovich method (see Kantorovich et al. [42]), implemented in *Scan*&*Solve*™(see Scan&Solve [74]), or *isogeometric analysis* (IGA) introduced by Hughes et al. [40].

IGA entails the use of the same basis functions for representing CAD geometries as well as for approximating solution fields. Usually, IGA is based on NURBS (see Cottrell et al. [24] and Hughes et al. [40]) because NURBS represent the standard for geometry description in contemporary CAD systems. NURBS have high continuity, allow for refinement on the identical geometry (see Boehm [14]) and raising the degree of the underlying polynomial functions (see also Cohen et al. [22], Piegl et al. [65], and Rogers [72]). NURBS basis functions show very good properties for analysis. In the past few years, many IGA researchers have demonstrated the advantages of these properties for analysis purposes, such as robustness and excellent approximation quality. IGA has been applied successfully in many fields of engineering such as structural mechanics (see Bauer et al. [5], Belytschko et al. [10], Benson et al. [13], Cottrell et al. [24], Cottrell et al. [25, 26], Hughes et al. [40], Nguyen-Thanh et al. [60], and Philipp et al. [63]), fluid mechanics (see Bazilevs et al. [8] and Hsu et al. [39]), fluid-structure interaction (see Bazilevs et al. [6, 9], Hsu et al. [38], and Zhang et al. [93]), contact mechanics (see Dimitri et al. [29], Lorenzis et al. [54], and Temizer et al. [88]), and optimization (see Kiendl et al. [45], Qian et al. [66], and Wall et al. [91]). The usage of the basis functions from design also for analysis simplifies greatly the integration of these two tasks. However, a general and consistent concept for creating direct and complete analysis models from CAD still is not available.

Recent publications have dealt with this shortcoming of IGA and suggested solutions pertaining to specific fields. Schillinger et al. [76] presented an *integrated design-through-analysis workflow* that is well suited for solving three-dimensional problems. Another approach for integrating design and analysis is the use of T-Splines (see Bazilevs et al. [7] and Sederberg et al. [82, 83]) or subdivision surfaces for geometric modeling and analysis (see Burkhart et al. [18], Cirak et al. [20], and Cirak et al. [21]). However, considering NURBS-based B-Rep models as standard for geometry description in CAD systems, the approaches mentioned above also require geometric transformation.

The goal of the proposed *CAD-integrated design-trough-analysis workflow,* named *analysis in computer aided design* (AiCAD), is to use the CAD standard geometry description (NURBS-based B-Rep models) for the entire workflow, i.e. for geometric and analysis modeling as well as the analysis itself (see Breitenberger et al. [17]). AiCAD allows for unification of the design and analysis models and can be fully automatized and entirely realized within existing CAD systems. AiCAD uses the newly developed *isogeometric B-Rep analysis* (IBRA) (see Breitenberger et al. [17]) for analyzing NURBS-based B-Rep models.

IBRA is a general and consequent extension of IGA, which in addition to the basis functions also uses the B-Rep description of the CAD model. This is expressed by the extension "B-Rep", a well known term in the CAD community (see Mäntylä [56] and Mortenson [57]). IBRA can directly use the design model, coming from a CAD system, for analysis, without meshing or re-parametrization. Hence, IBRA allows for the creation of a direct and complete analysis model from the CAD model in a consistent finite-element-like manner.

Thus far, AiCAD is realized within the integrated CAD/CAE/CAM program *Siemens NX* (see Siemens PLM Software [84]), within the CAD program *Rhinoceros* (see Rhinoceros [70]), and within the Pre- and Postprocessing software *GiD* (see GiD [34]). This demonstrates the general applicability of the presented concept.

This thesis provides guidelines for the realization of the AiCAD workflow in existing CAD programs and the implementation of IBRA for surface models in existing FE solvers. The focus lies on a modular disassembly of the necessary steps and a clear definition of the data interfaces. Aspects relevant to practice such as the application of arbitrarily located loads, local refinement, enforcing weak boundary conditions along trimming curves, dealing with non-matching multi-patches including gaps and overlaps, and the numerical integration of trimmed surfaces are discussed as well. For the numerical integration a new adaptive integration scheme is presented. In addition, the thesis highlights that AiCAD is very attractive and can be competitive to established workflows by solving industrial problems of shell structures.

The thesis is outlined as follows:

CHAPTER 2 explains in detail *NURBS-based B-Rep surface models* which are used throughout the entire AiCAD workflow and forms the basis for the proposed *isogeometric B-Rep analysis* (IBRA). Important aspects of NURBS-based B-Rep models relevant for structural shell analysis are discussed such as geometric continuities across edges including *non-watertightness*, geometrical refinement, trimming, trimming tolerances, and the topology of complex multi-patch models.

CHAPTER 3 reviews briefly the *finite element method* (FEM) for structural shell problems with a focus on using different geometry discretizations. In addition, a classification of the finite element techniques: classical FEA, IGA, and IBRA with respect to their geometry discretization is given.

CHAPTER 4 explains the *isogeometric B-Rep analysis* (IBRA), in particular the newly developed B-Rep elements, which can be used to handle discontinuous and trimmed geometries with gaps and overlaps for structural analysis in a finite-element-like manner.

CHAPTER 5 describes the systematic disassembly of CAD models into finite elements and defines data interfaces for an implementation of IBRA into existing FE solvers. In addition a new adaptive numerical integration scheme for trimmed surfaces is presented.

CHAPTER 6 describes the different steps resp. modules for the realization of the AiCAD workflow by combining existing CAD programs and FE-solvers.

CHAPTER 7 demonstrates the application of the proposed AiCAD workflow resp. IBRA to some academic and industrial shell problems. For these examples different aspects like trimming tolerances, *non-watertightness*, accuracy, and robustness are discussed.

CHAPTER 8 summarizes the thesis and gives an outlook to further possible research.

Where there is matter, there is geometry.

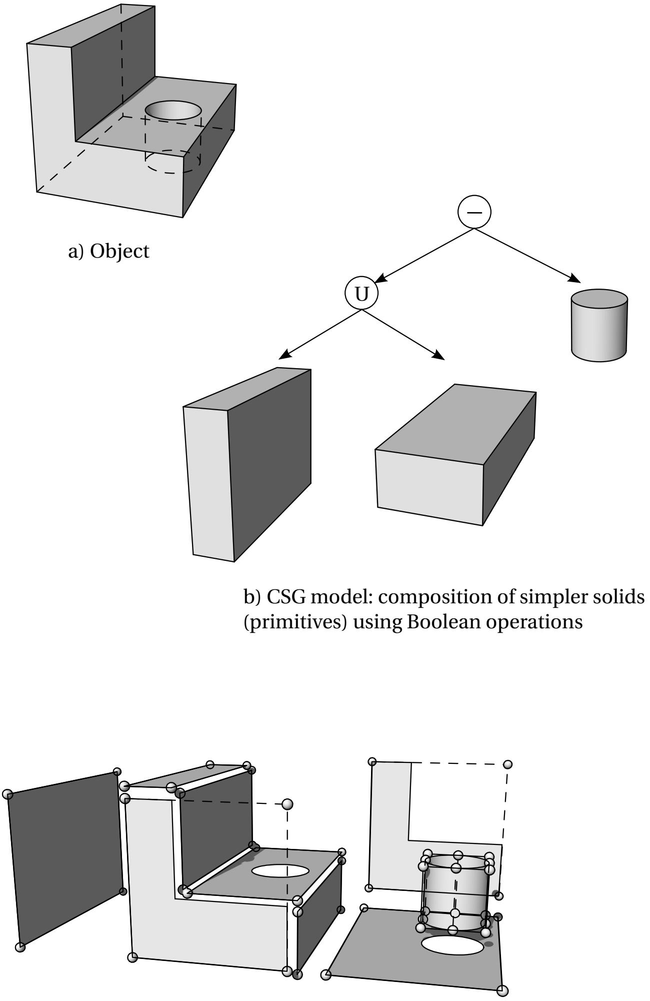
Johannes Kepler

Chapter 2

Geometric modeling and fundamentals

Design or geometric modeling is used for defining the shape and other geometric characteristics of an object. Since every day language is not useful to describe complex shapes, mathematical concepts and approaches are used for this purpose (see also Mortenson [57] and Rooney et al. [73]). Nowadays, geometric modeling is usually computer based and is performed using *computer-aided design* (CAD) systems. Geometric modeling with CAD systems is thus assumed throughout this thesis.

This chapter explains NURBS-based B-Rep models. They are the standard for geometry description in contemporary CAD systems for mechanical engineering and form the basis for the proposed CAD-integrated design-through-analysis workflow. Thus, important aspects relevant for structural analysis are discussed such as geometrical refinement, trimming, trimming tolerances, the topology of complex multi-patch models, and geometric continuities across edges including *non-watertightness*. Moreover, the chapter reviews differential geometry of trimmed surfaces.



a) Object

b) CSG model: composition of simpler solids (primitives) using Boolean operations

c) NURBS-based B-Rep model: composition of timmed NURBS surfaces

Figure 2.1: Different geometry representations of same object (after Choi [19])

2.1 Computer-Aided Design (CAD)

Computer-aided design (CAD) can be defined as usage of computer systems to assist in the creation, modification, analysis, or optimization of a design (see Narayan et al. [59]). Contemporary CAD systems provide many different modeling techniques to satisfy the increasingly demanding needs of the users, whereby each modeling technique has its own representation, like

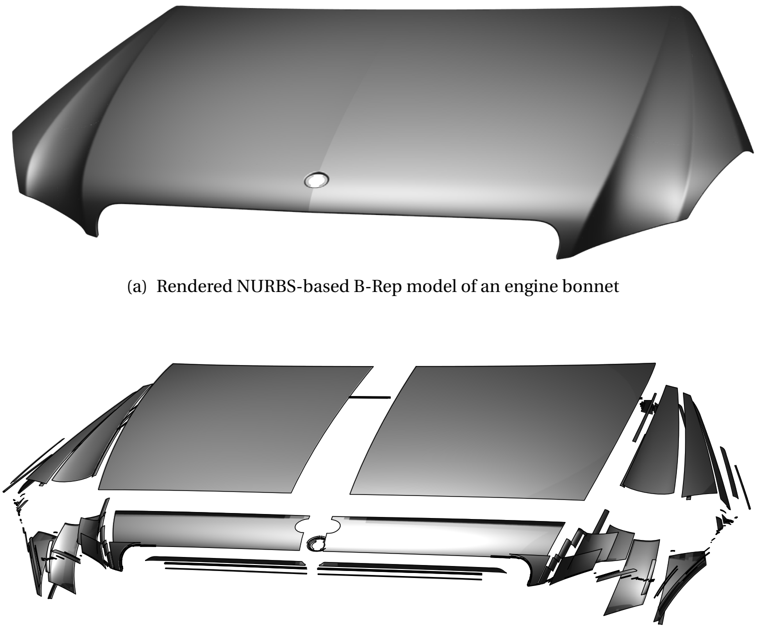
- cell decomposition,
- boundary representation (B-Rep),
- constructive solid geometry (CSG),
- ...

Owing to the limitations of each representation, a transformation to other representations is not always possible. Indeed, a CSG model, defined as a composition of simpler solids (primitives) using Boolean operations can easily be converted into a NURBS-based B-Rep model (see Figure 2.1), but in general it is not possible to derive a CSG model from a B-Rep model. This simple example already shows that NURBS-based B-Rep is very powerful. In addition, it is compatible with parametric design, which is indispensable for developing modern product designs. Thus NURBS-based B-Rep is included in the standard exchange format IGES (see Reed et al. [68]) and STEP (see ISO 10303 [41]) and is the industry standard in mechanical engineering especially for free-form geometries. Also commercial geometric modeling kernels like *Parasolid*[1] and *ACIS*[2], which currently dominate the kernel market, are also mainly based on NURBS-based B-Rep. For all the reasons above mentioned, NURBS-based B-Rep models (see Section 2.2) are assumed as output of geometric modeling.

The structure of such NURBS-based B-Rep resp. CAD models is explained in the following sections.

[1] owned by Siemens PLM Software [84]

[2] owned by Spatial Corporation [85] (part of Dassault Systemes [28])



(a) Rendered NURBS-based B-Rep model of an engine bonnet

(b) The CAD model consists of trimmed NURBS surfaces.

Figure 2.2: CAD model of an engine bonnet as a representative of a typical and complex industrial product described by a NURBS-based B-Rep model. (CAD model: by courtesy of Daimler AG [27])

2.2 NURBS-based B-Rep model

An example of a NURBS-based B-Rep model is shown in Figure 2.2, which illustrates an engine bonnet as a representative of a typical and complex industrial product. The geometry of such models is described by trimmed NURBS surfaces. This is shown exemplarily in Figure 2.1(c) and Figure 2.2(b).

In geometric modeling, B-Rep is a method for representing shapes using boundaries. The boundary representation of an object consist of two parts:

- shape (geometry), which defines the spatial position, the curvatures, etc.
- structure (topology), which allows to make links between geometrical entities

The following sections explain the geometrical description i.e. trimmed NURBS and the topology of NURBS-based B-Rep models with a focus on surface models.

2.2.1 Non-Uniform Rational B-Splines (NURBS)

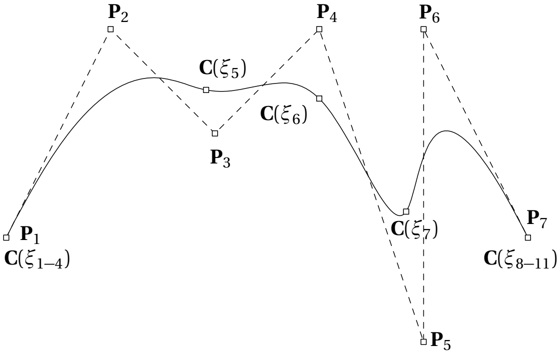
NURBS basis functions can be used for parametrizing curves in general as well as for surfaces and solids within tensor product construction. They can be used for modeling a large variety of shapes (see also Cohen et al. [23] and Rogers [72]). The term NURBS stands for *Non-Uniform Rational B-Splines* and indicates that NURBS are a generalization of B-Splines. Therefore, a short introduction to B-Splines is given first. For a detailed description of NURBS the reader is referred to Cohen et al. [23] and Piegl et al. [65].

B-Splines are non-interpolating, piecewise defined polynomial curves. They are defined by the following entities:

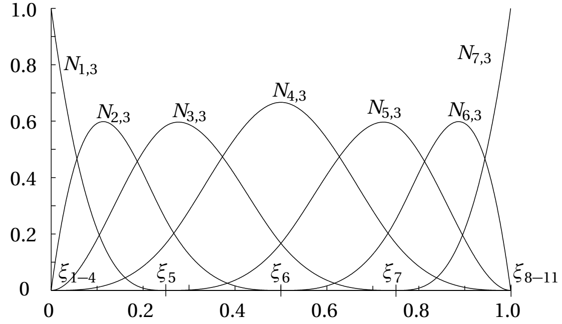
- set of control points, $\mathbf{P}_i,\ i = 1, ..., n$
- polynomial degree p
- knot vector $\Xi = [\xi_1, \xi_2, ...\xi_{\mathrm{n+p+1}}]$
- set of basis functions $N_i,\ i = 1, ..., n$

Knot vector

The knot vector Ξ is a set of parametric coordinates ξ_i arranged in ascending order. This set divides the B-Spline resp. NURBS into sections. If all knots are spaced equally, the knot vector is called *uniform* otherwise *non-uniform*. Knot values that appear more than once are called multiple knots. The intervals between two consecutive, distinct knots are called *non-zero knot spans*. In case the first and last entries in knot vectors have a multiplicity of $p+1$ they are called *clamped*. *Unclamped* knot vectors can be used for periodic B-Splines resp. NURBS (see Piegl et al. [65]), which are used to describe a closed shape, like ellipse and torus.



(a) B-Spline curve with its control point polygon (dashed lines) and knot positions on the curve.



(b) Corresponding basis functions defined in parameter space

Figure 2.3: Cubic B-Spline curve with the clamped knot vector $\Xi = [0,0,0,0,0.25,0.5,0.75,1,1,1,1]$ and its basis functions

Basis functions

B-Spline basis functions N_i are defined by the knot vector Ξ and the polynomial degree p. They can be computed using the Cox-deBoor recursion formula (see Boor [16], Cohen et al. [23], and Piegl et al. [65]). The computation starts with $p = 0$:

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

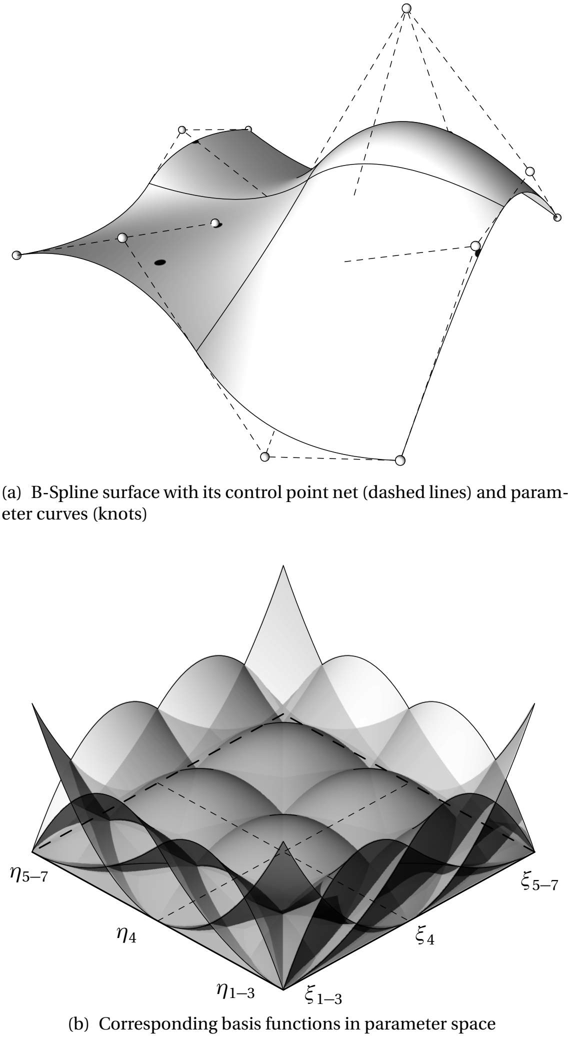
For $p \geqslant 1$ it is

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

The basis functions are C^∞ continuous inside a knot span and C^{p-1} continuous across single knots. At knots with multiplicity k the continuity of the basis functions is reduced to C^{p-k}. The following list contains some important properties of B-Spline basis functions:

- 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. $\sum_{i=1}^{n} N_{i,p}(\xi) = 1$
- non-negativity, i.e. $N_{i,p}(\xi) \geqslant 0$
- linear independence, i.e. $\sum_{i=1}^{n} \alpha_i N_{i,p}(\xi) = 0 \Leftrightarrow \alpha_i = 0,\ i = 1, 2, ..., n$

NURBS geometries have an additional weight for each control point, which can lead to rational basis functions. The weights are necessary for representing, for example, conical sections exactly. In case that all weights are equal to one, the basis functions reduce to simple piecewise polynomials. The formulas for NURBS basis functions are given in Equation (2.4) and Equation (2.6).



(a) B-Spline surface with its control point net (dashed lines) and parameter curves (knots)

(b) Corresponding basis functions in parameter space

Figure 2.4: Quadratic B-Spline surface with clamped knot vectors $\mathbf{\Xi} = \mathbf{H} = [0, 0, 0, 0.5, 1, 1, 1]$ and its basis functions

NURBS curves

B-Spline curves of degree p are computed using linear combinations of the control point coordinates and the corresponding basis functions. The formula for determining B-Spline curves is given by

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

NURBS curves have non-uniform knot vectors and the additional weight w_i for each control point $\mathbf{P}_i$ must be considered. The formula for NURBS curves with their basis functions $R_{i,p}$ is given as follows:

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} \frac{N_{i,p}(\xi) w_i \mathbf{P}_i}{\sum_{k=1}^{n} N_{k,p}(\xi) w_k} = \sum_{i=1}^{n} R_{i,p}(\xi)\mathbf{P}_i \tag{2.4}$$

An example of a cubic B-Spline with a clamped knot vector is shown in Figure 2.3. Owing to the clamped knot vector, the first and last control points ($\mathbf{P}_1$ and $\mathbf{P}_7$) are interpolated.

NURBS surfaces

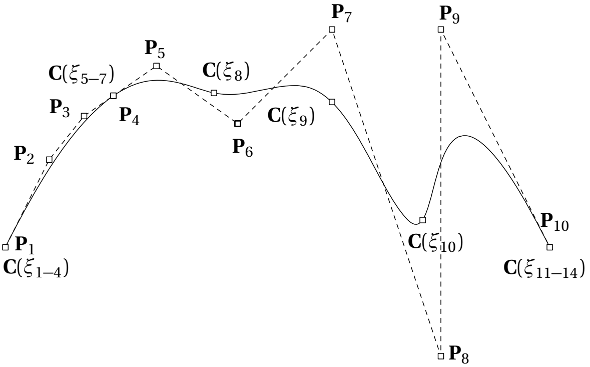
B-Spline surfaces are constructed as tensor products of B-Spline basis functions in the parametric dimensions ξ and η. They are defined by a net of $n \times m$ control points, two knot vectors Ξ and $\mathbf{H}$, two polynomial degrees p and q, and the corresponding basis functions $N_{i,p}(\xi)$ and $M_{j,q}(\eta)$. The formula for determining B-Spline surfaces $\mathbf{S}$ is

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi) M_{j,q}(\eta)\mathbf{P}_{ij}, \tag{2.5}$$

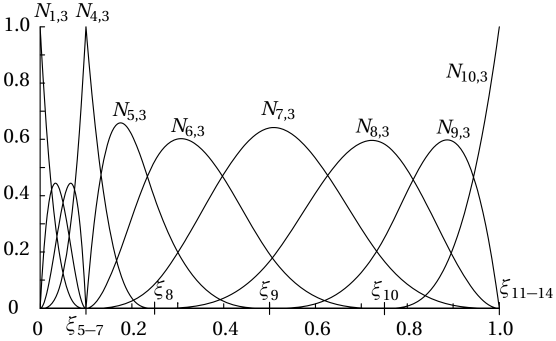
and NURBS surfaces are defined by

$$\begin{aligned}\mathbf{S}(\xi,\eta) &= \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{N_{i,p}(\xi) M_{j,q}(\eta) w_{ij}\mathbf{P}_{ij}}{\sum_{k=1}^{n}\sum_{l=1}^{m} N_{k,p}(\xi) M_{l,q}(\eta) w_{kl}} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} R_{ij}(\xi,\eta)\mathbf{P}_{ij}.\end{aligned} \tag{2.6}$$

An example of a quadratic B-Spline surface is illustrated in Figure 2.4, which has the clamped knot vectors $\Xi = \mathbf{H} = [0,0,0,0.5,1,1,1]$.

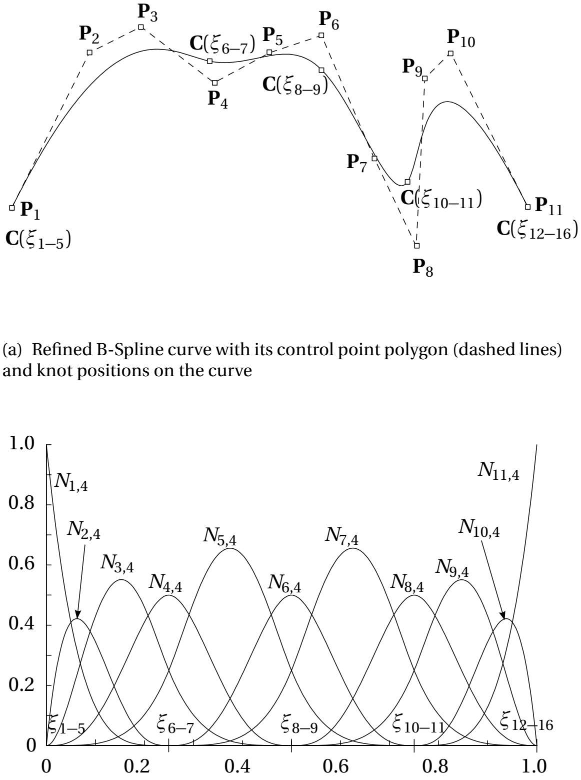


(a) Refined B-Spline curve with its control point polygon (dashed lines) and knot positions on the curve



(b) Corresponding refined basis functions

Figure 2.5: Knot insertion exemplified by B-Spline curve from Figure 2.3 with knot vector $\Xi = [0,0,0,0,0.25,0.5,0.75,1,1,1,1]$. After inserting three knots at $\xi = 0.1$ the knot vector becomes $\Xi = [0,0,0,0,0.1,0.1,0.1,0.25,0.5,0.75,1,1,1,1]$ and a C^0 continuity is introduced at the new knot. The refinement does not change the curve's shape.



(a) Refined B-Spline curve with its control point polygon (dashed lines) and knot positions on the curve

(b) Corresponding quartic basis functions

Figure 2.6: Degree elevation exemplified by B-Spline curve from Figure 2.3. The polynomial degree is increased by one to four. This results in the new knot vector $\boldsymbol{\Xi} = [0, 0, 0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1, 1, 1]$. The refinement does not change the curve's shape.

2.2.2 Geometry refinement

For the later mentioned purpose of analysis the geometry basis might need to be enriched to properly describe deformed geometries (see Section 3.5.1 and Hughes et al. [40]).

There are two ways of refining B-Spline and NURBS geometries, they are called *degree elevation* and *knot insertion* (see Boehm [14], Cohen et al. [22], and Piegl et al. [65]). Both refinement techniques increase the number of control points and thus impart the capability to represent a greater number of shapes. The refinement itself does not change the initial shape of the geometry.

The continuity of a curve can be decreased by generating knots with multiplicity greater than one (see Section 2.2.1). In case of multiplicity equal to $(p+1)$, the curve becomes interpolating at this knot and can be split easily into two curves. An example of a similar case is shown in Figure 2.5, which illustrates the *knot insertion* refinement of a B-Spline curve by introducing a C^0 continuity at $\xi = 0.1$. An example of *degree elevation* is shown in Figure 2.6.

For surfaces and solids refinement can be performed in the same manner and independently for each parameter direction.

2.2.3 Trimming

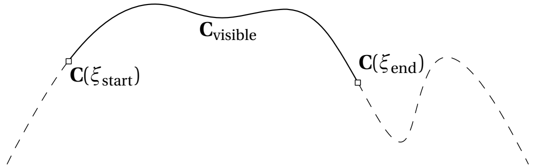
Trimming is the operation of modifying the visible part of geometries like surfaces or curves.

Trimmed NURBS curves

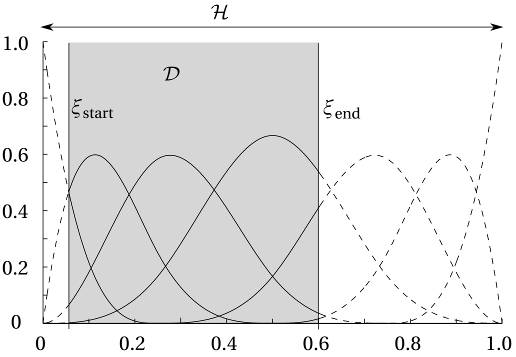
A trimmed NURBS curve is a partially visible curve defined by the *trimmed domain*

$$\mathcal{D} = \{\xi \in \mathcal{H} | \xi_{\text{start}} \leq \xi \leq \xi_{\text{end}}\}, \tag{2.7}$$

where $\mathcal{H}$ is the *parameter domain* of the curve $\mathbf{C}$ defined by the interval $[\xi_{p+1}, \xi_{n+1}]$ of the knot vector $\mathbf{\Xi}$ with p being the polynomial degree and n the number of control points. The additional parameters ξ_{start} and $\xi_{\text{end}} \in \mathcal{H}$


$\mathbf{C}_{\text{visible}}$
$\mathbf{C}(\xi_{\text{start}})$
$\mathbf{C}(\xi_{\text{end}})$

(a) Trimmed B-Spline curve with its start and end parameter on the curve. The dashed curve shows the remaining inactive curve segments.



(b) Corresponding basis functions defined on trimmed domain

Figure 2.7: Trimmed B-Spline curve from Figure 2.3 defined by the trimmed (active) domain $\mathcal{D}$ with $\xi_{\text{start}} = 0.055$ and $\xi_{\text{end}} = 0.6$. The trimming does not affect the knot vector $\boldsymbol{\Xi} = [0,0,0,0,0.25,0.5,0.75,1,1,1,1]$ and its basis functions.

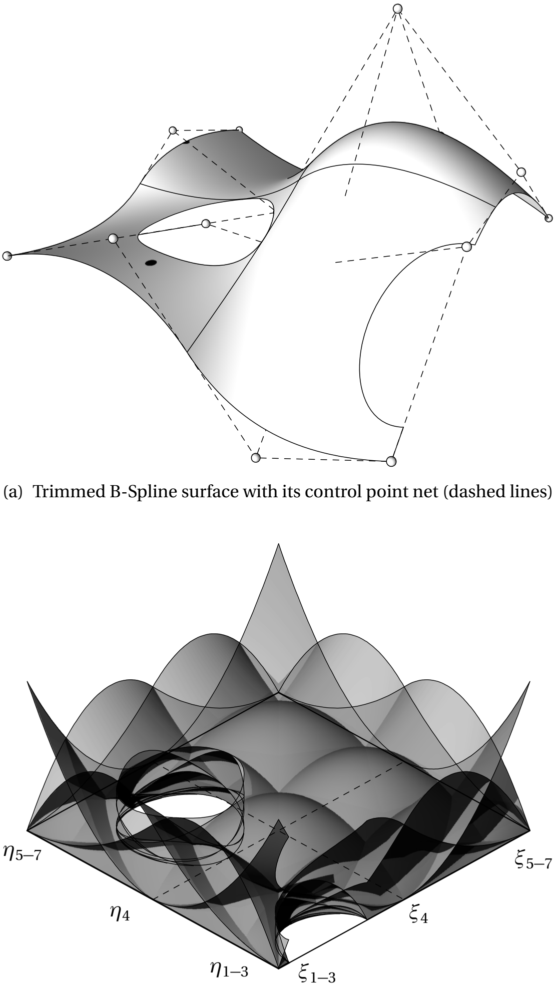
are parameter values which bound the visible part of the curve. Thus a trimmed curve is given by

$$\mathbf{C}_{\text{visible}} = \{\mathbf{C}(\xi) \mid \xi \in \mathcal{D}\} \tag{2.8}$$

An example of a trimmed curve is shown in Figure 2.7.

Trimmed NURBS surfaces

A trimmed NURBS surface is described by a NURBS surface and a set of M properly ordered boundary (trimming) curves $\tilde{\mathbf{C}}_k(\tilde{\xi})$ with $k = 1, ..., M$ lying within the *parameter domain* $\mathcal{H} = [\xi_{p+1}, \xi_{n+1}] \times [\eta_{q+1}, \eta_{m+1}]$ of the



(a) Trimmed B-Spline surface with its control point net (dashed lines)

(b) Corresponding basis functions defined on trimmed domain $\mathcal{D}$

Figure 2.8: Trimmed B-Spline surface and its basis functions on trimmed domain $\mathcal{D}$, which is defined by trimming curves.

surface (see also Piegl et al. [64]). Thus, a trimmed surface is a partially visible surface, defined by the *trimmed domain*

$$\mathcal{D} = \{(\xi,\eta) \in \mathcal{H} | \partial\mathcal{D} = \bigcup_{k=1}^{M} \tilde{\mathbf{C}}_k\}. \tag{2.9}$$

Here, $\partial\mathcal{D}$ describes the boundary of the closed *trimmed domain* $\mathcal{D}$.

In general, trimming curves $\tilde{\mathbf{C}}_k(\tilde{\xi})$ can be of any form, however, when dealing with NURBS entities, it is desirable to represent these with NURBS[3], too

$$\tilde{\mathbf{C}}_k(\tilde{\xi}) = \begin{bmatrix} \xi_k(\tilde{\xi}) \\ \eta_k(\tilde{\xi}) \end{bmatrix} = \sum_{i=1}^{n_k} R_{i,l}^{(k)}(\tilde{\xi})\,\tilde{\mathbf{P}}_i^k,\ k = 1,2,...,M. \tag{2.10}$$

Here, l is the polynomial degree, $\tilde{\xi}$ is the curve parameter, ξ_k and η_k are parameters of the surface representing the trimming curve $\tilde{\mathbf{C}}_k$and $\tilde{\mathbf{P}}_i^k$ are the control points of the trimming curve in the parameter space of the surface. The curves $\tilde{\mathbf{C}}_k(\tilde{\xi})$ are joined properly to form outer and inner loops. The outer loops are oriented *counter-clockwise*, whereas the inner loops are oriented *clockwise* (see also Section 2.3.3). The boundary of the surface is given by

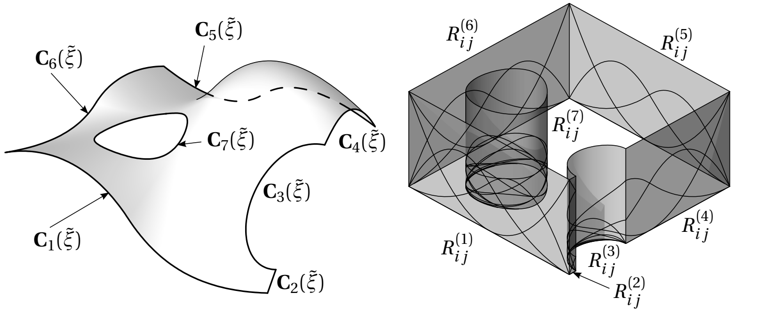
$$\partial\,\mathbf{S}_{\text{visible}} = \bigcup_{k=1}^{M} \partial\,\mathbf{S}_k, \tag{2.11}$$

where $\partial\,\mathbf{S}_k$ are implicitly defined curves, described by the basis functions of the surfaces as follows

$$\partial\,\mathbf{S}_k = \mathbf{S}(\tilde{\mathbf{C}}_k(\tilde{\xi})) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{ij}(\tilde{\mathbf{C}}_k(\tilde{\xi}))\mathbf{P}_{ij},\ \ k = 1,2,...,M. \tag{2.12}$$

Figure 2.9 shows exemplarily such basis functions for the trimmed surface in Figure 2.8(a). Since an explicit description of the boundary $\partial\,\mathbf{S}_k$ is needed for geometric modeling, the trimming curves $\tilde{\mathbf{C}}_k(\tilde{\xi})$ are mapped onto the surface as an explicit space curve $\mathbf{C}_k(\tilde{\xi})$ (see also Renner et al. [69]):

[3] within CAD systems in most cases just B-Splines are used


$\mathbf{C}_5(\tilde{\xi})$
$\mathbf{C}_6(\tilde{\xi})$
$\mathbf{C}_7(\tilde{\xi})$
$\mathbf{C}_4(\tilde{\xi})$
$\mathbf{C}_3(\tilde{\xi})$
$\mathbf{C}_1(\tilde{\xi})$
$\mathbf{C}_2(\tilde{\xi})$
$R_{ij}^{(6)}$
$R_{ij}^{(5)}$
$R_{ij}^{(7)}$
$R_{ij}^{(4)}$
$R_{ij}^{(1)}$
$R_{ij}^{(3)}$
$R_{ij}^{(2)}$

(a) Explicitly defined (approximated) space curves $\mathbf{C}_k$ of the trimming curves $\tilde{\mathbf{C}}_k$

(b) Implicitly defined basis functions on the boundaries

Figure 2.9: Description of the boundaries of the trimmed surface in Figure 2.8(a) with its implicitly defined basis functions and its explicitly defined (approximated) space curves.

Here, non-trivial mappings i.e. trimming curves which do not coincide with parameter lines, are usually approximated by using NURBS curves of third order (e.g. $\mathbf{C}_3$ and $\mathbf{C}_7$ in Figure 2.9(a)).

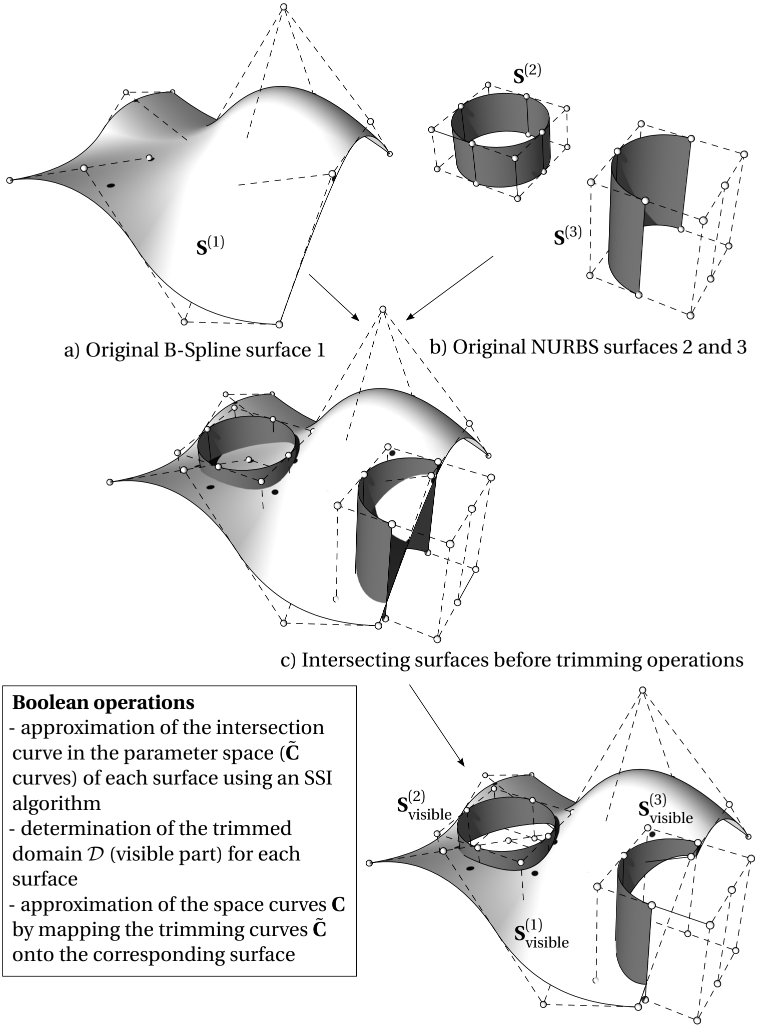
A trimmed surface is given by

$$\mathbf{S}_{\text{visible}} = \{\mathbf{S}(\xi,\eta) \mid (\xi,\eta) \in \mathcal{D}\}. \tag{2.13}$$

An example of a trimmed B-Spline surface with its basis functions, defined on a *trimmed domain* $\mathcal{D}$, is illustrated in Figure 2.8. The corresponding untrimmed B-Spline surface with its basis functions is shown in Figure 2.4.

Trimming operation

Trimming refers to the operation of creating trimmed geometries, e.g. by performing a Boolean or blending (see Choi [19]) operation. For curves, a point inversion algorithm (see Ma et al. [55]) can be used to determine the boundaries (ξ_{start} and ξ_{end}) of the *trimmed domain* $\mathcal{D}$. For surfaces, mathematically this results in a *surface-to-surface intersection* (SSI) problem (see Choi [19], Krishnan et al. [50], Patrikalakis [62], and Sederberg et al. [81]), where the trimming curves $\tilde{\mathbf{C}}$ of the intersection are determined in the parameter space.


$\mathbf{S}^{(2)}$
$\mathbf{S}^{(1)}$
$\mathbf{S}^{(3)}$
$\mathbf{S}^{(2)}_{\text{visible}}$
$\mathbf{S}^{(3)}_{\text{visible}}$
$\mathbf{S}^{(1)}_{\text{visible}}$

a) Original B-Spline surface 1

b) Original NURBS surfaces 2 and 3

c) Intersecting surfaces before trimming operations

Boolean operations
- approximation of the intersection curve in the parameter space ($\tilde{\mathbf{C}}$ curves) of each surface using an SSI algorithm
- determination of the trimmed domain $\mathcal{D}$ (visible part) for each surface
- approximation of the space curves $\mathbf{C}$ by mapping the trimming curves $\tilde{\mathbf{C}}$ onto the corresponding surface

d) Intersecting surfaces after trimming operations

Figure 2.10: Boolean operations including trimming of intersecting surfaces

Figure 2.10 shows an example of trimming operations with three intersecting surfaces before and after the execution of several Boolean operations. A trimming operation consists of the following steps:

- approximation of the intersection curve with the trimming curves $\tilde{\mathbf{C}}_k(\tilde{\xi})$ in the parameter space of each surface using an SSI algorithm (see Krishnan et al. [50], Patrikalakis [62], and Sederberg et al. [81])
- determination of the *trimmed domain* $\mathcal{D}$ for each surface
- approximation of the space curves $\mathbf{C}_k(\tilde{\xi})$ by mapping the trimming curves $\tilde{\mathbf{C}}_k(\tilde{\xi})$ onto the corresponding surface (see Renner et al. [69])

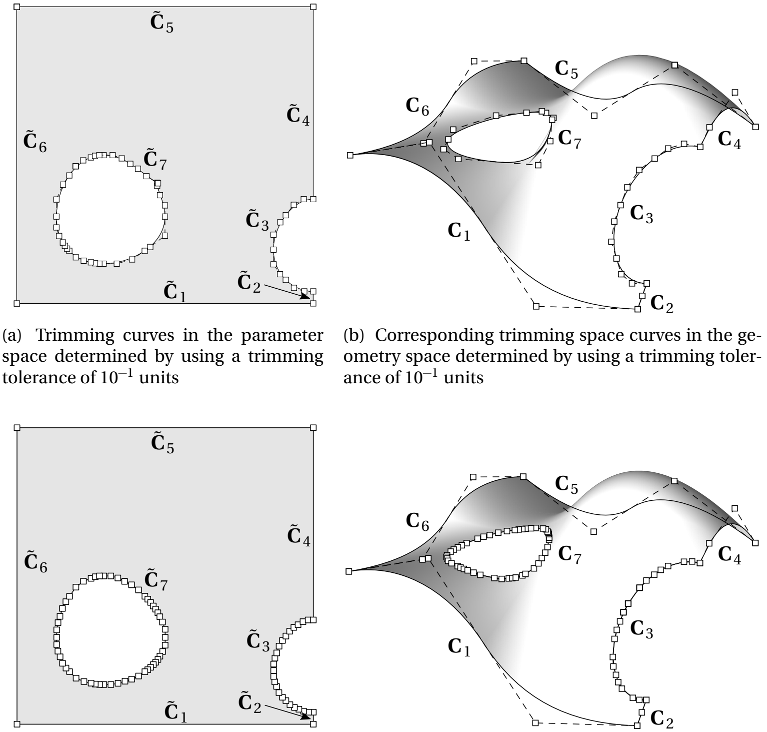
Note that the specification of the *trimmed domain* (visible part) depends on the type of Boolean operation (see also Figure 2.10).

For solving the SSI problem and mapping the trimming curves onto the surfaces most CAD systems use approximation algorithms, even if the mapping could be done exactly by using polynomials of high degrees (e.g. p=27) (see Renner et al. [69]). Indeed, the use of approximation techniques requires the specification of tolerances but it reduces the computational cost and increases the robustness against numerical instabilities (see Renner et al. [69]).

Tolerances

Figure 2.11 shows two different representations of the trimmed B-Spline surface $\mathbf{S}^{(1)}_{\text{visible}}$ in Figure 2.10(d). These have been created by using two different trimming tolerances. The trimmed surface shown in Figure 2.11(a) and Figure 2.11(b) was created in the CAD program *Rhinoceros 5* (see Rhinoceros [70]) by using a trimming tolerance of 10^{-1} units, while the surface described in Figure 2.11(c) and Figure 2.11(d) was created using a tolerance of 10^{-8} units. It can be seen that the more accurately trimmed surface requires a greater number of control points for the trimming curves (see $\tilde{\mathbf{C}}_3$ and $\tilde{\mathbf{C}}_7$) and space curves (see $\mathbf{C}_3$ and $\mathbf{C}_7$) to approximate the intersection curves.

An example of a very inaccurate approximated space curve is illustrated in Figure 2.12. The geometry has been created in the CAD program *Rhinoceros 5* by using a very high trimming tolerance of 1 unit. The obvious inaccuracy



(a) Trimming curves in the parameter space determined by using a trimming tolerance of 10^{-1} units

(b) Corresponding trimming space curves in the geometry space determined by using a trimming tolerance of 10^{-1} units

(c) Trimming curves in the parameter space determined by using a trimming tolerance of 10^{-8} units

(d) Corresponding trimming space curves in the geometry space determined by using a trimming tolerance of 10^{-8} units

Figure 2.11: Two different representations of trimmed B-Spline surface created in Figure 2.10 based on different trimming tolerances

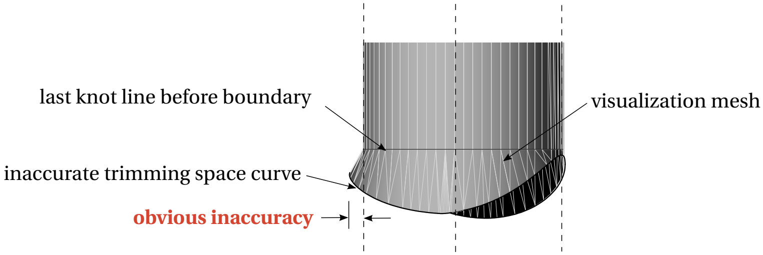

last knot line before boundary
visualization mesh
inaccurate trimming space curve
obvious inaccuracy

Figure 2.12: Very inaccurate approximated space curve of a trimming curve which is used as boundary for a cylindrical shape. The geometry has been created in the CAD program *Rhinoceros 5* by using a very high trimming tolerance of 1 unit. The dashed lines show the control point net. The obvious inaccuracy is highlighted.

of the trimming space curve is highlighted. Note that the visualization mesh which is used for representing trimmed surfaces within CAD systems uses the space curve as its boundary even though it lies not on the surface. This is mainly done to visually hide the mathematical *non-watertightness* (see also Section 2.4.1). The example demonstrates how CAD systems deal with inaccuracies caused by tolerances.

2.2.4 Topology of B-Rep

In addition to the geometric description of surfaces and curves, the *topology* of B-Rep needs to be discussed. The main topological entities are the following:

- *faces* (F)
- *edges* (E)
- *vertices* (V)

These topological entities are supported by their geometric forms: surfaces (**S**), curves (**C**), and points (**P**).

With B-Rep a three dimensional object is described by its closed *"skin"* between *"model"* and *"non-model"* (see Figure 2.1), whereas a dimension

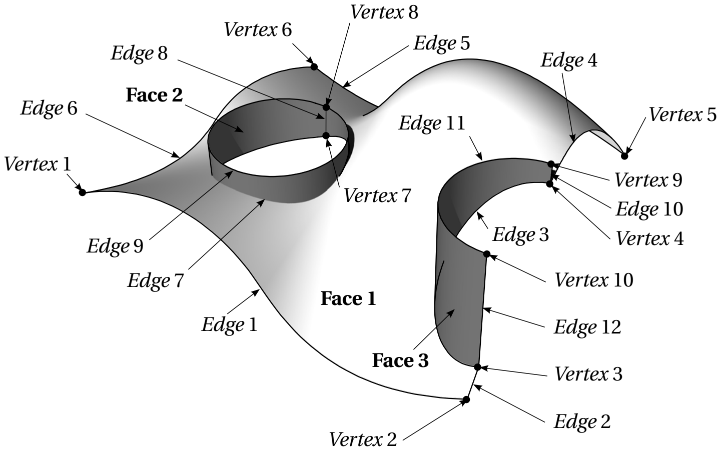


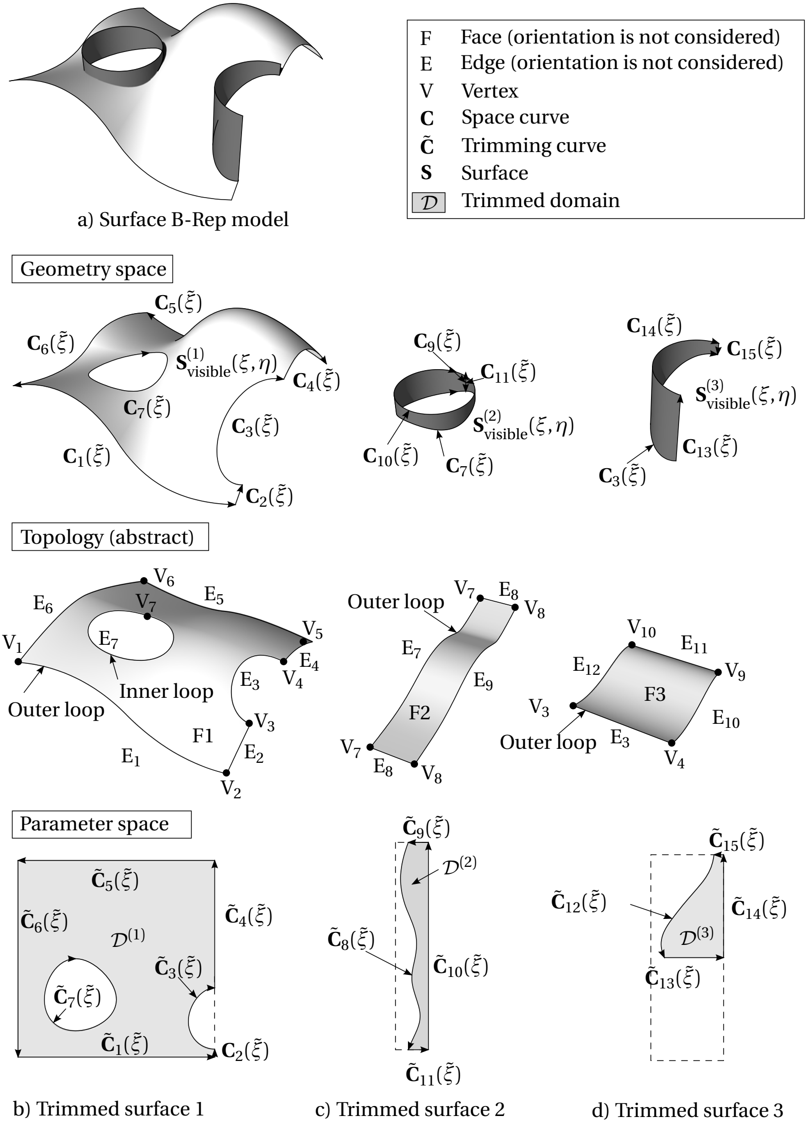
Figure 2.13: Simple surface model with its B-Rep topology entities (faces, edges and vertices). The object is represented by its *"skin"*.

reduced object (surface model) is described by the *"skin"* itself (see Figure 2.13). In both cases the skin is composed of a set of adjacent bounded surface elements i.e. trimmed NURBS surfaces (see Section 2.2.3)) called *faces*.

In the following, the explanation of topology is restricted to those parts which are needed for surface models. Figure 2.13 shows a simple surface model with its main topology entities and Figure 2.14 shows its decomposition into topological and geometrical entities.

Face

Each *face* consists of one trimmed surface and a set of properly joined *edges*, which define inner (*clockwise* oriented) and outer (*counter-clockwise* oriented) *loops*. The description of trimmed surfaces is described in Section 2.2.3 and is illustrated exemplarily in Figure 2.14 for the surface model in Figure 2.13.


F Face (orientation is not considered)
E Edge (orientation is not considered)
V Vertex
$\mathbf{C}$ Space curve
$\tilde{\mathbf{C}}$ Trimming curve
$\mathbf{S}$ Surface
$\mathcal{D}$ Trimmed domain
Geometry space
$\mathbf{C}_5(\tilde{\xi})$
$\mathbf{C}_6(\tilde{\xi})$
$\mathbf{S}^{(1)}_{\text{visible}}(\xi,\eta)$
$\mathbf{C}_4(\tilde{\xi})$
$\mathbf{C}_7(\tilde{\xi})$
$\mathbf{C}_3(\tilde{\xi})$
$\mathbf{C}_1(\tilde{\xi})$
$\mathbf{C}_2(\tilde{\xi})$
$\mathbf{C}_9(\tilde{\xi})$
$\mathbf{C}_{11}(\tilde{\xi})$
$\mathbf{S}^{(2)}_{\text{visible}}(\xi,\eta)$
$\mathbf{C}_{10}(\tilde{\xi})$
$\mathbf{C}_7(\tilde{\xi})$
$\mathbf{C}_{14}(\tilde{\xi})$
$\mathbf{C}_{15}(\tilde{\xi})$
$\mathbf{S}^{(3)}_{\text{visible}}(\xi,\eta)$
$\mathbf{C}_{13}(\tilde{\xi})$
$\mathbf{C}_3(\tilde{\xi})$
Topology (abstract)
V_6
E_6
V_7
E_5
V_5
V_1
E_7
E_4
V_4
E_3
Inner loop
Outer loop
V_3
F1
E_2
E_1
V_2
Outer loop
V_7
E_8
V_8
E_7
E_9
F2
V_7
E_8
V_8
V_{10}
E_{11}
E_{12}
V_9
F3
V_3
E_{10}
Outer loop
E_3
V_4
Parameter space
$\tilde{\mathbf{C}}_5(\tilde{\xi})$
$\tilde{\mathbf{C}}_6(\tilde{\xi})$
$\tilde{\mathbf{C}}_4(\tilde{\xi})$
$\mathcal{D}^{(1)}$
$\tilde{\mathbf{C}}_3(\tilde{\xi})$
$\tilde{\mathbf{C}}_7(\tilde{\xi})$
$\tilde{\mathbf{C}}_1(\tilde{\xi})$
$\tilde{\mathbf{C}}_2(\tilde{\xi})$
$\tilde{\mathbf{C}}_9(\tilde{\xi})$
$\mathcal{D}^{(2)}$
$\tilde{\mathbf{C}}_8(\tilde{\xi})$
$\tilde{\mathbf{C}}_{10}(\tilde{\xi})$
$\tilde{\mathbf{C}}_{11}(\tilde{\xi})$
$\tilde{\mathbf{C}}_{15}(\tilde{\xi})$
$\tilde{\mathbf{C}}_{12}(\tilde{\xi})$
$\tilde{\mathbf{C}}_{14}(\tilde{\xi})$
$\mathcal{D}^{(3)}$
$\tilde{\mathbf{C}}_{13}(\tilde{\xi})$

a) Surface B-Rep model

b) Trimmed surface 1

c) Trimmed surface 2

d) Trimmed surface 3

Figure 2.14: NURBS-based surface B-Rep model

Edge

Edges are topological entities of

- common parts of trimming curves, which bound surfaces (*faces*) on either side of an *edge* (see *Edge 3* and *Edge 7* in Figure 2.13), or
- free trimming curves (see e.g. *Edge 1* in Figure 2.13).

Within CAD systems an *edge* is described by one space curve bounded by two *vertices* given in spatial coordinates. The corresponding trimming curves for the adjacent surfaces within their parameters spaces are provided by the CAD system. The parameters $\tilde{\xi}_{\text{start}}$ and $\tilde{\xi}_{\text{end}}$ of the trimmed domain $\tilde{\mathcal{D}}$ of the trimming curve within the parameter space of the surface (see also Section 2.2.3) which defines the *edge* need to be computed by a point inversion algorithm (see Ma et al. [55]) using the spatial coordinates of the *vertices* which bound the *edge*.

Vertex

Vertices given in spatial coordinates are the topological entities of points where several *edges* meet and thus they define the boundaries of edges.

Depending on the purpose, the *face-edge-vertex* data model can also be augmented by additional elements such as *shells* and/or *loops* (see *topology* in Figure 2.14). For more information about B-Rep the reader is referred to Mäntylä [56] and Stroud [86].

2.3 Differential geometry of trimmed surfaces

This section reviews all basics of differential geometry of trimmed surfaces which are used within this thesis. Remember that within this thesis the Einstein summation convention as well as the convention that Latin indices like *i,j,k,l* take letters $\{1,2,3\}$ and Greek letters like $\alpha,\beta,\gamma,\delta$ take the values $\{1,2\}$ is used. In addition, derivatives w.r.t. to a quantity i are abbreviated by $(\cdot)_{,i}$.

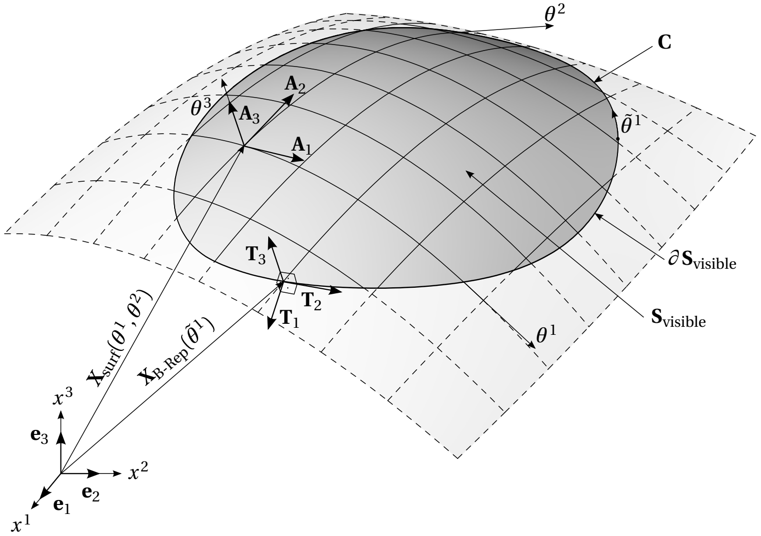


Figure 2.15: Geometry description of a trimmed surface

2.3.1 Differential geometry of surfaces

A spatial point $\mathbf{X}_{\text{surf}}(\theta^1,\theta^2) \in \mathbf{S}_{\text{visible}}$ of a trimmed NURBS surface can be computed with Equation (2.6) using parameters $\theta^\alpha \in \mathcal{D}$. The covariant basis $\mathbf{A}_\alpha$ with contravariant coordinates θ^α are given by

$$\mathbf{A}_\alpha = \frac{\partial \mathbf{X}_{\text{surf}}}{\partial \theta^\alpha}, \tag{2.14}$$

whereas the contravariant basis $\mathbf{A}^\alpha$ with their covariant coordinates θ_i can be computed as

$$\mathbf{A}^\alpha = \frac{\partial \mathbf{X}_{\text{surf}}}{\partial \theta_\alpha}. \tag{2.15}$$

Since a trimmed surface is a parametrically two-dimensional geometry and each point on the surface is described by two curvilinear coordinates θ^α, the two covariant base vectors $\mathbf{A}_\alpha$ can be obtained with Equation (2.14).

The third covariant base vector $\mathbf{A}_3$ is defined as a normalized vector, orthogonal to $\mathbf{A}^\alpha$ as

$$\mathbf{A}_3 = \frac{\mathbf{A}_1 \times \mathbf{A}_2}{\left\|\mathbf{A}_1 \times \mathbf{A}_2\right\|_2} \tag{2.16}$$

with the property

$$\mathbf{A}_3 = \mathbf{A}^3. \tag{2.17}$$

The metric tensor $\mathbf{A}$ of a surface can be expressed in the covariant and contravariant basis as

$$\mathbf{A} = A^{\alpha\beta}\mathbf{A}_\alpha \otimes \mathbf{A}_\beta = A_{\alpha\beta}\mathbf{A}^\alpha \otimes \mathbf{A}^\beta. \tag{2.18}$$

Here, covariant metric coefficients $A_{\alpha\beta}$ can be obtained with the so called *first fundamental form of surfaces* by the scalar product of covariant base vectors (see also Klingbeil [48]):

$$A_{\alpha\beta} = \mathbf{A}_\alpha \cdot \mathbf{A}_\beta \tag{2.19}$$

and the contravariant metric coefficients $A^{\alpha\beta}$ can be computed by inverting the covariant coefficient matrix

$$[A^{\alpha\beta}] = [A_{\alpha\beta}]^{-1}. \tag{2.20}$$

The metric coefficients can be used to switch between the two different bases

$$\mathbf{A}^\alpha = A^{\alpha\beta}\mathbf{A}_\beta \;\text{ resp. }\; \mathbf{A}_\alpha = A_{\alpha\beta}\mathbf{A}^\beta. \tag{2.21}$$

The explicit description of contravariant base vectors using covariant base vectors (see also Equation (2.17)) is given by

$$\mathbf{A}^1 = \frac{1}{\det[A_{\alpha\beta}]}(A_{22}\mathbf{A}_1 - A_{12}\mathbf{A}_2) \tag{2.22}$$

$$\mathbf{A}^2 = \frac{1}{\det[A_{\alpha\beta}]}(-A_{21}\mathbf{A}_1 + A_{11}\mathbf{A}_2) \tag{2.23}$$

with $\det[A_{\alpha\beta}]$ being the determinate of the covariant metric, which can be computed as follows:

$$\det[A_{\alpha\beta}] = A_{11}A_{22} - A_{12}A_{21} \tag{2.24}$$

Curvature tensor

The *second fundamental form of surfaces* describes the curvature properties of a surface. The covariant tensor coefficients of the curvature (see also Basar et al. [4] and Klingbeil [48]) are defined as :

$$B_{\alpha\beta} = -\mathbf{A}_{\alpha} \cdot \mathbf{A}_{3,\beta} = -\mathbf{A}_{\beta} \cdot \mathbf{A}_{3,\alpha} = \mathbf{A}_{\alpha,\beta} \cdot \mathbf{A}_3 \tag{2.25}$$

2.3.2 Differential geometry of surface boundaries

A spatial point $\mathbf{X}_{\text{B-Rep}}(\tilde{\theta}^1) \in \partial\,\mathbf{S}_{\text{visible}}$ of a surface boundary can be computed with Equation (2.6) using parameters $\theta^{\alpha}(\tilde{\theta}^1) \in \partial\mathcal{D}$. The covariant basis $\mathbf{A}_i$ (see Equation (2.14)) is not useful since their orientation is independent of the boundary description. Thus a new orthonormal local coordinate system $\mathbf{T}_i$ aligned with the boundary curve is introduced

$$\mathbf{T}_i = \frac{\breve{\mathbf{T}}_i}{\overline{T}_i} \tag{2.26}$$

with $\breve{\mathbf{T}}_i$ being the not normalized basis defined by

$$\begin{aligned} \breve{\mathbf{T}}_2 &= \frac{\partial \mathbf{X}_{\text{surf}}}{\partial \tilde{\theta}^1} = \mathbf{A}_1 \frac{\partial \theta^1}{\partial \tilde{\theta}^1} + \mathbf{A}_2 \frac{\partial \theta^2}{\partial \tilde{\theta}^1}, \\ \breve{\mathbf{T}}_3 &= \mathbf{A}_1 \times \mathbf{A}_2, \\ \breve{\mathbf{T}}_1 &= \tilde{\mathbf{T}}_2 \times \tilde{\mathbf{T}}_3, \end{aligned} \tag{2.27}$$

and $\overline{T}_i$ being the length of the vectors $\breve{\mathbf{T}}_i$

$$\overline{T}_i = \left\| \breve{\mathbf{T}}_i \right\|_2 . \tag{2.28}$$

Here, the base vector $\breve{\mathbf{T}}_2$ represents the tangent vector of the space curve, $\breve{\mathbf{T}}_3$ is the not-normalized vector of the surface normal and $\breve{\mathbf{T}}_1$ is the vector perpendicular to $\breve{\mathbf{T}}_1$ and $\breve{\mathbf{T}}_3$ pointing outwards from the surface (see Section 2.2.3 for the correct curve direction).

For the sake of implementation, it is more elegant to rewrite the vector $\breve{\mathbf{T}}_1$ with the pseudo parameter $\tilde{\theta}^2$ by using the *triple product expansion* as

$$\breve{\mathbf{T}}_1 = \mathbf{A}_1 \frac{\partial \theta^1}{\partial \tilde{\theta}^2} + \mathbf{A}_2 \frac{\partial \theta^2}{\partial \tilde{\theta}^2} \tag{2.29}$$

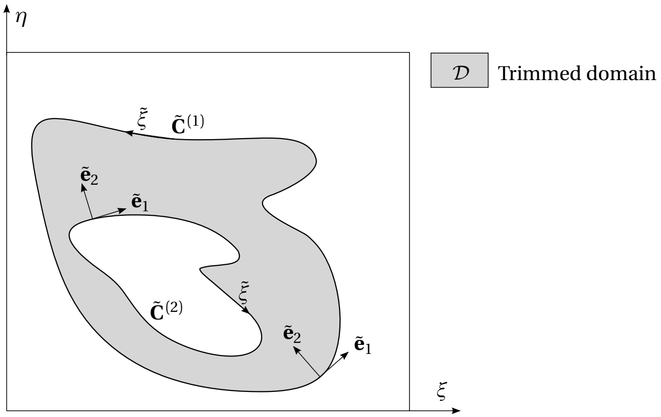


Figure 2.16: Differential geometry of trimming curves within parameter space. Considering correctly oriented curves the vector $\tilde{\mathbf{e}}_2$ points always towards the visible domain i.e. the latter is always on the left hand side of the trimming curves.

where the derivatives $\frac{\partial \theta^\alpha}{\partial \tilde{\theta}^2}$ are determined by

$$\frac{\partial \theta^1}{\partial \tilde{\theta}^2} = \check{\mathbf{T}}_2 \cdot \mathbf{A}_2 \quad \text{and} \quad \frac{\partial \theta^2}{\partial \tilde{\theta}^2} = -\check{\mathbf{T}}_2 \cdot \mathbf{A}_1. \tag{2.30}$$

Curvature tensor

The curvature tensor at the boundary can be computed as described in Section 2.3.1, but for the determination of curvature continuities (see Section 2.4.3) across edges it is advantageous to express the curvature within the above defined local coordinate system $\mathbf{T}_i$ (see Equation (2.26)) with the coefficients $c_{\gamma\delta}$.

$$c_{\gamma\delta} = b_{\alpha\beta}(\mathbf{T}_\gamma \cdot \mathbf{A}^\alpha)(\mathbf{A}^\beta \cdot \mathbf{T}_\delta) \tag{2.31}$$

2.3.3 Differential geometry of trimming curves

The surface parameters ξ and $\eta \in \partial\mathcal{D}$ of a spatial point $\mathbf{X}_{\text{B-Rep}} \in \partial\,\mathbf{S}_{\text{visible}}$ can be computed with Equation (2.10) using the corresponding curve parameter $\tilde{\xi}$. The tangent of the trimming curve $\tilde{\mathbf{C}}_{,\tilde{\xi}}$ within parameter space can be computed as

$$\tilde{\mathbf{T}} = \tilde{\mathbf{C}}_{,\tilde{\xi}} = \begin{bmatrix} \frac{\partial \theta^1}{\partial \tilde{\theta}^1} \\ \frac{\partial \theta^2}{\partial \tilde{\theta}^1} \end{bmatrix} \tag{2.32}$$

where $\tilde{\mathbf{C}}_{,\tilde{\xi}}$ is the derivative of the trimming curve and the normalized tangent is defined by

$$\tilde{\mathbf{e}}_1 = \frac{\tilde{\mathbf{T}}}{\left\|\tilde{\mathbf{T}}\right\|_2}. \tag{2.33}$$

Assuming $\tilde{\mathbf{e}}_1$ in $\mathbb{R}^3$ the vector $\tilde{\mathbf{e}}_2 \in \mathbb{R}^3$ can be computed as

$$\tilde{\mathbf{e}}_2 = \tilde{\mathbf{e}}_3 \times \tilde{\mathbf{e}}_1, \tag{2.34}$$

with $\tilde{\mathbf{e}}_3 \in \mathbb{R}^3$ given by

$$\tilde{\mathbf{e}}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{2.35}$$

Figure 2.16 illustrates an example of a trimmed domain with the local orthonormal coordinate system $\tilde{\mathbf{e}}_i$.

2.4 Geometric continuities

This section deals with different orders of geometric continuity G^k with $k \in \mathbb{N}$ indicating the order of continuity, across a common *edge* E (see Section 2.2.4) of two adjacent *faces*, i.e. trimmed surfaces (see Section 2.2.3) defined by $\mathbf{S}^{(\alpha)}_{\text{visible}}$. The edge E as a topological entity links two boundary subsets

$$\partial\,\mathbf{S}_{\mathrm{E}} \subset \partial\,\mathbf{S}^{(\alpha)}_{\text{visible}}, \tag{2.36}$$

such that within the B-Rep description (see Section 2.2.4) they belong together.

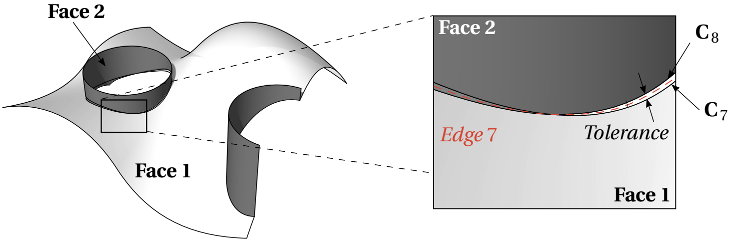


Figure 2.17: *Non-watertightness* of NURBS-based B-Rep models. Since trimming (space) curves only approximate the exact intersection curve a gap within the specified tolerance can occur. Here, the trimming is done with a very unusual high tolerance of 0.1 units. For illustrating the gap each face uses its own space curve for representing the common edge. Note that CAD systems use just one curve for both faces (see also Figure 2.12) such that the *non-watertightness* can not be seen (see also Figure 2.18).

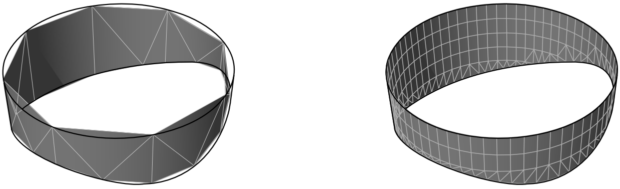
2.4.1 Positional continuity – $\mathbf{G}^0$ (Watertightness)

Positional continuity (*watertightness*) on the set of points of an edge $\partial\,\mathbf{S}_\mathrm{E}$ is given if the subsets $\partial\,\mathbf{S}_\mathrm{E}^{(\alpha)}$ coincide geometrically:

$$\partial\,\mathbf{S}_\mathrm{E}^{(1)} = \partial\,\mathbf{S}_\mathrm{E}^{(2)} = \partial\,\mathbf{S}_\mathrm{E} \tag{2.37}$$

In CAD systems, the intersection curve of two intersecting surfaces is defined by one trimming curve on each surface and represented by the corresponding space curves (see Section 2.2.3). Since all these curves are usually approximated, gaps between boundaries, but representing the same edge, can occur. Such gaps pertain to *non-watertightness* (see also Sederberg et al. [80]). The maximum tolerance for such gaps depends strongly on the application. In contemporary CAD systems, the user can usually specify the desired (relative and/or absolute) tolerance. In *Rhinoceros* (see Rhinoceros [70]) for instance the default value for large models is 0.01 units and that for small models is 0.001 units.

Figure 2.17 illustrates the *non-watertightness* between two trimmed surfaces (face 1 and 2) of the example in Figure 2.10 and Figure 2.14. It can be seen that the space curve $\mathbf{C}_7$ of the surface $\mathbf{S}_\mathrm{visible}^{(1)}$ as well as the corre-



(a) Coarse visualization mesh

(b) Fine visualization mesh

Figure 2.18: Comparison of two different visualization meshes of the same trimmed surface (see Figure 2.8). The space curves are independent of the fineness of the visualization mesh and thus in both figures the same.

sponding space curve $\mathbf{C}_8$ of the surface $\mathbf{S}^{(2)}_{\text{visible}}$ do not match. For a better illustration of the gap the trimming in Figure 2.17 is done with a very unusually high tolerance of 0.1 units and each face uses its own space curve for representing the common edge. Note that CAD systems usually use just one curve for both faces (see also Figure 2.12) such that the *non-watertightness* can not be seen (see also Figure 2.17).

In practice such gaps are not problematic as long as they are within the required tolerance which can be specified within the CAD system.

Beside the above mentioned *non-watertightness* caused by approximating the intersection curves, a *visual non-watertightness* caused by representing trimmed surfaces by visualization meshes exists as well. In case the mesh is too coarse like in Figure 2.18(a), one might see gaps between surfaces. Such gaps can easily be reduced by using a finer visualization mesh (see Figure 2.18(b)) and are not related to the quality of the CAD model itself.

2.4.2 Tangential continuity – $\mathbf{G}^1$

Tangential continuity on the edge E is given if at each point $\mathbf{X}_{\text{B-Rep}} \in \partial\, \mathbf{S}_{\text{E}}$ with

$$\mathbf{X}_{\text{B-Rep}} = \mathbf{X}^{(1)}_{\text{B-Rep}} = \mathbf{X}^{(2)}_{\text{B-Rep}} \tag{2.38}$$

the vectors $\mathbf{T}_1^{(\alpha)}$ of the locally defined coordinate systems $\mathbf{T}_i^{(\alpha)}$ (see Equation (2.26)) aligned to the edge (different orientation of $\mathbf{T}_2^{(\alpha)}$ possible) point in opposition direction.

$$\mathbf{T}_1^{(1)} = -\mathbf{T}_1^{(2)} \tag{2.39}$$

Kinks

In the case, only the vectors $\mathbf{T}_2^{(1)}$ are collinear, a kink exists with an angle ϕ around $\mathbf{T}_2^{(\alpha)}$ at the corresponding position

$$\phi = \pm\arccos(\mathbf{T}_1^{(1)} \cdot \mathbf{T}_1^{(2)}) \tag{2.40}$$

2.4.3 Curvature continuity – G^2

Curvature continuity on the edge E is given if at each point $\mathbf{X}_{\text{B-Rep}} \in \partial\,\mathbf{S}_{\text{E}}$ besides tangential continuity also the curvature tensor, with its coefficients $c_{\alpha\beta}^{(\alpha)}$ (see Equation (2.31)) coincides.

$$\begin{aligned} c_{\alpha\beta}^{(1)} &= c_{\alpha\beta}^{(2)} \text{ with } \mathbf{T}_3^{(1)} = -\mathbf{T}_3^{(2)} \\ c_{\alpha\beta}^{(1)} &= -c_{\alpha\beta}^{(2)} \text{ with } \mathbf{T}_3^{(1)} = \mathbf{T}_3^{(2)} \end{aligned} \tag{2.41}$$

Depending on the geometric continuity, CAD models are classified to be of standard or high quality. The latter require a high geometrical continuity with a low tolerance level. They are used for aesthetic purposes, such as car bodies and consumer product outer forms. The higher geometrical continuity is mainly required for a better light reflection. Figure 2.19 shows the light and *zebra pattern* reflection of an edge with different geometrical continuities, representing almost the same shape. One can see that the higher the continuity the smoother the transition of the reflection becomes. In Figure 2.19 the "zebra pattern" reflection, which gives visual feedback for geometric continuities, illustrates that for G^0 the pattern is not continuous, for G^1 it has a kink and for G^2 and higher it is smooth. In addition, Figure 2.19 shows the curvature distribution perpendicular to the edge.

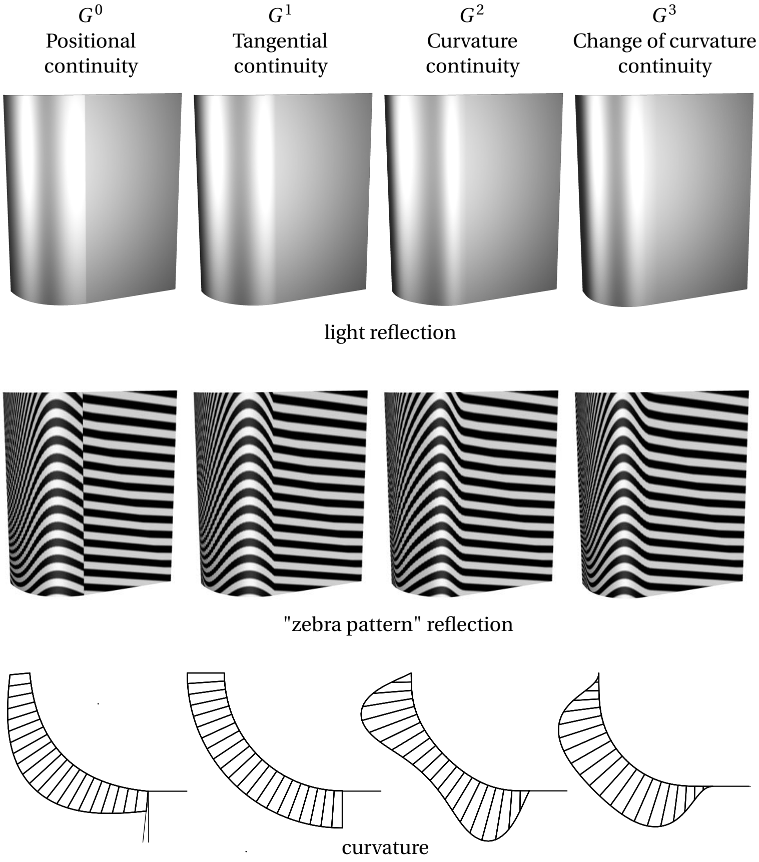


Figure 2.19: Comparison of different geometrical continuities along an edge, representing the curvature perpendicular to the edge and the reflection of light as a "zebra pattern".

Summary and conclusion of Chapter 2

The chapter explains NURBS-based B-Rep models which are used in industry for describing complex geometrical shapes (see Figure 2.2). The chapter explains the aspects which need to be considered for IBRA. These are the following:

- geometrical refinement
- description of trimmed surfaces
- trimming tolerances
- topology of complex multi-patch geometries
- geometric continuities across edges

Moreover, the chapter briefly explains the differential geometry of trimmed surfaces as a basis for the following chapters.

Nature always tends to act in
the simplest way.

Daniel Bernoulli

CHAPTER 3

Finite element method for structural shell analysis

This chapter briefly reviews important aspects of the *finite element method* (FEM) for structural shell analysis in order to differentiate

- classical *finite element analysis* (FEA) (see Section 3.4),
- *isogeometric analysis* (IGA) (see Section 3.5), and
- *isogeometric B-Rep analysis* (IBRA) (see Chapter 4),

w.r.t. their geometrical discretizations used for analysis. The chapter explains the different element definitions used for classical FEA and IGA as well as their shape functions, and data structures.

In addition, the different refinement strategies (h, p and k-refinement) for IGA are briefly reviewed.

3.1 Shell structures

Shells are thin-walled, curved structures like car bodies, roof shells or cooling towers. For the analysis of such structures, shell formulations are attractive because they concentrate on the mechanically relevant effects and thus the computational effort can be reduced significantly. For solving shell problems two main theories are widely used: the Reissner-Mindlin (RM) and the Kirchhoff-Love (KL) theory.

3.2 *Kirchhoff-Love* shell theory

Within this thesis the *Kirchhoff-Love* (KL) shell theory is used as basis for solving structural shell problems. The assumptions of this theory are the following:

- the director remains straight and perpendicular to the midsurface during deformation
- no transverse shear deformation is taken into account
- the thickness t remains constant
- the ratio $\frac{R}{t}$ of the characteristic radius R and the thickness t is larger than 20

The geometrical description and the strong form of the static equilibrium for KL shell problems is given in the sequel, whereas the weak form of the static equilibrium is described in Section 3.3.2 and Section 4.2.

3.2.1 Geometrical description

The body resp. physical domain of a KL shell can be described by the position vector **X** resp. **x** in the reference resp. in the current configuration as follows:

$$\begin{aligned}\mathbf{X}(\theta^1,\theta^2,\theta^3) &= \mathbf{X}_{\mathrm{surf}}(\theta^1,\theta^2)+\theta^3\mathbf{A}_3,\\ \mathbf{x}(\theta^1,\theta^2,\theta^3) &= \mathbf{x}_{\mathrm{surf}}(\theta^1,\theta^2)+\theta^3\mathbf{a}_3,\end{aligned} \tag{3.1}$$

where $\mathbf{X}_{\text{surf}}$ and $\mathbf{x}_{\text{surf}}$ are the midsurfaces in the different configurations and θ^{α} are the contravariant coordinates. The third contravariant coordinate θ^3 is the parameter for the thickness t defined in the range $\left[-\frac{t}{2}, +\frac{t}{2}\right]$, whereby t remains constant. The displacement of the physical domain between the two configurations can thus be described by using the displacement of the midsurface

$$\mathbf{u}_{\text{surf}} = \mathbf{x}_{\text{surf}} - \mathbf{X}_{\text{surf}}. \tag{3.2}$$

3.2.2 Strong form of equilibrium

The equilibrium for a geometrically nonlinear KL shell is given by the two vectorial equations in the current configuration (Basar et al. [4]) by

$$\begin{aligned} \mathbf{n}^{\alpha}|_{\alpha} + \mathbf{p} &= \mathbf{0}, \\ \mathbf{m}^{\alpha}|_{\alpha} + \mathbf{a}_{\alpha} \times \mathbf{n}^{\alpha} &= \mathbf{0}, \end{aligned} \tag{3.3}$$

where $\square|_{\alpha}$ is the covariant derivative w.r.t. the parameters θ^{α}, $\mathbf{a}_{\alpha}$ are the covariant base vectors, $\mathbf{n}^{\alpha}$ (normal forces) and $\mathbf{m}^{\alpha}$ (moments) are the stress resultants derived from the *Cauchy* stress tensor and $\mathbf{p}$ is the external load.

3.3 Finite element method

Structural analysis problems, e.g. Equation (3.3) with a geometrical description and appropriate boundary conditions, are usually solved with the finite element method (FEM) by using the same discretization for geometry and solution fields (*isoparametric concept* – introduced by B. M. Irons [3]).

3.3.1 Definition of nomenclature

To avoid confusions in the usage of different nomenclature within FEM important terms are defined for this thesis as follows:

- The set of parameters $\mathcal{A}$ used for a function $f(\mathcal{A})$ is called **parameter (trimmed) domain**.
- A **parametric domain** is the geometry defined by $\mathcal{G}(\mathcal{A}) = \sum_i f^i(\mathcal{A})$ which at its turn is described by the functions $f^i(\mathcal{A})$ sharing the same *parameter domain* $\mathcal{A}$.

- An **integration domain** is a domain on which an integration scheme, e.g. Gaussian quadrature, is applied.
- With respect to the above mentioned terms a **finite element** is defined differently for *classical finite element analysis* (see Section 3.4.1), *isogeometric analysis* (Section 3.5.3) and *isogeometric B-Rep analysis* (Section 4.3).

3.3.2 Weak form of equilibrium

For arriving at a solution to a structural problem using approximation methods like FEM, the internal and external forces need to be in equilibrium in a weak sense. This can be expressed by the principle of virtual work, which is defined as the sum of the internal and external virtual work (see also Zienkiewicz, O.C. and Taylor, R.L. and Zhu, J.Z. [94]) and is given by

$$\delta W = \delta W_{\text{int}} + \delta W_{\text{ext}} = 0. \tag{3.4}$$

The internal and external virtual work for a KL shell in the reference configuration on the physical domain (no boundary contributions considered – see also Section 4.2) can be formulated as follows (see also Basar et al. [4]):

$$\begin{aligned} \delta W_{\text{int}} &= -\int_{\Omega} \left(\mathbf{N}\,\delta\boldsymbol{\epsilon} + \mathbf{M}\,\delta\boldsymbol{\kappa}\right) d\Omega, \\ \delta W_{\text{ext}} &= \int_{\Omega} \boldsymbol{p}\;\delta\mathbf{u}\, d\Omega, \end{aligned} \tag{3.5}$$

where Ω is the midsurface of the physical domain in the reference configuration, $\mathbf{N}$ (normal force) and $\mathbf{M}$ (moments) are the stress resultants derived from the 2^{nd} Piola Kirchhoff (PK2) stress tensor, $\delta\boldsymbol{\epsilon}$ (virtual normal strain) and $\delta\boldsymbol{\kappa}$ (virtual change in curvature) are the energetically conjugated quantities derived from the virtual Green-Lagrange strain tensor all given in *Voigt* notation, $\boldsymbol{p}$ is the external force, and $\delta\mathbf{u}$ is the virtual displacement of the midsurface.

The equilibrium condition in Equation (3.4) resp. Equation (3.5) must also be fulfilled for the variation w.r.t. the virtual displacement field $\delta\mathbf{u}$:

$$\delta W = \frac{\partial W}{\partial \mathbf{u}} \delta\mathbf{u} = 0 \tag{3.6}$$

3.3.3 Discretization

Applying a discretization for Equation (3.6) the equilibrium can be written as

$$\delta W = -\mathbf{R} \cdot \delta \mathbf{u}_\mathrm{h} = 0, \tag{3.7}$$

which means that for an arbitrary virtual discretized displacement vector $\delta \mathbf{u}_\mathrm{h}$ the corresponding residual force vector $\mathbf{R}$ must vanish. The nonlinear expression in Equation (3.7) is linearized to solve it with an iterative solution approach such as the *Newton-Raphson method*. This results in the linear expression

$$\mathbf{K}\, \Delta \mathbf{u} = \mathbf{R}, \tag{3.8}$$

where the tangential stiffness matrix $\mathbf{K}$ and the residual force $\mathbf{R}$ are used iteratively to obtain the displacement increment $\Delta \mathbf{u}$ until equation (3.7) is satisfied. The components of $\mathbf{K}$ and $\mathbf{R}$ are given by

$$R_r = -\frac{\partial W}{\partial u_r} = -\frac{\partial W_\mathrm{int}}{\partial u_r} - \frac{\partial W_\mathrm{ext}}{\partial u_r} = R_r^\mathrm{int} + R_r^\mathrm{ext}, \tag{3.9}$$

$$K_{rs} = \frac{\partial R_r}{\partial u_s} = -\frac{\partial W^2}{\partial u_r \partial u_s} = -\frac{\partial W_\mathrm{int}^2}{\partial u_r \partial u_s} - \frac{\partial W_\mathrm{ext}^2}{\partial u_r \partial u_s} = K_{rs}^\mathrm{int} + K_{rs}^\mathrm{ext}, \tag{3.10}$$

where $r, s \in \mathbb{N}$ are indices used for the discretization components.

Basis functions

Following the *isoparametric concept* the discretized solution $\mathbf{u}_\mathrm{h}$ i.e. the displacement of the midsurface for a KL shell problem can be written as

$$\mathbf{u}_\mathrm{h} = \mathbf{x}_\mathrm{h} - \mathbf{X}_\mathrm{h} = \sum_r N_r (\hat{\mathbf{x}}^r - \hat{\mathbf{X}}^r) = \sum_r N_r \hat{\mathbf{u}}^r, \tag{3.11}$$

with N_r being the basis functions, $\hat{\mathbf{x}}^r$ resp. $\hat{\mathbf{X}}^r$ the corresponding geometry discretization parameters in vector form, and $\hat{\mathbf{u}}^r$ the corresponding variables for the displacement field $\mathbf{u}_\mathrm{h}$. Considering that for the basis functions usually more than one *parameter domain* is used, the virtual work expressions in Equation (3.5) can be computed as

$$\begin{aligned}
\delta W_\mathrm{int} &= -\sum_k \int_{\Omega^{(k)}} \left(\mathbf{N}\,\delta\boldsymbol{\epsilon} + \mathbf{M}\,\delta\boldsymbol{\kappa}\right) d\Omega^{(k)}, \\
\delta W_\mathrm{ext} &= \sum_k \int_{\Omega^{(k)}} \boldsymbol{p}\ \delta\mathbf{u}\, d\Omega^{(k)},
\end{aligned} \tag{3.12}$$

where $\Omega^{(k)}$ is the *parametric domain k*. Since the KL shell formulation relies on the evaluation of curvature resp. second derivatives of the shape functions, the formulation requires at least C^1 continuity within all *parametric domains* and G^1 continuity (see also Section 4.2) across their boundaries.

The equilibrium condition in Equation (3.4) with Equation (3.5) does not consider any virtual work contributions on boundaries of *parametric domains* and thus it requires the strong fulfillment of

- continuity on common boundaries of *parametric domains* i.e. tangential continuity G^1 including positional continuity G^0 (see also Section 2.4), and
- *Dirichlet* boundary conditions.

Neither classical *finite element analysis* (FEA), which usually uses *Lagrange* polynomials as basis functions, nor the NURBS-based isogeometric analysis (IGA) fulfill these requirements. The advantage of NURBS-based IGA is that one *parametric domain* i.e. a NURBS geometry (see also Section 2.2.1) resp. *patch* is able to represent a large amount of complex shapes and also allows for an accurate approximation of the solution field due to the possibility of geometry refinement (see also Section 2.2.2).

Classical FEA with its low order polynomials is mainly restricted to RM shell formulations which only require G^0 continuity across *parametric domains*. The latter can be achieved easily by sharing *degrees of freedom* (DOFs) at common boundaries. In contrast to that for NURBS-based IGA a strong fulfillment of G^0 across *parametric domains* represents a huge restriction in geometric modeling and analysis because it requires matching patches over the entire physical domain. Nevertheless the high continuity within *patches* allows for KL shell formulations (see Kiendl et al. [44]) as well as *matching multi-patch* geometries by using the *bending-strip method* (see Kiendl [46]).

Chapter 4 explains how these limitations for design and analysis can be overcome by considering virtual work contributions on arbitrarily defined boundaries of *parametric domains* which finally allow for the direct analysis of surface CAD models. As reference for Chapter 4 the following sections briefly explain the geometry description of surface models for classical FEA and IGA.

3.4 Classical finite element analysis (FEA)

The classical finite element analysis usually uses linear polynomials for solving RM shell problems, whereby they are used as basis for representing geometry and solution fields (*isoparametric concept*). The basis function N_i of a node i (see also Figure 3.1) is given by

$$N_i = \bigcup_k \tilde{N}_i^{(k)} \quad \text{with} \quad k \in \mathbb{N} \tag{3.13}$$

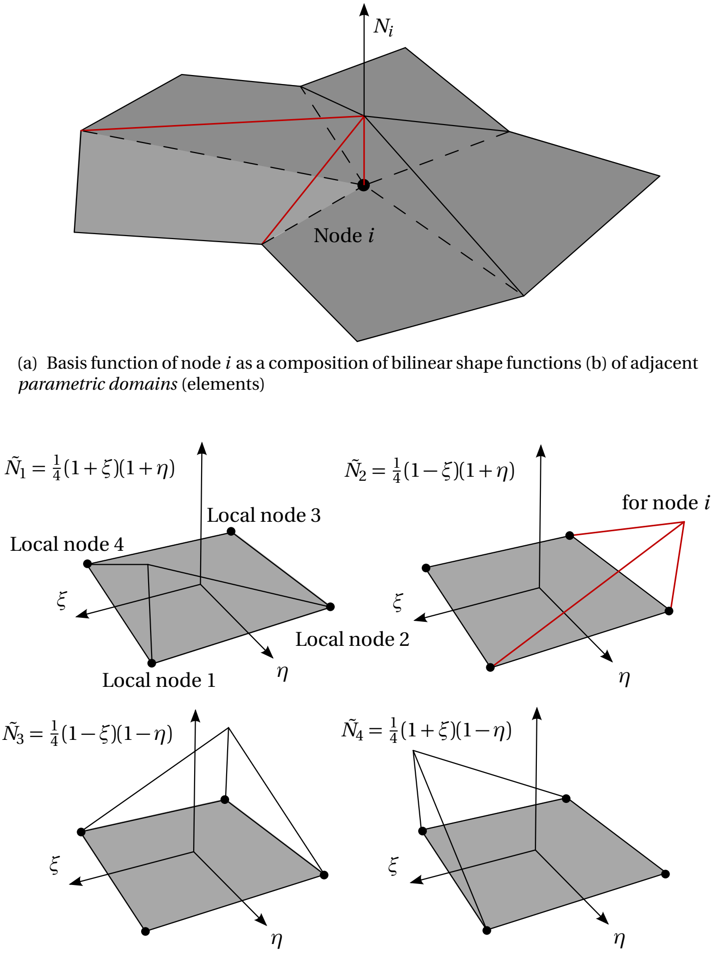
where k is the index used for the *parametric domains,* which contribute to node i, and $\tilde{N}_i^{(k)}$ are the corresponding shape functions (see also Figure 3.1(b)).

3.4.1 Definition of classical finite elements

In classical FEA an element is described by a *parametric domain* whose *parameter domain* coincides with the *integration domain* (see also Section 3.3.1). This definition of an element usually leads to fully populated element stiffness matrices and allows for a simple and efficient implementation.

3.4.2 Geometry description (finite element mesh)

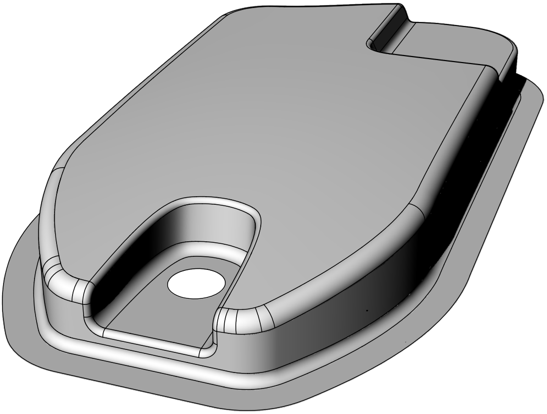
For a structural shell analysis of a given CAD model (see Chapter 2) the *classical finite element analysis* (FEA) requires a geometry transformation called *meshing* (see Figure 3.2). The resulting discretization is named *finite element* (FE) mesh and is used for describing the geometry and solution


N_i
Node i
$\tilde{N}_1 = \frac{1}{4}(1+\xi)(1+\eta)$
Local node 3
Local node 4
ξ
Local node 2
Local node 1
η
$\tilde{N}_2 = \frac{1}{4}(1-\xi)(1+\eta)$
for node i
ξ
η
$\tilde{N}_3 = \frac{1}{4}(1-\xi)(1-\eta)$
ξ
η
$\tilde{N}_4 = \frac{1}{4}(1+\xi)(1-\eta)$
ξ
η

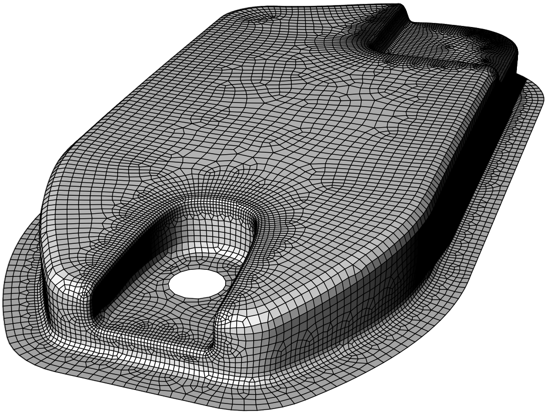
(a) Basis function of node i as a composition of bilinear shape functions (b) of adjacent *parametric domains* (elements)

(b) Bilinear shape functions of a quadrilateral element

Figure 3.1: Example of a basis function used within classical FEA



(a) CAD (NURBS-based B-Rep) model of an oil sump



(b) A corresponding finite element mesh

Figure 3.2: Comparison of a CAD geometry and a classical FE geometry discretization representing the same object. The operation of creating a FE mesh from a CAD model is called *meshing*.

Node ID	X	Y	Z
2016	1.2452	5.6546	8.1116
4908	9.8787	5.1315	8.8454
...	...	...	...

Table 3.1: Exemplary list of FE nodes with coordinates

Elem ID	Node IDs			
342	2016	2017	3015	1554
757	15	6476	3645	
7588	151	6541	2415	
...	...	...	...	

Table 3.2: Exemplary *list of elements* with quadrilateral and triangular elements

fields. Since the discretization parameters of the classical FEA use low order *Lagrange* polynomials, the meshing represents just an approximation of the CAD model and thus it introduces an error. In addition, the link to the shape parameters of CAD models gets lost and last but not least meshing is a highly complex task in itself because FE meshes have to fulfill strict criteria (see also Abel Coll Sans [1], Hansen et al. [37], Knupp et al. [49], and Topping [89]). An example of a FE mesh derived from a CAD model is illustrated in Figure 3.2.

A FE mesh can be described by

- a list of *nodes* (e.g. Table 3.1) and
- a *list of elements* which defines the topology of the mesh (e.g. Table 3.2).

3.5 Isogeometric analysis (IGA)

The idea behind *isogeometric analysis* (IGA), introduced by Hughes et al. [40], is to avoid the *meshing* required in classical FEA by using the basis functions used for representing CAD geometry also for approximating solution fields. Usually, IGA is based on NURBS because these represent the standard for geometry description in current CAD systems (see Chapter 2). A detailed description of NURBS basis functions and surfaces is given in Section 2.2.1. Since NURBS surfaces used for representing the shape of an object are usually not able to represent solution fields with a satisfying accuracy, NURBS surfaces need to be refined resp. enriched with control points (see also Section 2.2.2). Note that such a refinement neither affects the initial shape nor its parametrization.

3.5.1 *h*-,*p*-,*h*-*p*-, and *k*-refinement

Depending on the sequence of applying *knot insertion* and *degree elevation* one distinguishes between *h*-,*p*-,*h*-*p*, and *k*-refinement (see also Cottrell et al. [24] and Hughes et al. [40]):

- For *h*-refinement additional single knots are added to the knot vector but no degree elevation is performed. Note that each inserted knot reduces the continuity C at its position.

- *p*-refinement is achieved by performing *degree elevation*, with the result that the polynomial degree within each *non-zero knot span* is increased without modifying the continuities at knots.

- *h*-*p*-refinement is achieved by performing *degree elevation* after *h*-refinement, with the result that the polynomial degree within each *non-zero knot span* is increased without modifying the continuities at knots.

- *k*-refinement is achieved by first applying *degree elevation* and then *knot insertion*. With this sequence one ends up with the same number of *non-zero knot spans* as with *h*-refinement and the same polynomial degree as with *p*-refinement but with a higher continuity between *non-zero knot spans* than *h*-*p*-refinement.

Since high continuity is beneficial for KL shell analysis, *k*-refinement is assumed in the sequel.

Node ID	X	Y	Z	W
2016	1.2452	5.6546	8.1116	1.0000
3015	2.3524	6.1297	8.3435	1.0000
4908	9.8787	5.1315	8.8454	1.0000
...	...	...	...	...

Table 3.3: Exemplary list of FE nodes with coordinates and weights

Patches	Description
NurbsID=1561	id of NURBS surface
p=2	polynomial degree in ξ-direction
q=2	polynomial degree in η-direction
U=[0,0,0, ... 1,1,1]	knot vector $\mathbf{\Xi}$
V=[0,0,0, ... 1,1,1]	knot vector $\mathbf{H}$
NodeIDs=[101,87,12-16,...,12]	list of node ids
NurbsID=243	id of NURBS surface
...	...

Table 3.4: Exemplary list of *patches* with their NURBS description consisting of polynomial degrees, knot vectors, and node ids

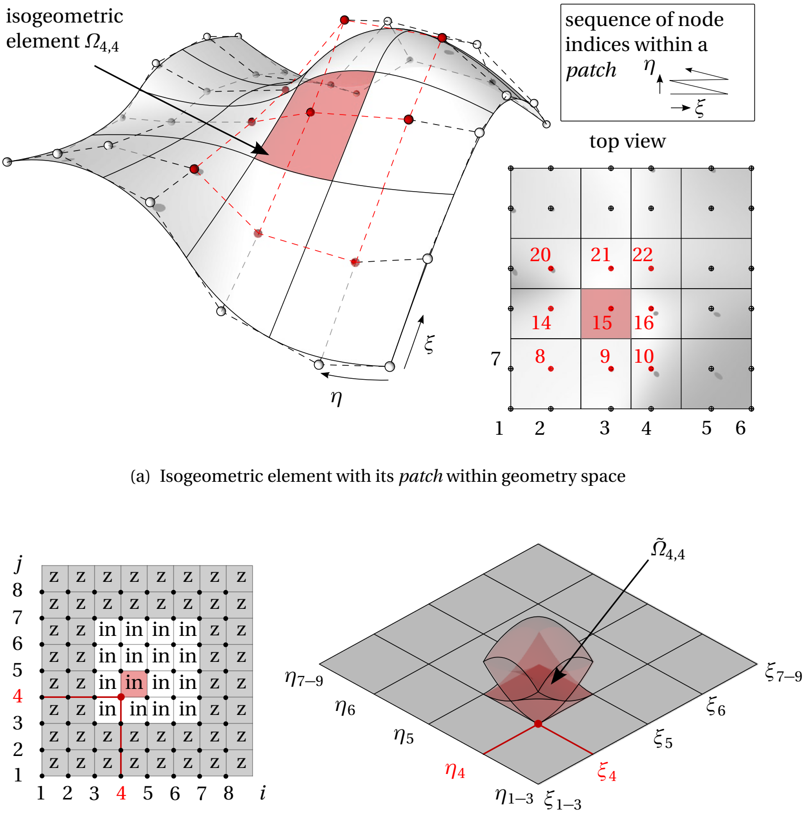
3.5.2 Geometry description

In contrast to a classical FE mesh, which consists of a list of nodes and a *list of elements* (see Section 3.4.2), a description of an isogeometric surface model requires

- a list of *nodes*, which contain node coordinates and their weights (e.g. Table 3.3), and
- a list of *patches* (e.g. Table 3.4).

For a complete data set in Table 3.4 the following holds

$$n_{\text{nodes } \xi\text{-dir}} = (n_{\text{knots } \xi\text{-dir}} - p - 1) \tag{3.14}$$



(a) Isogeometric element with its *patch* within geometry space

(b) Index space

(c) Parameter space with the shape functions of the element

Figure 3.3: Example of an isogeometric element within geometry, parameter, and index space

$$n_{\text{nodes } \eta\text{-dir}} = (n_{\text{knots } \eta\text{-dir}} - q - 1) \tag{3.15}$$

$$n_{\text{nodes}} = n_{\text{nodes } \xi\text{-dir}} \cdot n_{\text{nodes } \eta\text{-dir}} \tag{3.16}$$

$$n_{\text{node ids}} = n_{\text{nodes}} \tag{3.17}$$

where $n_\square$ stands for the number of different entities.

3.5.3 Definition of isogeometric elements

Within this thesis *isogeometric elements* are defined as proposed in Hughes et al. [40] by *non-zero knot spans* i.e. on *parameter subdomains* which are also used as *integration domains*. This element definition usually leads to fully populated element stiffness matrices and allows for a simple and efficient assembly. Thus from an implementation point of view it is better suited than an element definition according to the classical finite element analysis (see also Section 3.4.1).

For creating isogeometric elements the *parameter domain* $\mathcal{H}$ of a *patch* (see also Section 2.2.3 and Table 3.4) needs to be decomposed into *knot spans* resp. *parameter subdomains* $\mathcal{H}_{ij} \subset \mathcal{H}$ with $i, j \in \mathbb{N}$

$$\mathcal{H} = \bigcup_i \bigcup_j \mathcal{H}_{ij}. \tag{3.18}$$

The isogeometric surface element Ω_{ij} is then described by the *parameter subdomain* $\tilde{\Omega}_{ij}$ which corresponds to the *non-zero knot span* $\mathcal{H}_{ij}$

$$\begin{aligned}&\Omega_{ij} = \{\mathbf{S}(\xi,\eta) \mid (\xi,\eta) \in \tilde{\Omega}_{ij}\}, \text{ where}\\&\tilde{\Omega}_{ij} = \mathcal{H}_{ij} = [\xi_i, \xi_{i+1}[\times [\eta_j, \eta_{j+1}[\subset \mathcal{H} \text{ with } \xi_i \neq \xi_{i+1} \text{ and } \eta_j \neq \eta_{j+1}.\end{aligned} \tag{3.19}$$

The identification of an element within a given *patch* (surface), is given by the *knot indices* $i, j \in \mathbb{N}$ of the knot span $\mathcal{H}_{ij}$. An example of an isogeometric surface element within the geometry space is shown in Figure 3.3(a), whereas Figure 3.3(b) and Figure 3.3(c) illustrate the corresponding element within the parameter resp. index space (see also Cottrell et al. [24]).

Table 3.5 contains indicators which can be used for categorizing knot spans of *patches*. After the assignment of these indicators isogeometric elements can be derived from the *patches* straight forward by considering the sequence of nodes (see also Figure 3.3(a)) and the relation between number of knots and nodes per direction (see Equation (3.14) and Equation (3.15)).

Geometrical description of isogeometric elements

Like classical finite elements (see Section 3.4.2) isogeometric elements also require a set of nodes (control points) and "shape functions" for their

Indicators	Description
in	non-zero knot span
z	zero knot span

Table 3.5: Indicators for distinguishing different types of knot spans

Element (i, j)	Description
id=1433	unique id of the element
p=2	polynomial degree in ξ-direction
q=2	polynomial degree in η-direction
U=$[\xi_{i-p},...,\xi_{i+p+1}]$	knot vector $\mathbf{\Xi}$
V=$[\eta_{j-q},...,\eta_{j+q+1}]$	knot vector **H**
NodeIDs=[54,56,...,78,89]	list of element node ids

Table 3.6: Example of an isogeometric element and its geometry discretization

geometrical description. Table 3.6 exemplarily summarizes the geometry description of a NURBS-based surface element.

Besides the nodes (see also Figure 3.3(a)) an isogeometric element needs its polynomial degrees and relevant parts of the knot vectors i.e. $U = [\xi_{i-p},...,\xi_{i+p+1}]$ and $V = [\eta_{j-q},...,\eta_{j+q+1}]$ for computing the basis functions and their derivatives (see e.g. algorithm A2.3 in Piegl et al. [65]).

Summary and conclusion of Chapter 3

This chapter explains the difference between classical FEA and IGA w.r.t. to their geometry discretizations used for analysis and serves as reference for IBRA (see Chapter 4). The geometrical element description used for classical FEA and IGA, including their shape functions and possible data structures are explained as well.

IGA in its original form provides the basis for developing element formulations capable of analyzing CAD models but it does not provide the framework for handling the topology of complex (trimmed multi-patch) CAD models (see also Chapter 2). This gap will be closed with the new *isogeometric B-Rep analysis* (IBRA) which is explained in Chapter 4 for geometrically nonlinear KL shell problems.

"Das geht nicht." Dann kam
einer, der wusste das nicht,
und hat es gemacht.

Hilbert Meyer,
Professor für Schulpädagogik
an der Universität Oldenburg

CHAPTER 4

Isogeometric B-Rep analysis

Isogeometric B-Rep analysis (IBRA) introduced by Breitenberger et al. [17] is a finite element approach which enables the direct analysis on surface CAD models (see also Section 2.2). IBRA extends *isogeometric analysis* (IGA), introduced by Hughes et al. [40] (see also Section 3.5), by adapting the basis functions as well as the B-Rep description of the CAD model for approximating solution fields. IBRA provides a consistent framework to deal with the geometry and topology of CAD models for structural analysis in a finite-element-like manner and closes the gap between CAD models and the *finite element method* (FEM).

The chapter explains IBRA for geometrically nonlinear *Kirchhoff-Love* (KL) shell problems. A data structure for *isogeometric (trimmed) elements* as well as for *B-Rep elements* is proposed. Moreover, a *B-Rep element* formulation based on a penalty approach is introduced which can be used to couple discontinuous and trimmed geometries and to enforce *Dirichlet* boundary conditions.

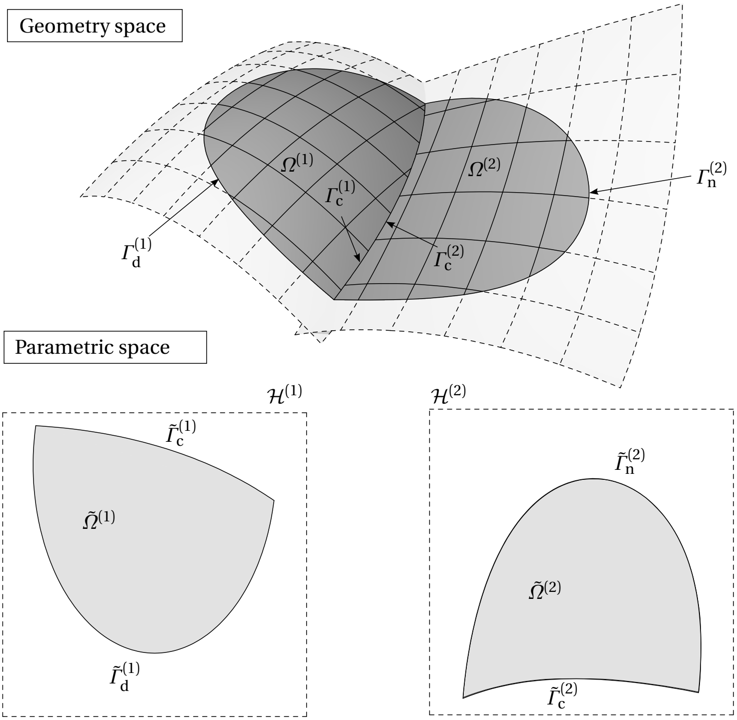

Geometry space
$\Omega^{(1)}$
$\Omega^{(2)}$
$\Gamma_{\mathrm{c}}^{(1)}$
$\Gamma_{\mathrm{n}}^{(2)}$
$\Gamma_{\mathrm{d}}^{(1)}$
$\Gamma_{\mathrm{c}}^{(2)}$
Parametric space
$\mathcal{H}^{(1)}$
$\mathcal{H}^{(2)}$
$\tilde{\Gamma}_{\mathrm{c}}^{(1)}$
$\tilde{\Gamma}_{\mathrm{n}}^{(2)}$
$\tilde{\Omega}^{(1)}$
$\tilde{\Omega}^{(2)}$
$\tilde{\Gamma}_{\mathrm{d}}^{(1)}$
$\tilde{\Gamma}_{\mathrm{c}}^{(2)}$

Figure 4.1: Description of a shell with arbitrarily defined boundary conditions denoted with the indices c,d, and n within isogeometric B-Rep shell analysis

4.1 Geometrical description of shells

In the context of IBRA the midsurface of a KL shell structure $\Omega \subset \mathbb{R}^3$ is described resp. given by a NURBS-based B-Rep (CAD) model (see also Section 2.2) and thus it is a composition of open and bounded sets $\Omega^{(a)} \subset \mathbb{R}^3$

$$\Omega = \bigcup_a \Omega^{(a)}, \tag{4.1}$$

with $a \in \mathbb{N}$ (see also Figure 4.1). Each subset $\Omega^{(a)}$ is represented by one trimmed surface $\mathbf{S}^{(a)}_{\text{visible}}$ and its boundary $\Gamma^{(a)}$ by $\partial\,\mathbf{S}^{(a)}_{\text{visible}}$. The corresponding trimmed domains and trimming curves are denoted with $\tilde{\Omega}^{(a)} = \mathcal{D}^{(a)} \subset \mathcal{H}^{(a)}$ and $\tilde{\Gamma}^{(a)} = \partial\,\tilde{\Omega}^{(a)}$ in the parameter space, respectively (see also Section 2.3). Figure 4.1 exemplarily shows a shell in geometry and parameter space, with a decomposition of the boundaries $\Gamma^{(a)}$ resp. $\tilde{\Gamma}^{(a)}$ into subsets:

$$\begin{aligned}\Gamma^{(a)} &= \left(\bigcup_k \Gamma_{\mathrm{d}}^{(a)\;k}\right) \cup \left(\bigcup_l \Gamma_{\mathrm{n}}^{(a)\;l}\right) \cup \left(\bigcup_m \Gamma_{\mathrm{c}}^{(a)\;m}\right),\\ \tilde{\Gamma}^{(a)} &= \left(\bigcup_k \tilde{\Gamma}_{\mathrm{d}}^{(a)\;k}\right) \cup \left(\bigcup_l \tilde{\Gamma}_{\mathrm{n}}^{(a)\;l}\right) \cup \left(\bigcup_m \tilde{\Gamma}_{\mathrm{c}}^{(a)\;m}\right),\end{aligned} \tag{4.2}$$

where the following indices are used for expressing the different types of boundary conditions:

- d for *Dirichlet* boundary conditions (e.g. displacement, rotation)
- n for *Neumann* boundary conditions (e.g. pressure load)
- c for coupling (internal) boundary conditions (preservation of continuities)

Mechanical boundary formulations like cables (see Philipp et al. [63]) or beams (see Bauer et al. [5]) are not considered in the remainder of this thesis. An example of an internal boundary is shown in Figure 4.1 where the sets $\Omega^{(1)}$ and $\Omega^{(2)}$ share a common boundary Γ_c

$$\Gamma_c = \Gamma_c^{(1)} = \Gamma_c^{(2)}. \tag{4.3}$$

4.2 Weak form of equilibrium

In contrast to classical FEA and IGA *isogeometric B-Rep analysis* (IBRA) requires the consideration of virtual work contributions along domain boundaries because a strong fulfillment of *Dirichlet* boundary conditions on arbitrarily defined trimming curves is not possible (see also Section 3.3).

The weak form of the static equilibrium in Equation (3.3) for a geometrically nonlinear KL shell including boundary terms is given by (see Basar et al. [4], Eq. 8.3.141 – other notation is used)

$$\begin{aligned}\delta W = &\int_{\Omega} \boldsymbol{p}\ \delta \mathbf{u}\, d\Omega - \int_{\Omega} \mathbf{N}\, \delta \boldsymbol{\epsilon} + \mathbf{M}\, \delta \boldsymbol{\kappa}\ d\Omega \\ &+ \int_{\Gamma} \mathbf{n}_{\Gamma} \delta \mathbf{u} + m_{\mathbf{T}_2} \delta \omega_{\mathbf{T}_2}\, d\Gamma.\end{aligned} \tag{4.4}$$

The first two integrals are the same as they are already used in Equation (3.12). The third one represents the virtual work contribution on the domain boundary with $\mathbf{n}_{\Gamma}$ (traction force) and $m_{\mathbf{T}_2}$ (moment around the boundary tangent $\mathbf{T}_2$) being the stress resultants derived from the 1st Piola-Kirchhoff (PK1) stress tensor using the local coordinate system as it is defined in Equation (2.26). The term $\delta \omega_{\mathbf{T}_2}$ expresses the virtual rotation around the local base vector $\mathbf{T}_2$ on the boundary (see also Figure 4.5). The rotation $\omega_{\mathbf{T}_2}$ is given by

$$\omega_{\mathbf{T}_2} = \arcsin(\boldsymbol{\omega} \cdot \mathbf{T}_2), \tag{4.5}$$

where $\boldsymbol{\omega}$ is the rotation vector (see Basar et al. [4]) that is defined as

$$\boldsymbol{\omega} = \mathbf{T}_3 \times \mathbf{w}, \tag{4.6}$$

with $\mathbf{w}$ being the displacement of the tip of the local base vector $\mathbf{t}_3$ in the current configuration, given by

$$\mathbf{w} = \mathbf{t}_3 - \mathbf{T}_3. \tag{4.7}$$

Because the *arcsin* function is involved, the rotations at the shell boundaries are restricted to be less than $\frac{\Pi}{2}$.

4.2.1 Equilibrium along internal boundaries

Considering common internal boundaries as exemplarily given in Equation (4.3) the equilibrium needs to be fulfilled also on those boundaries and is given by

$$\mathbf{n}_{\Gamma} = \mathbf{n}_{\Gamma}^{(1)} = -\mathbf{n}_{\Gamma}^{(2)} \tag{4.8}$$

for the traction forces and for the moments it is given by

$$m_{\mathbf{T}_2} = m_{\mathbf{T}_2}^{(1)} = \pm m_{\mathbf{T}_2}^{(2)}. \tag{4.9}$$

Note that the traction forces $\mathbf{n}_{\Gamma}^{(\alpha)}$ point in opposite directions and the sign in Equation (4.9) depends on how the tangent vectors $\mathbf{T}_2$ of the involved boundary curves are oriented with respect to each other. The negative sign is used in the case that both tangent vectors $\mathbf{T}_2$ point in the same direction.

The virtual work along a common internal boundary can thus be computed as follows

$$\delta W_{\text{coupling}}^{\text{displacement}} = \int_{\Gamma_{\text{c}}^{(1)}} \mathbf{n}_{\Gamma}(\delta \mathbf{u}^{(1)} - \delta \mathbf{u}^{(2)})\, d\Gamma_{\text{c}}^{(1)}, \tag{4.10}$$

$$\delta W_{\text{coupling}}^{\text{rotation}} = \int_{\Gamma_{\text{c}}^{(1)}} m_{\mathbf{T}_2}(\delta \omega_{\mathbf{T}_2}^{(1)} \pm \delta \omega_{\mathbf{T}_2}^{(2)})\, d\Gamma_{\text{c}}^{(1)}, \tag{4.11}$$

using the sign convention explained above. Applying a test function $\boldsymbol{\delta}\mathbf{u}$ which fulfills the required continuity conditions

- G^0 continuity for Equation (4.10)
- G^1 continuity for Equation (4.11)

The corresponding virtual work term for coupling geometrical continuity along element edges vanishes. For *isogeometric B-Rep analysis* (IBRA) the test function (NURBS-based B-Rep model) is not able to meet these requirements thus the corresponding boundary terms need to be considered for analysis.

As a remark, for RM shells within classical FEA the boundary terms can be neglected because the test function fulfills the required G^0 continuity (see also Section 3.4) by sharing all DOFs at common elements.

4.2.2 Equilibrium used for isogeometric B-Rep analysis

In the context of IBRA for KL shell problems the virtual work contributions on the boundaries (see third integral in Equation (3.5) , Equation (4.10), and Equation (4.11)) are collected within the expression $\delta W_{\text{B-Rep}}$ because their geometrical description is provided by the entities of the *boundary* *representation* (B-Rep) i.e. *edges* resp. trimming curves of the B-Rep model. The equilibrium for IBRA can thus be written as

$$\delta W = \delta W^*_{\text{int}} + \delta W^*_{\text{ext}} + \delta W_{\text{B-Rep}} = 0, \tag{4.12}$$

with δW^* being virtual work expressions explicitly without boundary contributions. Hence for solving Equation (4.12) the terms in Equation (3.9) and Equation (3.10) known from classical FEA and IGA need to be enriched by the components derived from the B-Rep contributions i.e. the entries of the residual force vector and the stiffness matrix

$$R_r = -\frac{\partial W^*_{\text{int}}}{\partial u_r} - \frac{\partial W^*_{\text{ext}}}{\partial u_r} - \frac{\delta W_{\text{B-Rep}}}{\partial u_r} = R_r^{\text{int}} + R_r^{\text{ext}} + R_r^{\text{B-Rep}}, \tag{4.13}$$

$$K_{rs} = -\frac{\partial W^*_{\text{int}}}{\partial u_r \partial u_s} - \frac{\partial W^*_{\text{ext}}}{\partial u_r \partial u_s} - \frac{\partial W_{\text{B-Rep}}}{\partial u_r \partial u_s} = K_{rs}^{\text{int}} + K_{rs}^{\text{ext}} + K_{rs}^{\text{B-Rep}}. \tag{4.14}$$

4.3 Definition of elements

In the context of IBRA the following two types of elements need to be defined:

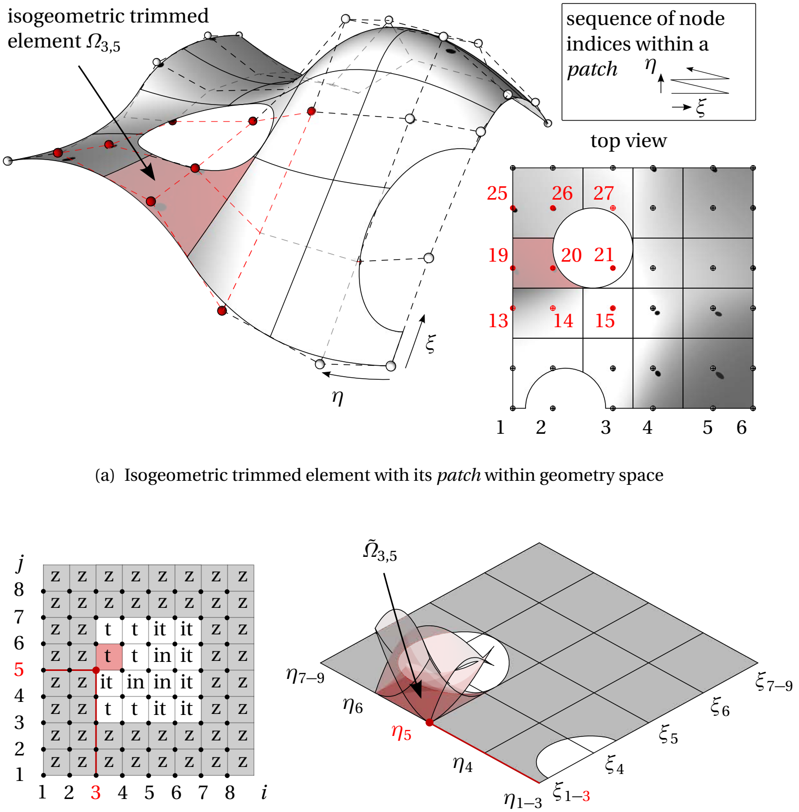
- *isogeometric (trimmed) elements* and
- *isogeometric B-Rep elements* or just *B-Rep elements.*

After the definition of these two element types the following sections explain corresponding element formulations which can be used for KL shell problems.

4.3.1 Isogeometric (trimmed) elements

Within this thesis an *isogeometric (trimmed) element* is basically defined as it is done in Section 3.5.3 just with the extension of considering the *trimmed domain* $\mathcal{D} \subset \mathcal{H}$ of the trimmed NURBS surface (see also Section 2.2.3) with

$$\mathcal{D} = \bigcup_i \bigcup_j \mathcal{D}_{ij} \text{ and } \mathcal{D}_{ij} \subset \mathcal{H}_{ij}. \tag{4.15}$$



(a) Isogeometric trimmed element with its *patch* within geometry space

z	z	z	z	z	z	z	z
z	z	z	z	z	z	z	z
z	z	t	t	it	it	z	z
z	z	t	t	in	it	z	z
z	z	it	in	in	it	z	z
z	z	t	t	it	it	z	z
z	z	z	z	z	z	z	z
z	z	z	z	z	z	z	z

(b) Index space

(c) Parameter space with the basis functions of the trimmed element

Figure 4.2: Example of an *isogeometric trimmed element* within geometry, parameter, and index space

An *isogeometric (trimmed) surface element* Ω_{ij} is thus described by a subdomain $\tilde{\Omega}_{ij}$ which corresponds to the *trimmed domain* of the *non-zero knot span* $\mathcal{H}_{ij}$ with

$$\begin{aligned}&\Omega_{ij} = \{\mathbf{S}_{\text{visible}}(\xi,\eta) \mid (\xi,\eta) \in \tilde{\Omega}_{ij}\}, \text{ where}\\&\tilde{\Omega}_{ij} = \mathcal{D}_{ij} \subset [\xi_i, \xi_{i+1}[\times [\eta_j, \eta_{j+1}[\subset \mathcal{H} \text{ with and } \xi_i \neq \xi_{i+1} \;\; \eta_j \neq \eta_{j+1}.\end{aligned} \tag{4.16}$$

Note that in contrast to *isogeometric elements* as defined in Section 3.5.3 the numerical integration of *isogeometric trimmed elements* can require the usage of more than just one *integration domain* $\mathcal{I}_{ij}^{(h)}$ (see also Section 5.4.2) such that

$$\mathcal{H}_{ij} = \bigcup_h \mathcal{I}_{ij}^{(h)} \tag{4.17}$$

where the following holds

$$\mathcal{D}_{ij} = \bigcup_h \mathcal{D}_{ij}^{(h)} \quad \text{with} \quad \mathcal{D}_{ij}^{(h)} \subset \mathcal{I}_{ij}^{(h)} \tag{4.18}$$

An example of an *isogeometric trimmed surface element* within the geometry space is shown in Figure 4.2(a), whereas Figure 4.2(b) and Figure 4.2(c) illustrate the corresponding element within the parameter resp. index space. The indicators applied within the index space are used for categorizing knot spans of the *patch*. Table 4.1 contains an overview of possible indicators for such a categorization. The first three indicators describe *active* elements which contribute to the virtual work expression in Equation (3.5) and the rest describes *inactive* elements. The fine distinction within *active* resp. *inactive* elements allows for a more efficient numerical integration resp. a better visualization of analysis results.

Geometrical description of isogeometric (trimmed) elements

In addition to a set of nodes and "shape functions" (compare Table 3.4) *isogeometric trimmed elements* require a set of properly oriented (trimmed) curve segments which lie inside the element and define the visible domain of the element. The additional flag *FaceNormal* ∈ {TRUE, FALSE} indicates

Indicators	Description
active	
in	completely inside trimmed domain with no boundary curve
it	completely inside trimmed domain with boundary curves
t	trimmed i.e. only partially contributing to trimmed domain
inactive	
st	trimmed knot span with negligible contribution to trimmed domain
ot	outside with basis functions contributing to trimmed domain
out	outside without any contribution to trimmed domain
z	zero knot span

Table 4.1: Extension of the indicators in Table 3.5 for distinguishing different types of knot spans

whether the orientation of the face, from which the element is derived, corresponds to the one of the element given by the surface normal (see base vector $\mathbf{A}_3$ in Section 2.3.1) which at its turn depends on its parametrization. Such flags are used in CAD systems for simplifying the usage of geometry models with equally oriented faces. Moreover, *isogeometric trimmed elements* make use of the indicators (see Table 4.1) assigned to the corresponding knot span. Table 4.2 summarizes exemplarily the geometry description of a trimmed NURBS-based surface element.

4.3.2 B-Rep elements

B-Rep elements are finite elements derived from B-Rep entities such as *vertices* and *edges*. They can be used for imposing analysis properties. The following list contains some examples of such properties

- *Neumann* boundary conditions (e.g. forces, moments)
- *Dirichlet* boundary conditions (e.g. displacements, rotations)
- Internal (coupling) boundary conditions (see also Section 4.2.1)
- Mechanically motivated entities (e.g. cables, beams)

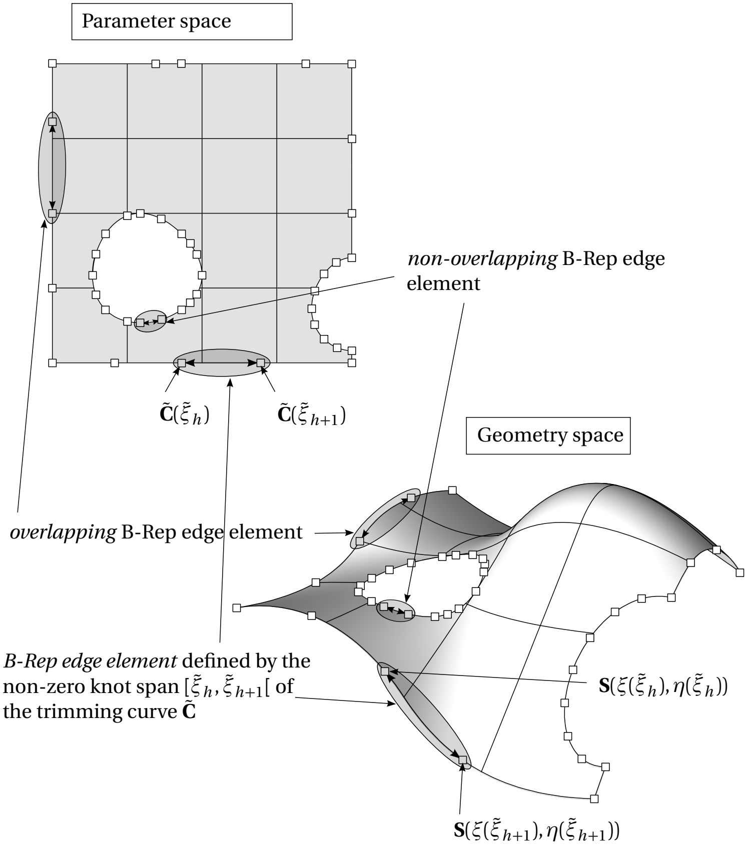
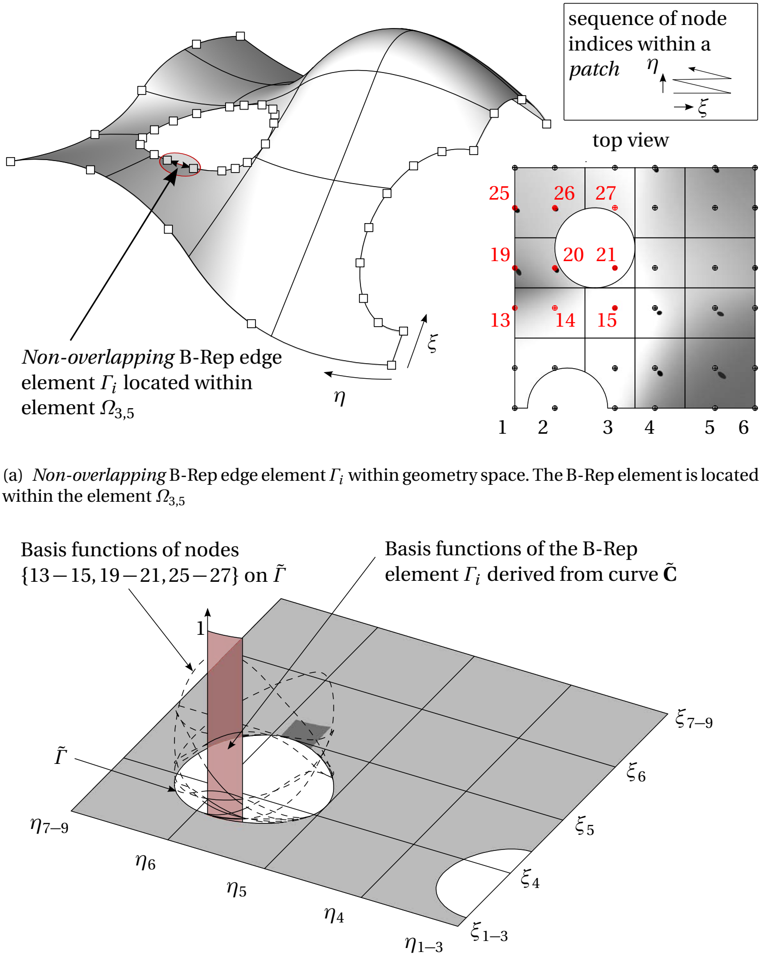

Parameter space
non-overlapping B-Rep edge element
$\tilde{\mathbf{C}}(\tilde{\xi}_h)$
$\tilde{\mathbf{C}}(\tilde{\xi}_{h+1})$
Geometry space
overlapping B-Rep edge element
B-Rep edge element defined by the non-zero knot span $[\tilde{\xi}_h, \tilde{\xi}_{h+1}[$ of the trimming curve $\tilde{\mathbf{C}}$
$\mathbf{S}(\xi(\tilde{\xi}_h), \eta(\tilde{\xi}_h))$
$\mathbf{S}(\xi(\tilde{\xi}_{h+1}), \eta(\tilde{\xi}_{h+1}))$

Figure 4.3: Examples of *untrimmed B-Rep edge elements* in parameter and geometry space, with arbitrarily chosen discretizations of the trimming curves. The figure shows an example of an *overlapping* and a *non-overlapping* B-Rep edge element, as well as the general definition of a B-Rep edge element.


sequence of node indices within a *patch*
η
ξ
top view
25
26
27
19
20
21
13
14
15
1
2
3
4
5
6
Non-overlapping B-Rep edge element Γ_i located within element $\Omega_{3,5}$
ξ
η
Basis functions of nodes $\{13-15, 19-21, 25-27\}$ on $\tilde{\Gamma}$
Basis functions of the B-Rep element Γ_i derived from curve $\tilde{\mathbf{C}}$
1
$\tilde{\Gamma}$
ξ_{7-9}
ξ_6
ξ_5
ξ_4
ξ_{1-3}
η_{7-9}
η_6
η_5
η_4
η_{1-3}

(a) *Non-overlapping* B-Rep edge element Γ_i within geometry space. The B-Rep element is located within the element $\Omega_{3,5}$

(b) Parameter space with the basis functions of the B-Rep edge element

Figure 4.4: Example of a *non-overlapping* B-Rep edge element within geometry, parameter, and index space.

Element (i, j)	Description
id=2345	id of the element
p=2	polynomial degree in ξ-direction
q=2	polynomial degree in η-direction
U=$[\xi_{i-p}, \xi_{i+p+1}]$	knot vector $\boldsymbol{\Xi}$
V=$[\eta_{j-q}, \eta_{j+q+1}]$	knot vector $\mathbf{H}$
NodeIDs=[54,56,...,78,89]	list of element node ids
FaceNormal=true	flag used for face orientation
Indicator=t	indicator from Table 4.1
Curve segments id $[\tilde{\xi}_{\text{start}}, \tilde{\xi}_{\text{end}}]$	curve id and active range
23 [0,0.3]	
...	

Table 4.2: Example of an isogeometric trimmed element and its geometry discretization

The basis functions of NURBS-based B-Rep elements adapt the properties of their underlying functional basis (see also Section 2.2.1) like

- partition of unity, i.e. $\sum_{i=1}^{n} N_{i,p}(\xi) = 1$.
- non-negativity, i.e. $N_{i,p}(\xi) \geqslant 0$.
- linear independence, i.e. $\sum_{i=1}^{n} \alpha_i N_{i,p}(\xi) = 0 \Leftrightarrow \alpha_i = 0,\ i = 1, 2, ..., n$

because they are defined on subsets of the functional *parameter domain*.

4.3.3 *B-Rep edge* elements

A *B-Rep edge element* $\varGamma_i$ is a one-dimensional B-Rep element embedded into a surface description. It is defined by the *parameter subdomain* $\tilde{\varGamma}_i$ which corresponds to the *non-zero knot span* $\mathcal{H}_i$ of the trimming curve $\tilde{\mathbf{C}}$ (see also Section 2.2.3)

$$\begin{aligned} &\varGamma_i = \{\partial \mathbf{S}_{\text{visible}}(\xi, \eta) \mid (\xi, \eta) \in \tilde{\varGamma}_i\}, \text{ where} \\ &\tilde{\varGamma}_i = \{\tilde{\mathbf{C}}(\tilde{\xi}) | \tilde{\xi} \in \mathcal{H}_i = [\tilde{\xi}_i, \tilde{\xi}_{i+1}[\subset \boldsymbol{\Xi} \text{ with } \tilde{\xi}_i \neq \tilde{\xi}_{i+1}\}. \end{aligned} \tag{4.19}$$

In this thesis, *B-Rep edge elements* are embedded into NURBS surfaces and make use of the basis functions from these surfaces. Figure 4.3 shows examples of such elements. For *B-Rep elements* one distinguishes between *non-overlapping* and *overlapping* elements. *Non-overlapping* elements extract the non-zero basis functions from only one underlying isogeometric element but *overlapping* elements extract them from more than one underlying elements. Figure 4.4(b) exemplarily illustrates the basis functions of a *non-overlapping* B-Rep edge element. Usually, no explicit description of these basis functions is available, but this is not required for IBRA because only the evaluation at quadrature points is necessary.

B-Rep edge element	Description
id=234	id of the B-Rep element
CurveID=44	id of trimming curve
Range=[0, 0.5]	active range where the B-Rep element is defined
ElementID=23	id of the underlying surface element

Table 4.3: Example of a *trimmed non-overlapping* B-Rep edge element description which can be used for free edges i.e. no coupling

B-Rep edge element	Description
id=433	id of the B-Rep element
master	
CurveID=165	id of the trimming curve (master)
Range=[0, 0.1]	active range of master curve where the element is defined
ElementID=11	id of the underlying surface element (master)
slave	
CurveID=44	id of the trimming curve (slave)
Range=[0.4, 0.3]	active range of slave curve where the element is defined
ElementID=12	id of the underlying surface element (slave)

Table 4.4: Example of the description of a trimmed *non-overlapping* B-Rep edge coupling element

Description of trimmed *B-Rep edge* elements

In analogy to trimmed curves (see Section 2.2.3) *B-Rep elements* can also be trimmed. The variable *Range* in Table 4.3 extends *B-Rep edge elements* by the trimming concept. *Trimmed B-Rep edge elements* are more flexible w.r.t. the definition of *non-overlapping* B-Rep edge elements and make a refinement of trimming curves unnecessary.

Description of trimmed *B-Rep edge* coupling elements

B-Rep edge coupling elements establish a connection between *isogeometric (trimmed) elements* of different *patches*. Table 4.3 describes exemplarily a trimmed B-Rep edge element. A description of a *trimmed coupling B-Rep edge element* is given in Table 4.4.

B-Rep vertex element	Description
id=24	id of the B-Rep element
u=0.86	ξ parameter which describes the position of the element
v=0.67	η parameter which describes the position of the element
ElementID=3	surface element id

Table 4.5: Example of a *B-Rep vertex element* description

4.3.4 *B-Rep vertex* elements

A *B-Rep vertex element* is defined by the surface parameters which describe the geometrical position of the *vertex* element. Table 4.5 contains the exemplarily description of such an element. A *vertex element* can also be extended for a pointwise coupling of patches analog to the edge coupling elements (see Section 4.3.3).

4.4 NURBS-based *Kirchhoff-Love* shell formulation

The element formulation for geometrically nonlinear KL problems can be derived from the internal virtual work (see also Equation (3.5)) given by

$$\delta W_{\text{int}} = \int_{\Omega} \mathbf{N}\, \delta \boldsymbol{\epsilon} + \mathbf{M}\, \delta \boldsymbol{\kappa}\; d\Omega. \tag{4.20}$$

For a detailed description of such a shell element formulation, using a NURBS discretization, the reader is referred to Kiendl [46].

4.4.1 Alternative shell formulations

Besides the NURBS-based KL shell formulation of Kiendl et al. [44] there are also other NURBS-based shell formulations available like the RM shell formulations of Benson et al. [12] and Dornisch et al. [30, 31], the hierarchical shell formulation of Echter et al. [33] or the blended shell of Benson et al. [11] which can also be used for IBRA.

4.5 *B-Rep edge* element formulations

This section explains a *B-Rep edge element* formulation which can be used for coupling *parameteric domains* (see also Equation (4.10) and Equation (4.11)). It is based on a penalty approach and can be used for geometrically nonlinear KL shell problems (see Breitenberger et al. [17]). Alternative formulations are possible of course.

4.5.1 Internal boundary conditions – penalty approach

The virtual work contribution for the proposed element formulation can be written as follows:

$$\delta W^{\text{coupling}}_{\text{B-Rep}} = \delta W^{\text{disp}}_{\text{B-Rep}} + \delta W^{\text{rot}}_{\text{B-Rep}}, \tag{4.21}$$

where $\delta W^{\text{disp}}_{\text{B-Rep}}$ and $\delta W^{\text{rot}}_{\text{B-Rep}}$ represent the terms in Equation (4.10) and Equation (4.11).

Applying a penalty approach, the traction force $\mathbf{n}_\Gamma$ in the boundary is assumed to be

$$\mathbf{n}_\Gamma = -\alpha_{\text{disp}}(\mathbf{u}^{(1)}_{\text{B-Rep}} - \mathbf{u}^{(2)}_{\text{B-Rep}}), \tag{4.22}$$

where α_{disp} is the penalty factor which can be seen as spring stiffness and is selected manually. Guidelines for selecting an appropriate penalty factor

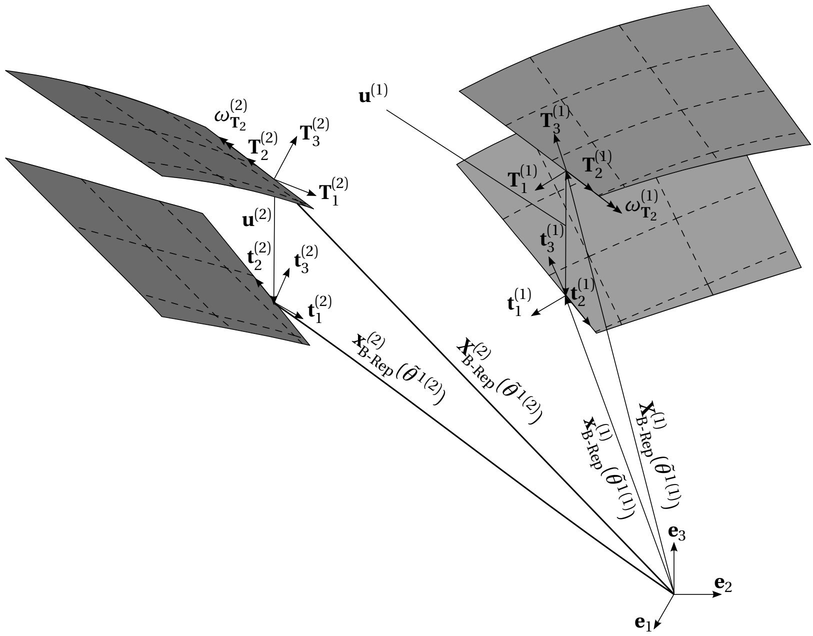


Figure 4.5: Internal boundary: for coupling two *parametric domains* two local Cartesian coordinate systems $\mathbf{T}^{(\alpha)}_i$ are used which are alinged on the boundary curve.

are given at the end of this section. The virtual work term for the force equilibrium within an internal boundary can then be expressed as

$$\delta W^{\text{disp}}_{\text{B-Rep}} = -\alpha_{\text{disp}} \int\limits_{\Gamma^{(1)}_{\text{c}}} \left(\mathbf{u}^{(1)}_{\text{B-Rep}} - \mathbf{u}^{(2)}_{\text{B-Rep}}\right)\left(\delta\mathbf{u}^{(1)}_{\text{B-Rep}} - \delta\mathbf{u}^{(2)}_{\text{B-Rep}}\right) \mathrm{d}\Gamma^{(1)}_{\text{c}}. \tag{4.23}$$

Thus, the components of the residual force vector and the tangential stiffness matrix are given by

$$R^{\text{B-Rep disp}}_{r} = \alpha_{\text{disp}} \int\limits_{\Gamma^{(1)}_{\text{c}}} \left(\mathbf{u}^{(1)} - \mathbf{u}^{(2)}\right)\left(\mathbf{u}^{(1)}_{,r} - \mathbf{u}^{(2)}_{,r}\right) \mathrm{d}\Gamma^{(1)}_{\text{c}}, \tag{4.24}$$

$$K_{rs}^{\text{B-Rep disp}} = \alpha_{\text{disp}} \int\limits_{\Gamma_{\text{c}}^{(1)}} \left(\mathbf{u}_{,s}^{(1)} - \mathbf{u}_{,s}^{(2)}\right) \cdot \left(\mathbf{u}_{,r}^{(1)} - \mathbf{u}_{,r}^{(2)}\right) \, \mathrm{d}\Gamma_{\text{c}}^{(1)}. \tag{4.25}$$

Applying a penalty approach, the moment along the boundary $m_{\mathbf{T}_2}$ is assumed to be

$$m_{\mathbf{T}_2} = -\alpha_{\text{rot}} (\omega_{\mathbf{T}_2}^{(1)} - \omega_{\mathbf{T}_2}^{(2)}), \tag{4.26}$$

where α_{rot} is the penalty factor which can be seen as a rotational spring stiffness and is selected manually as well (see also end of this section).

The virtual work term for the moment equilibrium within an internal boundary (see also Section 2.3.2) can be expressed as

$$\delta W_{\text{B-Rep}}^{\text{rot}} = -\alpha_{\text{rot}} \int\limits_{\Gamma_{\text{c}}^{(1)}} \left(\omega_{\mathbf{T}_2}^{(1)} \pm \omega_{\mathbf{T}_2}^{(2)}\right) \left(\delta \omega_{\mathbf{T}_2}^{(1)} \pm \delta \omega_{\mathbf{T}_2}^{(2)}\right) \, \mathrm{d}\Gamma_{\text{c}}^{(1)} \tag{4.27}$$

and its residual force vector and the tangential stiffness matrix components are

$$R_{r}^{\text{B-Rep rot}} = \alpha_{\text{rot}} \int\limits_{\Gamma_{\text{c}}^{(1)}} \left(\omega_{\mathbf{T}_2}^{(1)} \pm \omega_{\mathbf{T}_2}^{(2)}\right) \left(\omega_{\mathbf{T}_2,r}^{(1)} \pm \omega_{\mathbf{T}_2,r}^{(2)}\right) \, \mathrm{d}\Gamma_{\text{c}}^{(1)}, \tag{4.28}$$

$$\begin{aligned} K_{rs}^{\text{B-Rep rot}} = \alpha_{\text{rot}} \int\limits_{\Gamma_{\text{c}}^{(1)}} \Big[&\left(\omega_{\mathbf{T}_2,s}^{(1)} \pm \omega_{\mathbf{T}_2,s}^{(2)}\right) \left(\omega_{\mathbf{T}_2,r}^{(1)} \pm \omega_{\mathbf{T}_2,r}^{(2)}\right) + \\ &\left(\omega_{\mathbf{T}_2}^{(1)} \pm \omega_{\mathbf{T}_2}^{(2)}\right) \left(\omega_{\mathbf{T}_2,rs}^{(1)} \pm \omega_{\mathbf{T}_2,rs}^{(2)}\right) \Big] \, \mathrm{d}\Gamma_{\text{c}}^{(1)}. \end{aligned} \tag{4.29}$$

The first and second variations of the rotation $\omega_{\mathbf{T}_2}$ (Equation (4.5)) are given by

$$\omega_{\mathbf{T}_2,r} = \frac{\boldsymbol{\omega}_{,r} \cdot \mathbf{T}_2}{\sqrt{1 - (\boldsymbol{\omega} \cdot \mathbf{T}_2)^2}}, \tag{4.30}$$

$$\omega_{\mathbf{T}_2,rs} = \frac{\boldsymbol{\omega}_{,rs}\cdot\mathbf{T}_2}{\sqrt{1-(\boldsymbol{\omega}\cdot\mathbf{T}_2)^2}} + \frac{(\boldsymbol{\omega}_{,r}\cdot\mathbf{T}_2)(\boldsymbol{\omega}\cdot\mathbf{T}_2)(\boldsymbol{\omega}_{,s}\cdot\mathbf{T}_2)}{\left(1-(\boldsymbol{\omega}\cdot\mathbf{T}_2)^2\right)^{\frac{3}{2}}}, \tag{4.31}$$

where the variations of the rotation vector $\boldsymbol{\omega}$ (Equation (4.6)) and the displacement $\mathbf{w}$ (Equation (4.7)) are given by

$$\boldsymbol{\omega}_{,r} = \mathbf{T}_3 \times \mathbf{w}_{,r}, \qquad \boldsymbol{\omega}_{,rs} = \mathbf{T}_3 \times \mathbf{w}_{,rs}, \tag{4.32}$$

$$\mathbf{w}_{,r} = \mathbf{t}_{3,r}, \qquad \mathbf{w}_{,rs} = \mathbf{t}_{3,rs}. \tag{4.33}$$

The variations of $\mathbf{t}_3$ correspond to those of $\mathbf{a}_3$ and can be found in Kiendl [46].

Penalty factors

Since the penalty factor should be in the range of the usual stiffness characterized by the Young's modulus E, a relative penalty factor is used, which is defined as follows:

$$\alpha_{\text{rel}} = \frac{\alpha}{E} \tag{4.34}$$

A relative penalty factor in the range of $10^2 - 10^5$ usually gives good results (see also Section 7.5 and Breitenberger et al. [17]).

For patches with C^0 continuity inside the *parametric domain*, this B-Rep element formulation can also be used for preserving the relative angle within the domain as well. One simply needs to add parameter curves along the knots with C^0 continuities.

4.5.2 *Dirichlet* boundary conditions – penalty approach

The B-Rep element formulation for *Dirichlet* boundary conditions can be formally written in the same way, as the coupling from the previous section, with the exception that the displacements resp. rotations for reference boundary are set to zero. The virtual work contribution for an edge Γ_d can thus be written as follows:

$$\delta W^{\text{Dirichlet}}_{\text{B-Rep}} = \delta W^{\text{disp}}_{\text{B-Rep}} + \delta W^{\text{rot}}_{\text{B-Rep}}, \tag{4.35}$$

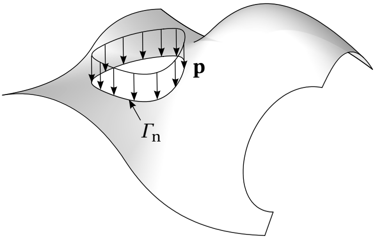


Figure 4.6: Example of a B-Rep edge load

where the virtual work term for constraining the displacements is given by

$$\delta W_{\text{B-Rep}}^{\text{disp}} = -\alpha_{\text{disp}} \int_{\Gamma_{\mathrm{d}}} \left(\mathbf{u}_{\text{B-Rep}}^{(1)}\right) \cdot \left(\delta \mathbf{u}_{\text{B-Rep}}^{(1)}\right) \, \mathrm{d}\Gamma_{\mathrm{d}}, \tag{4.36}$$

and the term for constraining the rotations is given by

$$\delta W_{\text{B-Rep}}^{\text{rot}} = -\alpha_{\text{rot}} \int_{\Gamma_{\mathrm{d}}} \left(\omega_{\mathbf{T}_2}^{(1)}\right) \left(\delta \omega_{\mathbf{T}_2}^{(1)}\right) \, \mathrm{d}\Gamma_{\mathrm{d}}. \tag{4.37}$$

The corresponding variations can be computed in a straight forward manner, as shown in Section 4.5.1.

4.5.3 *Neumann* boundary conditions

A *B-Rep edge element* as an *integration domain* can also be used for *Neumann* boundary conditions. The virtual work contribution of a load **p** on an edge Γ_{n} can be computed as follows:

$$\delta W_{\text{B-Rep}}^{\text{Edge load}} = \int_{\Gamma_{\mathrm{n}}} \mathbf{p} \, \mathrm{d}\Gamma_{\mathrm{n}}. \tag{4.38}$$

The corresponding residual force vector and possibly stiffness matrix (see also Section 4.2) in the case of deformation dependent loads can be derived straight forward. The numerical integration is explained in Section 5.4.1.

4.6 *B-Rep vertex* element formulations

Element formulations for *vertex* elements can be adapted from *B-Rep edge elements* like e.g. Section 4.5.1 by using a *Dirac* function within the integral resp. by just evaluating the basis functions of the surface at the *vertex* location.

Summary and conclusion of Chapter 4

This chapter explains the *isogeometric B-Rep analysis* (IBRA) which allows the usage of NURBS-based B-Rep (CAD) models for structural shell analysis. The newly developed *B-Rep elements* can be used to couple discontinuous and trimmed geometries with gaps and overlaps (see also Section 7.5). A corresponding element formulation based on a penalty approach is explained as well. Moreover a data structure for *isogeometric (trimmed) elements* and *B-Rep elements* is proposed.

With IBRA a new method has been developed which provides the framework for analyzing CAD models without any modifications considering all the aspects mentioned in Chapter 2. The implementation of IBRA and its integration into a CAD system is explained in the following chapters.

I have not failed. I've just found 10,000 ways that won't work.

Thomas A. Edison

CHAPTER 5

Implementation of the isogeometric B-Rep analysis

This chapter explains the systematic disassembly of *NURBS-based B-Rep models* (see Section 2.2) into finite elements resp. *integration domains* and defines clear data interfaces for an implementation of the *isogeometric B-Rep shell analysis* into existing FE-solvers. In addition, a new adaptive quadrature scheme for the numerical integration of trimmed domains is introduced. The new procedure is named *adaptive Gaussian integration procedure* and is abbreviated with AGIP. Moreover special aspects of IBRA like local refinement, flying nodes, and *clipping* are discussed.

5.1 Overview

Figure 5.1 gives an overview of the required steps for analyzing CAD models directly with the *isogeometric B-Rep analysis* and using an existing FE solver.

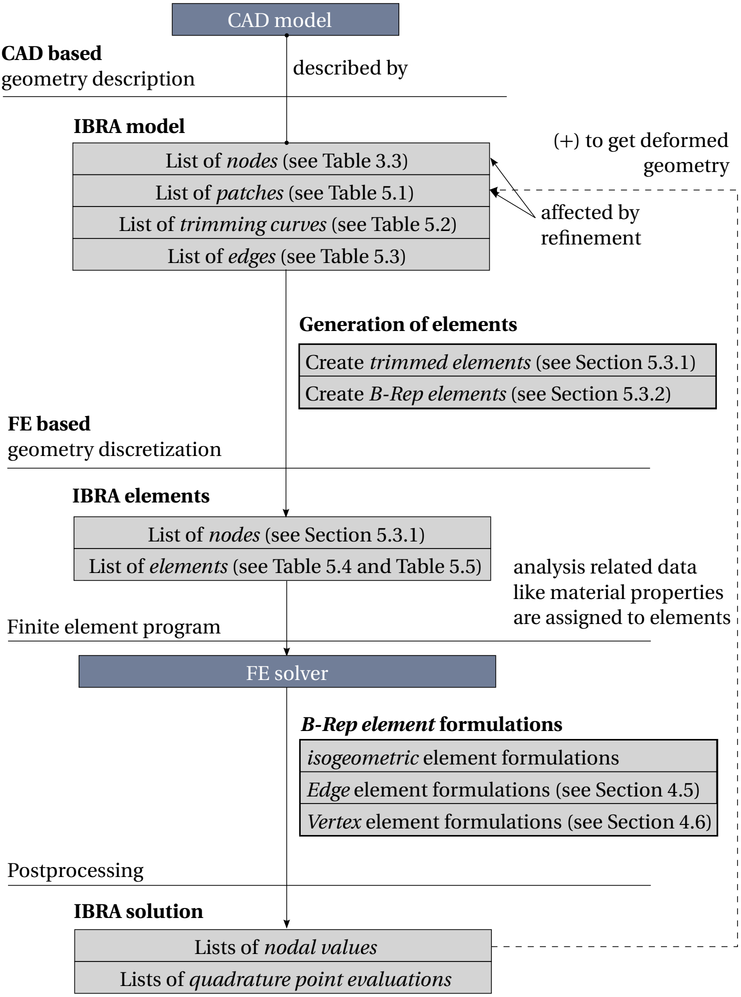

CAD model
CAD based
geometry description
described by
IBRA model
List of *nodes* (see Table 3.3)
List of *patches* (see Table 5.1)
List of *trimming curves* (see Table 5.2)
List of *edges* (see Table 5.3)
(+) to get deformed geometry
affected by refinement
Generation of elements
Create *trimmed elements* (see Section 5.3.1)
Create *B-Rep elements* (see Section 5.3.2)
FE based
geometry discretization
IBRA elements
List of *nodes* (see Section 5.3.1)
List of *elements* (see Table 5.4 and Table 5.5)
analysis related data like material properties are assigned to elements
Finite element program
FE solver
***B-Rep element* formulations**
isogeometric element formulations
Edge element formulations (see Section 4.5)
Vertex element formulations (see Section 4.6)
Postprocessing
IBRA solution
Lists of *nodal values*
Lists of *quadrature point evaluations*

Figure 5.1: Overview of the components which are involved for analyzing CAD models directly and using an existing FE solver.

Patches (B-Rep faces)	Description
NurbsID=1561	id of NURBS surface
p=2	polynomial degree in ξ-direction
q=2	polynomial degree in η-direction
U=[0,0,0, ... 1,1,1]	knot vector $\mathbf{\Xi}$
V=[-1,-1,-1, ... 5,5,5]	knot vector $\mathbf{H}$
NodeIDs=[101,87,12-16,...,12]	list of node ids
FaceNormal=true	flag used for face orientation
CurveIDs=[23,54,23,...,13]	list of trimming curve ids
NurbsID=243	id of NURBS surface
...	...

Table 5.1: Exemplary list of *B-Rep faces* resp. *patches* with their description of NURBS (see also Table 3.4) and their visible domains defined by trimming curves. The flag *FaceNormal* indicates whether the *face* has the same orientation as the surface normal.

5.2 Data structure of IBRA model

For describing a NURBS-based B-Rep surface model within IBRA (see also Section 2.2) the following data are required:

- a list of *nodes* (e.g.Table 3.3)
- a list of *B-Rep faces* resp. *patches* (e.g. Table 5.1),
- a list of *trimming curves* (e.g. Table 5.2), and
- a list of *B-Rep edges* (e.g. Table 5.3).

List of B-Rep faces

Table 5.1 shows exemplarily a list of *B-Rep faces* resp. *patches* which can be used for IBRA. The table extends the content of a *patch* in Table 3.4 by a list of trimming curve ids (see *CurveIDs*) and a flag (see *FaceNormal*). The trimming curves define the visible domain (see also Section 2.2.3 and

Trimming curves	Description
NurbsID=23	id of NURBS trimming curve
p=2	polynomial degree
U=[0,0,0, ... 1,1,1]	knot vector Ξ
Range=[0, 1]	defines active range of knot vector
Ctrlpts (ξ,η,w)	list of control points
3.445 2.234 1.00	
3.179 1.954 1.00	
... ...	
NurbsID=24	id of NURBS trimming curve
...	...

Table 5.2: Exemplary list of *trimming curves* with their description of NURBS (see also Section 2.2.3). Note that these curves lay within the parameter spaces of surfaces. A control point, therefore, requires just two coordinates (ξ and η) and its weight w.

Figure 2.16) of the *patch* (see also Section 2.2.3 and Figure 2.16). The additional flag *FaceNormal* $\in$ {TRUE, FALSE} indicates whether the orientation of the *B-Rep face* corresponds to the one of the surface normal (see also Section 4.3.1).

List of trimming curves

Table 5.2 contains a list of all trimming curves which are used to describe the *trimmed domains* of the *patches* in Table 5.1. Note that IBRA requires only the trimming curves within the parameter space. For an efficient implementation it might be useful to assign the geometrical description of trimming curves directly to the NURBS surfaces.

List of B-Rep edges

Last but not least the topology of *patches* needs to be described by the *edges*. Table 5.3 shows exemplarily a description of a free and an internal edge which connects two adjacent *faces* resp. *patches* (see *NurbsIDs*). Within IBRA thus an *edge* is described by one or a pair of trimming curve segments (see *CurveIDs*) bounded by curve parameters (see *Range*). Since two curve segments belonging to the same edge are just approximated, gaps between the curves can occur (see Section 2.2.3). The tolerance describes the maximal distance (see *Tol*) between two space curve segments and can be used to check the desired quality of the model. Usually, this tolerance is provided by the CAD model resp. system.

Patches	Description
EdgeID=23	id of the edge
NurbsID=165	id of the patch
CurveID=11	id of the trimming curve (master)
Range=[0, 0.5]	active range of master curve which defines the edge
EdgeID=24	id of the edge
master	
NurbsID=165	id of the patch (master)
CurveID=11	id of the trimming curve (master)
Range=[0, 0.5]	active range of master curve which defines the edge
slave	
NurbsID=32	id of the patch (slave)
CurveID=12	id of the trimming curve (slave)
Range=[0, 0.9]	active range of slave curve which defines the edge
Tol=0.001	absolute coupling tolerance

Table 5.3: Exemplary list of *B-Rep edges* which define the overall topology of the geometry model composed of the *patches* in Table 5.1

5.3 Generation of elements

This section describes how one can straight forward derive a set of finite elements from the data structure described in Section 5.2. These elements can then be used to solve Equation (4.4) for NURBS-based B-Rep models in a finite-element-like manner.

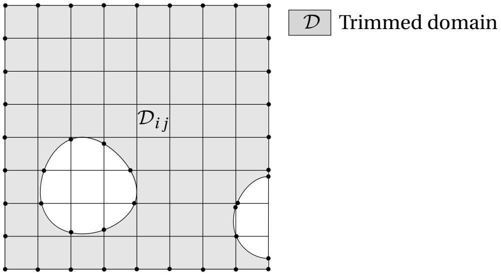


Figure 5.2: The intersection points of trimming curves and (refined) knot lines are determined as curve parameters for each curve.

5.3.1 Isogeometric (trimmed) elements

For creating *isogeometric (trimmed) elements* (see Section 4.3.1), all intersections between each trimming curve and the knot lines of the corresponding refined *patch*, need to be determined (see also Figure 5.2) as parameters of the trimming curve. For sake of simplicity the remainder of this thesis terms such curve parameters as *intersection parameters*. The latter can then be used to define (trimmed) curve segments (see also Section 2.2.3) lying within one knot span $\mathcal{H}_{ij}$ (see also Equation (3.18)). Following a systematic disassembling procedure these curve segments are assigned to the corresponding knot span.

After the identification of all trimmed knot spans and considering the curve directions (see also Figure 2.16) one can assign to each knot span $\mathcal{H}_{ij}$ one of the indicators given in Table 4.1. The assigned indicators allow to create a list of *active* elements (see also Section 4.3.1) considering the trimming curve segments for trimmed elements.

Optimized element description

Isogeometric elements as *parameter subdomains* are not well suited for visualizing IBRA results e.g. deformed geometry because within CAD systems NURBS are handled *patch*-wise. Thus *isogeometric (trimmed) elements* are purely used for numerical integration. This means that one can replace the

Element (i,j)	Description
id=2345	id of the element
p=2	polynomial degree in ξ-direction
q=2	polynomial degree in η-direction
U=$[\xi_{i-p},\xi_{i+p+1}]$	knot vector $\boldsymbol{\Xi}$
V=$[\eta_{j-q},\eta_{j+q+1}]$	knot vector $\mathbf{H}$
NodeIDs=[54,56,...,78,89]	list of element node ids
FaceNormal=true	flag used for face orientation
Quadrature points id (ξ_k,η_k) $\tilde{w}$	list of quadrature points with weighting factors
12 (0.342, 0.433) 0.324	
13 (0.434, 0.656) 0.333	
...	...

Table 5.4: Example of an isogeometric element description which is optimized for numerical integration. This description can be used for untrimmed and trimmed elements.

trimming curves by a set of quadrature points which consider the *trimmed domain* $\mathcal{D}_{ij}$ of the element Ω_{ij}.

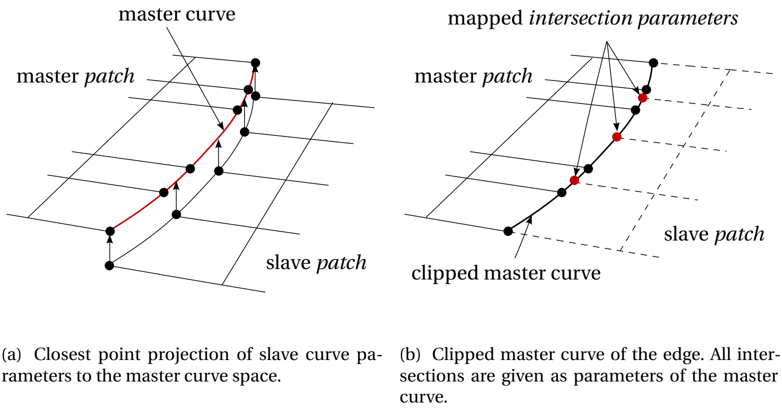
Therefore the element description in Table 4.2 is replaced by the one in Table 5.4. Having just a list of quadrature points with

- global ids (used for postprocessing e.g. stress evaluations),
- coordinates of the quadrature points expressed as surface parameters ξ and η, and
- quadrature weighting factors $\tilde{w}$ (used for numerical integration)

allows to treat trimmed and untrimmed elements in exactly the same way. Section 5.4 explains the determination of the locations and weighting factors of the quadrature points.

5.3.2 B-Rep elements

Having all *intersection parameters*, the *B-Rep edge elements* can be derived straight forward from the active range of the trimming curve belonging to



(a) Closest point projection of slave curve parameters to the master curve space.

(b) Clipped master curve of the edge. All intersections are given as parameters of the master curve.

Figure 5.3: Clipping operation using a closest point projection

an isogeometric trimmed element. Note, that the creation of B-Rep coupling elements from Table 5.3 requires an additional step, called *clipping*.

Clipping

Clipping is required for an accurate integration along an edge between two *non-conforming* patches. It describes the step of mapping *intersection parameters* from the slave curve to the *parameter space* of the master curve using a *closest point projection* (*Newton-Raphson* scheme is used within this thesis) within the geometry space. These mapped parameters are named *mapped intersection parameters*. Figure 5.3 illustrates *clipping* for a simple example. Having also the *mapped intersection parameters* for the master curve one can straight forward derive *trimmed non-overlapping B-Rep edge coupling elements*. Table 4.4 contains the description of such elements not yet optimized for numerical integration.

Optimized element description

An exemplary element description of a B-Rep coupling element optimized for numerical integration is given in Table 5.5. It requires just the coordinates of the evaluation points on the master and slave patch and the corresponding tangent vectors $\tilde{\mathbf{T}}$ of the trimming curves (see also Equation (2.32)). A pair of evaluation points is computed by using a *closest point projection* (*Newton-Raphson* scheme is used within this thesis). The basis functions used by the evaluation points are provided by the isogeometric elements where the evaluations points are located in (see *ElementIDs*). Moreover, a global id and a *weighting factor* is assigned to the evaluation points on the master side for a later stress evaluation and for numerical integration along the B-Rep edge (see Section 5.4.1).

B-Rep edge element	Description
id=433	id of the B-Rep element
master	
ElementID=11	id of patch (master)
Quadrature points id (ξ_k, η_k) $\tilde{\mathbf{T}}$ $\tilde{w}$	list of quadrature points
34 $(0.32, 0.43)$ $[0.0, 0.1]$ 0.32	
35 $(0.32, 0.45)$ $[0.0, 0.1]$ 0.33	
...	...
slave	
ElementID=12	id of patch (slave)
Slave points (ξ_k, η_k) $\tilde{\mathbf{T}}$	list of evaluation points
$(0.32, 0.55)$ $[0.1, 0.0]$	
$(0.33, 0.55)$ $[0.1, 0.0]$	
...	...

Table 5.5: Example of the description of a trimmed *non-overlapping* B-Rep edge coupling element optimized for numerical integration.

5.4 Numerical integration of elements

This section deals with numerical integration of *B-Rep edge* and isogeometric (trimmed) surface elements.

5.4.1 *B-Rep edge* elements

For the numerical integration of *B-Rep edge elements* given in Table 4.3 and Table 4.4 a *Gaussian* quadrature scheme is applied with $n_g = \max\{p, q\} + 1$ quadrature points where p and q are the highest polynomial degrees occurring within the master and slave *patches*, respectively. The length $|\mathit{\Gamma}_\mathrm{e}|$ of a *B-Rep edge element* can be computed as

$$|\mathit{\Gamma}_\mathrm{e}| = \int_{\mathit{\Gamma}_\mathrm{e}} d\mathit{\Gamma}_\mathrm{e} = \int_{\tilde{\theta}^1} \tilde{J}_1 \, \mathrm{d}\tilde{\theta}^1 = \int_{\mathcal{G}} \tilde{J}_1 \tilde{J}_2 \, \mathrm{d}\mathcal{G}, \tag{5.1}$$

where $\mathit{\Gamma}_\mathrm{e}$ is the boundary segment of the B-Rep element and $\tilde{J}_1$ the mapping from the geometry space to the parameter space

$$\tilde{J}_1 = \left\| \frac{d\mathbf{X}_\mathrm{surf}}{d\tilde{\theta}^1} \right\|_2 = \left\| \breve{\mathbf{T}}_2 \right\|_2 = \left\| \mathbf{A}_1 \frac{\partial \theta^1}{\partial \tilde{\theta}^1} + \mathbf{A}_2 \frac{\partial \theta^2}{\partial \tilde{\theta}^1} \right\|_2 . \tag{5.2}$$

$\tilde{J}_2$ is given as the mapping from the parameter space of the trimming curve to the *Gaussian domain* $\mathcal{G} = [-1, 1]$

$$\tilde{J}_2 = \frac{\partial \tilde{\xi}}{\partial \xi_\mathrm{G}}, \tag{5.3}$$

with ξ_G being the parameter of the *Gaussian space*. Introducing the *weighting factor* $\tilde{w}_l$

$$\tilde{w}_l = \tilde{J}_2(\tilde{\xi}_l) w_l, \tag{5.4}$$

with w_l being the *Gaussian* quadrature weights, the numerical integration of a *B-Rep edge element* can be computed as

$$|\mathit{\Gamma}_\mathrm{e}| \approx \sum_{l=1}^{n_g} \tilde{J}_1(\tilde{\xi}_l) \, \tilde{w}_l. \tag{5.5}$$

5.4.2 Isogeometric (trimmed) elements

Since a trimmed domain $\mathcal{D}_{ij}$ of an element (see also Section 5.3.1) is decoupled from the underlying surface parametrization a *Gaussian* quadrature scheme can not directly be used as it is the case for untrimmed surfaces (see also Hughes et al. [40]).

Existing integration approaches

For the numerical integration of trimmed domains in the context of structural analysis, several techniques have been proposed. Kim et al. [47] use NURBS-enhanced triangles, Breitenberger et al. [17] use NURBS surfaces, while Wang et al. [92] use a cell based approach for parametrizing the *trimmed domain* of a trimmed element within the parameter space. Other approaches are the quad- resp. octree techniques or the smart octree approach developed by Kudela et al. [51, 52]. The latter mentioned techniques are mainly used for the *finite cell method* (FCM)(see Parvizian et al. [61]). Moreover, Nagy et al. [58] introduced an integration technique for determining a set of quadrature points and their weights satisfy the moment fitting equations up to a predefined tolerance. However it requires a non-trivial optimization process for determining the number and locations of the quadrature points as well as their weights for trimmed domains approximated by polygons.

An alternative approach to the integration within parameter space is presented in Schmidt [78], which makes use of the space curves of the trimmed domain.

A general answer to the question of the most robust and efficient integration scheme is very difficult because it strongly depends on the shape of the *trimmed domain* resp. surface and the required accuracy.

New integration approach

Within this thesis an adaptive quadrature scheme termed *adaptive Gaussian integration procedure* (AGIP) (see Section 5.5) is used. The application of AGIP allows to unify the numerical integration of trimmed and untrimmed elements and is very accurate (see also Section 7.2). AGIP only requires the fulfillment of two conditions for each *integration domain* (see next paragraph) and allows for a straight forward determination of the quadrature points (locations and weights). In case the required conditions for AGIP are violated the *integration domain* is subdivided until each *integration domain* is AGIP conform. AGIP allows the usage of the optimized element description given in Table 5.4. The computation of stiffness matrix and residual force vector of all elements can thus be treated in exactly the same way.

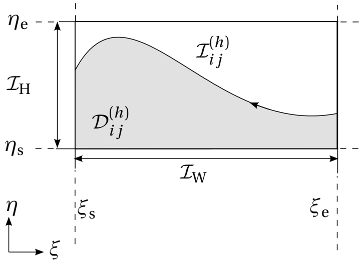


Figure 5.4: Example of an AGIP conform *integration domain* described by its bounding box (ξ_s, ξ_e, η_s, and η_e) within the parameter space of a surface.

Definition of AGIP conform integration domains

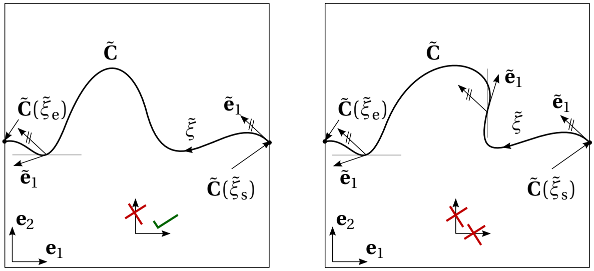
Within this thesis an *integration domain* $\mathcal{I}_{ij}^{(h)}$ as it is used in Equation (4.17) can be described by a bounding box which at its turn is described by four surface parameters (see ξ_s, ξ_e, η_s, and η_e). An example of an *integration domain* is illustrated in Figure 5.4. The bounding box defines the size and position of the *integration domain* within the parameter space of the surface. For applying AGIP (see Section 5.5) the following two conditions must be fulfilled for the *integration domain*:

- the *integration domain* must not have more than one assigned decoupled and not closed trimming curve segment $\tilde{\mathbf{C}}_{(\tilde{\xi})}$ defined by $\tilde{\xi} \in [\tilde{\xi}_s, \tilde{\xi}_e]$ (see also Section 5.3.1),
- the curve segment must not have undercuts at least for one of the two directions ξ and η.

The latter mentioned condition can mathematically be expressed by fulfilling at least one of the following conditions

$$\begin{aligned} &\operatorname{sign}(\tilde{\mathbf{e}}_1(\tilde{\xi}) \cdot \mathbf{e}_1) = \operatorname{sign}(\tilde{\mathbf{e}}_1(\tilde{\xi}_s) \cdot \mathbf{e}_1) \quad \forall \tilde{\xi} \in [\tilde{\xi}_s, \tilde{\xi}_e] \quad (1) \quad \text{and/or} \\ &\operatorname{sign}(\tilde{\mathbf{e}}_1(\tilde{\xi}) \cdot \mathbf{e}_2) = \operatorname{sign}(\tilde{\mathbf{e}}_1(\tilde{\xi}_s) \cdot \mathbf{e}_2) \quad \forall \tilde{\xi} \in [\tilde{\xi}_s, \tilde{\xi}_e] \quad (2), \end{aligned} \tag{5.6}$$

where $\mathbf{e}_\alpha$ are the unit base vectors aligned to the parameterization of the surface and $\tilde{\mathbf{e}}_\alpha$ are the unit base vectors aligned to the trimming curve (see



(a) **AGIP conform**: No undercuts in the direction of $\mathbf{e}_1$ and thus condition (1) in Equation (5.6) is fulfilled whereas condition (2) is violated.

(b) **Non AGIP conform**: Undercuts in both directions and thus both, conditions (1) and (2) in Equation (5.6) are violated.

Figure 5.5: Comparison of two different *integration domains* with one curve segment.

Section 2.3.3). Figure 5.5 shows exemplarily a comparison of an invalid (not AGIP conform) and valid (AGIP conform) *integration domain*.

In case an *integration domain* contains more than one disconnected curve segment and/or both conditions in Equation (5.6) are violated the *integration domain* is bisected.

Bisecting integration domains

In case an *integration domain* is not AGIP conform it requires a decomposition. There are many ways of decomposing such domains. One can for example halve each subdomain until all subdomains fulfill the requirements or using the bounding boxes of each curve segment as basis for bisectioning or a combination of it. To keep the development simple within this thesis a halving strategy is used.

Description of integration domains

For the sake of implementation it is advantageous to save the *integration domains* within a *binary tree data structure*. Table 5.6 describes exemplarily the data structure of an *integration domain* which is used within this thesis. Each *integration domain* consists of a bounding box (*RangeU* and *RangeV*) which indicates the size and position of the *integration domain* within the parameter space of the surface, a list of curve segments, an indicator from Table 5.6, and two links to possible *child domains*.

Integration domain	Description
RangeU=[0.1, 0.2]	surface parameters ξ_s and ξ_e
RangeV=[-0.05, 0.1]	surface parameters η_s and η_e
Indicator=bs	indicators from Table 5.7
List of curve segments	
Curve 1	
CurveID=11	id of trimming curve
Range=[0, 0.5]	active range of curve segment (*interaction parameters*)
Curve 2	
CurveID=13	id of trimming curve
Range=[0, 0.4]	active range of curve segment (*interaction parameters*)
...	...
ChildDomain1=2	id of first child domain
ChildDomain2=3	id of second child domain

Table 5.6: Example of a description of an *integration domain*

The indicator *bs* is used to describe non AGIP conform domains which require a bisectioning. The indicator *in* and *out* label domains which are completely inside resp. outside the trimmed domain. The indicators *st* is used for domains with a negligible visible part and *trim* is used for all trimmed *integration domains* which are AGIP conform. Note that bisectioning of *integration domains* is rarely required for appropriate refined surfaces for analysis purposes.

5.5 Adaptive Gaussian integration procedure (AGIP)

The *adaptive Gaussian integration procedure* (AGIP) can be used for integrating the *trimmed domain* within an AGIP conform *integration domain* (see Section 5.4.2) and thus it can also be used for computing stiffness

Indicators	Description
mother indicator	
bs	domain is split into two domains
child indicators	
in	completely inside the trimmed domain
trim	AGIP conform domain
st	domain with negligible contribution to trimmed domain
out	completely outside the trimmed domain

Table 5.7: Indicators used for *integration domains*

matrices and residual force vectors for *isogeometric (trimmed) elements* (see Section 4.3.1 and Section 3.5.3).

The idea of AGIP is to handle the trimming of *integration domains* within the *Gaussian space*. Thus the curve segment of a *trimmed integration domain* is mapped (shifting, scaling, and rotating) into the *Gaussian space* such that only the height $\mathcal{G}_\mathrm{H}$ of the *adaptive Gaussian domain* $\mathcal{G}^\mathrm{A}$ varies w.r.t. ξ_G (see Figure 5.6). For an AGIP conform *integration domain* and allowing a rotation mapping $(0°, 90°, 180°, \text{or } 270°)$ this can always be done.

As an example the area $|\Omega_{ij}^{(h)}|$ of an *isogeometric (trimmed) element segment* $\Omega_{ij}^{(h)} \subset \Omega_{ij}$ defined by the *trimmed domain* $\mathcal{D}_{ij}^{(h)} \subset \mathcal{I}_{ij}^{(h)}$ (see also Equation (4.18)) can be computed as

$$|\Omega_{ij}^{(h)}| = \int_{\Omega_{ij}^{(h)}} d\Omega_{ij}^{(h)} = \int_{\mathcal{G}^\mathrm{A}} J_1\, J_2\, d\mathcal{G}^\mathrm{A}, \tag{5.7}$$

where $\mathcal{G}^\mathrm{A}$ is the *adaptive Gaussian domain* (see Figure 5.6). The Jacobian J_1 indicates the mapping from the geometry to the parameter space and is expressed as

$$J_1 = \left\| \mathbf{A}_1 \times \mathbf{A}_2 \right\|_2 . \tag{5.8}$$

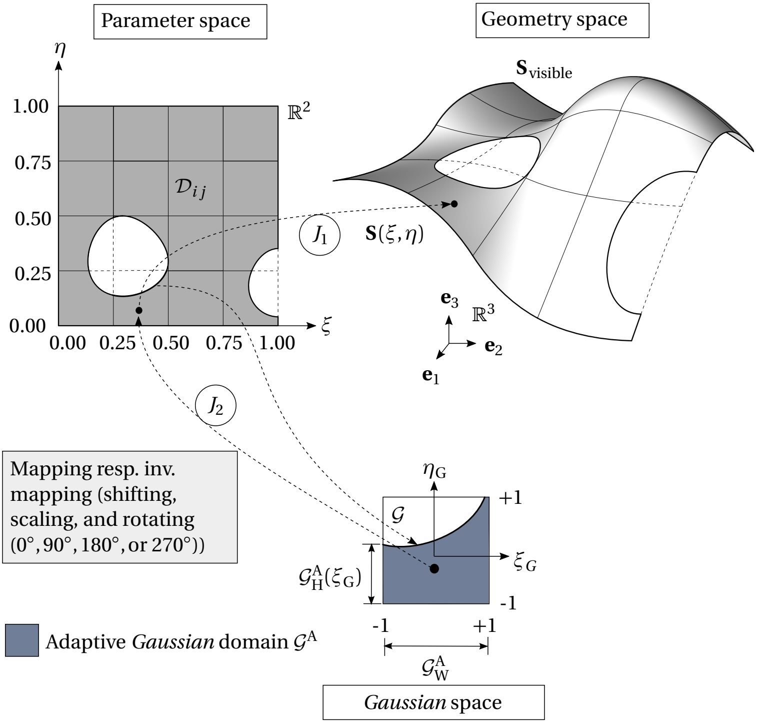


Figure 5.6: Overview of the different spaces relevant for AGIP integration for trimmed surfaces

Here $\mathbf{A}_\alpha$ are the base vectors as defined in Section 2.3.1. The Jacobian J_2 indicates the linear mapping (scaling and rotating) from the parameter space to the *Gaussian* space $\mathcal{G} = [-1,1] \times [-1,1]$ and is given by

$$J_2 = \left| \frac{\partial \xi}{\partial \xi_G} \frac{\partial \eta}{\partial \eta_G} \right|, \tag{5.9}$$

for a rotation of 0° and 180° or

$$J_2 = \left| \frac{\partial \xi}{\partial \eta_G} \frac{\partial \eta}{\partial \xi_G} \right|, \tag{5.10}$$

for a rotation of 90° and 270° with ξ_{G} and η_{G} being the parameters of the *Gaussian* space. Using n_g quadrature points, the area of an *isogeometric (trimmed) element segment* $\Omega_{ij}^{(h)} \subset \Omega_{ij}$ can be computed as

$$|\Omega_{ij}^{(h)}| \approx \sum_{l=1}^{n_g} J_1(\hat{\xi}_l^{\mathrm{A}}, \hat{\eta}_l^{\mathrm{A}})\, J_2(\hat{\xi}_l^{\mathrm{A}}, \hat{\eta}_l^{\mathrm{A}})\, w_l^{\mathrm{A}}, \tag{5.11}$$

where $(\hat{\xi}_l^{\mathrm{A}}, \hat{\eta}_l^{\mathrm{A}})$ are the coordinates obtained by an appropriate rotation $(0°, 90°, 180°, \text{or } 270°)$ of the coordinates $(\xi_l^{\mathrm{A}}, \eta_l^{\mathrm{A}})$ of the *adaptive Gaussian quadrature points* determined within the *Gaussian* space. w_l^{A} are the corresponding *adaptive quadrature weights*. The determination of the *adaptive Gaussian quadrature points* is explained in the following two sections.

For sake of implementation and simplicity Equation (5.11) is rewritten as

$$|\Omega_{ij}^{(h)}| \approx \sum_{l=1}^{n_g} J_1\, \tilde{w}_l, \tag{5.12}$$

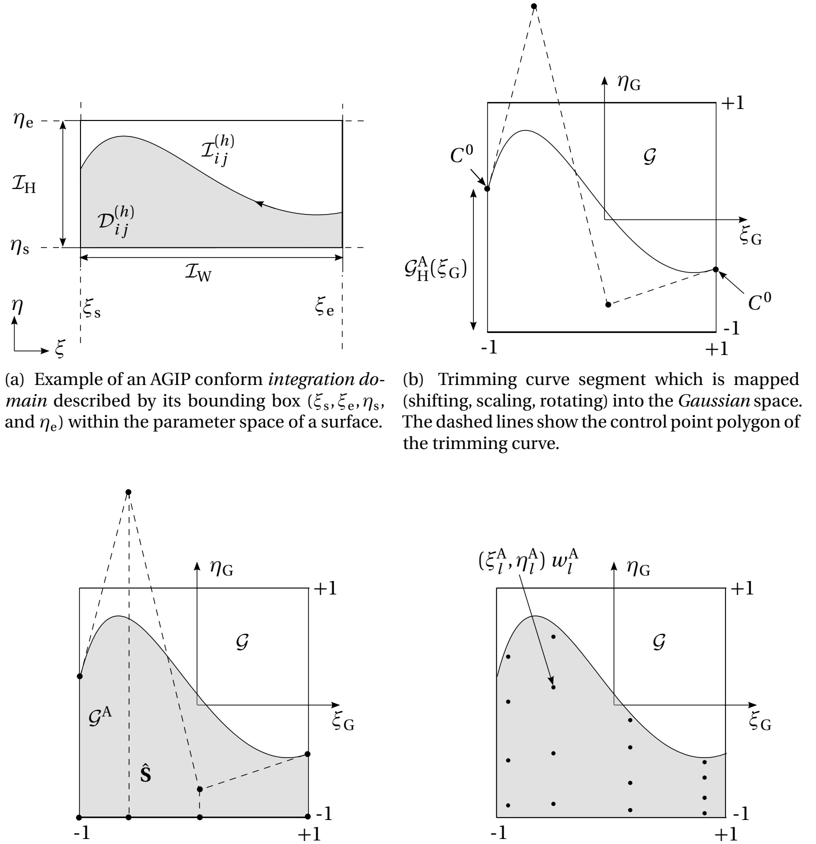
where the newly introduced *weighting factor* $\tilde{w}_l$ is given by

$$\tilde{w}_l = J_2\, w_l^{\mathrm{A}}. \tag{5.13}$$

The *weighting factor* $\tilde{w}_l$ remains constant during analysis and thus it must be computed just once. The *adaptive Gaussian quadrature points* are mapped (rotating, scaling, and shifting) to the parameter space of the surface and can then be used for the element description in Table 5.4.

5.5.1 Full Gaussian domain

In the special case $\mathcal{G}_{\mathrm{H}}$ is constant and equal to 2, i.e. the area of $\mathcal{G}^{\mathrm{A}} = 4$ and no rotation is performed, a full integration of the *Gaussian* domain $\mathcal{G} = [-1,1] \times [-1,1]$ is performed (classical *Gaussian integration*). This is the case for *isogeometric untrimmed elements* (see Section 3.5.3). For a more efficient numerical integration a reduced integration scheme analog to the one presented in Dornisch et al. [31] can be used for full elements lying completely inside the trimmed domain (indicator *in* in Table 4.1). Within this thesis only the integration scheme proposed in Hughes et al. [40] ($p+1$ integration points per direction) is used.



(a) Example of an AGIP conform *integration domain* described by its bounding box ($\xi_\mathrm{s}, \xi_\mathrm{e}, \eta_\mathrm{s}$, and η_e) within the parameter space of a surface.

(b) Trimming curve segment which is mapped (shifting, scaling, rotating) into the *Gaussian* space. The dashed lines show the control point polygon of the trimming curve.

(c) The trimming curve is used for creating a surface $\hat{\mathbf{S}}$ which parametrizes the *adaptive Gaussian domain* $\mathcal{G}^\mathrm{A}$. The dashed lines show the control point net of the surface.

(d) The locations $(\xi_l^\mathrm{A}, \eta_l^\mathrm{A})$ and weights (w_l^A) of the *adaptive quadrature points* can be determined by applying the integration scheme proposed in Hughes et al. [40] to the untrimmed surface $\hat{\mathbf{S}}$. The highest occurring polynomial degree (here $p = 3$ for the trimming curve and $p = q = 3$ for the surface to be integrated with AGIP) is used as reference for the integration scheme. Thus here 4x4 quadrature points are used as integration scheme for the non-zero knot span of the untrimmed surface $\hat{\mathbf{S}}$.

Figure 5.7: Procedure for determining the locations and *weights* for the adaptive quadrature points.

5.5.2 Trimmed Gaussian domain

In the case of *trimmed Gaussian domains*, i.e. the area $|\mathcal{G}^{\mathrm{A}}| < 4$, the *adaptive Gaussian quadrature points* $(\xi_l^{\mathrm{A}}, \eta_l^{\mathrm{A}})$ and the corresponding *quadrature weights* w_l need to be determined such that

- all quadrature points lie within the *adaptive Gaussian domain*
- the moment fitting equations are fulfilled
- the sum of all *adaptive Gaussian quadrature weights* w_l^{A} corresponds to the area of $\mathcal{G}^{\mathrm{A}}$ i.e. $\sum_{l=1}^{n_g} w_l^{\mathrm{A}} = |\mathcal{G}^{\mathrm{A}}|$

An efficient way to obtain an appropriate set of such quadrature points is to parametrize $\mathcal{G}^{\mathrm{A}}$ with an untrimmed NURBS surface $\hat{\mathbf{S}}$ using the trimming curve segment which is mapped for each trimmed *integration domain* into the *Gaussian space* (see also Figure 5.7). Note that C^0 continuities needs to be generated at the boundaries of the trimming curve segment (see also Figure 2.5) if not already given. The surface $\hat{\mathbf{S}}$ can then be used to determine the adaptive quadrature points as it is explained in the following paragraph.

Determination of the adaptive quadrature points

Using the integration scheme proposed by Hughes et al. [40] for integrating the untrimmed surface $\hat{\mathbf{S}}$, a set of quadrature points $\mathcal{A}$ can be determined. Within this thesis the highest occurring polynomial degree of the trimming curve and surface, which is to be integrated with AGIP, is used as reference for the integration scheme. Having these quadrature points the coordinates $(\xi_l^{\mathrm{A}}, \eta_l^{\mathrm{A}})$ of the *adaptive quadrature points* can be computed as

$$(\xi_l^{\mathrm{A}}, \eta_l^{\mathrm{A}}) = \hat{\mathbf{S}}(\xi_l, \eta_l) \quad \text{with} \quad (\xi_l, \eta_l) \in \mathcal{A}. \tag{5.14}$$

The corresponding *adaptive quadrature weights* w_l^{A} are given by

$$w_l^{\mathrm{A}} = \tilde{J}_1(\xi_l, \eta_l)\, \tilde{J}_2(\xi_l, \eta_l)\, w_l, \tag{5.15}$$

where w_l are the Gauss quadrature weights used for integrating the surfaces $\hat{\mathbf{S}}$ and $\tilde{J}_1$ resp. $\tilde{J}_2$ represent the mapping analogously to Equation (5.8) resp. Equation (5.9).

The example in Section 7.2 underlines the accuracy of AGIP in the context of IBRA.

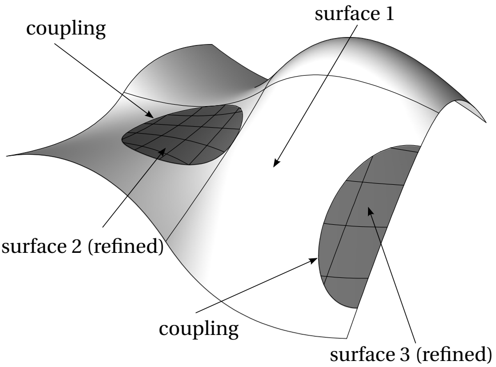


Figure 5.8: Local refinement within IBRA. After trimming the surface, the desired regions are refined and the boundaries are coupled.

5.6 Special aspects

This section contains some aspects which are different for IBRA in contrast to classical FEA. Thus they are briefly explained.

5.6.1 Local refinement

By considering coupling boundary conditions (see Section 4.5.1) within IBRA, quasi arbitrary local refinement can be realized easily. One simply needs to split the surfaces into different pieces, refine the desired trimmed regions and couple the boundaries. An example of local refinement within IBRA is shown in Figure 5.8. Note that this local refinement differs from those presented in previous publications like Kuru et al. [53], Schillinger [75], Scott et al. [79], and Vuong et al. [90].

5.6.2 Flying nodes

In IBRA, control points with very small influence on the *trimmed domain* are termed *flying nodes*, because it is possible that the displacements of these nodes change during nonlinear analysis. Nevertheless, the solution converges owing to the negligible contribution of the flying nodes on the *trimmed domain* resp. solution.

These flying nodes are closely related to the control points in the *fictitious domain* (FCM), which also make a minor contribution to the solution. According to Schillinger [75], these points should "…'move' as freely as possible to permit a smooth extension of the physical solution into the fictitious domain". This rationale can be applied to the flying nodes as well. For structural problems, the small contribution of the flying nodes has no influence on the solution. It only influences the conditioning of the linear system of equations which needs to be solved. In consequence this might require an additional preconditioning step which is provided by most of the commercial solvers.

Summary and conclusion of Chapter 5

This chapter describes the process of creating a set of finite elements from a NURBS-based B-Rep model which is provided by a CAD system. The elements can then be used to solve structural shell problems in a finite-element-like manner within an existing FE-solver. Possible data structures are provided as well. In addition, a new adaptive integration scheme is presented which allows to treat all *integration domains* (untrimmed and trimmed) in exactly the same way.

All decisions made for IBRA had the goal to develop a simple and robust analysis approach which can analyze complex CAD models without any modifications and can be implemented easily into an existing FE-solver.

Simplicity is the ultimate sophistication.

Leonardo da Vinci

CHAPTER 6

Analysis in Computer Aided Design

The goal of an efficient design-through-analysis workflow is to simulate the real behavior of a structure within a short time. The chosen analysis approach must therefore deliver solutions which

- meet a specified level of reliability and
- are obtained under minimal usage of resources (human effort, computational cost, etc.).

In times of almost unlimited and relatively cheap computer power, one of or probably the biggest challenge for an efficient design-through-analysis workflow is the minimization of human effort. Because the latter is time consuming and thus expensive from a business point of view.

This chapter introduces a new concept for minimizing the human effort for structural shell problems by using the proposed *isogeometric B-Rep analysis* (IBRA, see Chapter 4). The resulting CAD-integrated design-through-analysis workflow is named *Analysis in Computer Aided Design* (AiCAD) (see also Breitenberger et al. [17]). The implementation of an AiCAD workflow requires a CAD system, a FE-solver, and an interface software. After a short description of the boundary conditions given by well established product development processes this chapter explains how the components mentioned above need to be linked for realizing a CAD-integrated design-through-analysis workflow.

6.1 Product development process using CAx systems

A product development process driven by *computer-aided systems* (CAx) can be simplified as it is shown in Figure 6.1. The first step of such a process is to model a first shape of a product within a *computer-aided design* (CAD) system. As a next step the CAD model needs to be evaluated with *computer-aided engineering* (CAE) tools, which at their turn usually lead to CAD model updates. These steps have to be repeated until all requirements on the product are fulfilled. Usually this process leads to many geometry updates within the *design-through-analysis* (CAD-CAE) workflow. An efficient analysis modeling thus plays a crucial role for the efficiency of the overall process. Especially because an optimal use of resources requires the creation of multiple different analysis models.

The remainder of this section summarizes a few important aspects of design and analysis modeling with special attention to the two analysis approaches

- classical FEA (see also Section 3.4) and
- *isogeometric B-Rep analysis* (IBRA) (see also Chapter 4)

for solving structural shell problems.

6.1.1 Design modeling

An important aspect for design modeling is the fast creation and modification of geometrical shapes. A NURBS-based B-Rep model is well suited for this task. This is one reason why it is standard in industry especially for freeform geometries (see Section 2.2). To allow for easy shape modifications such models are fully parametrized. An example of such a parametrized CAD model is shown in Figure 6.2, where the angle of a line is changed by moving a point – the CAD parameter. This parameter modification entails

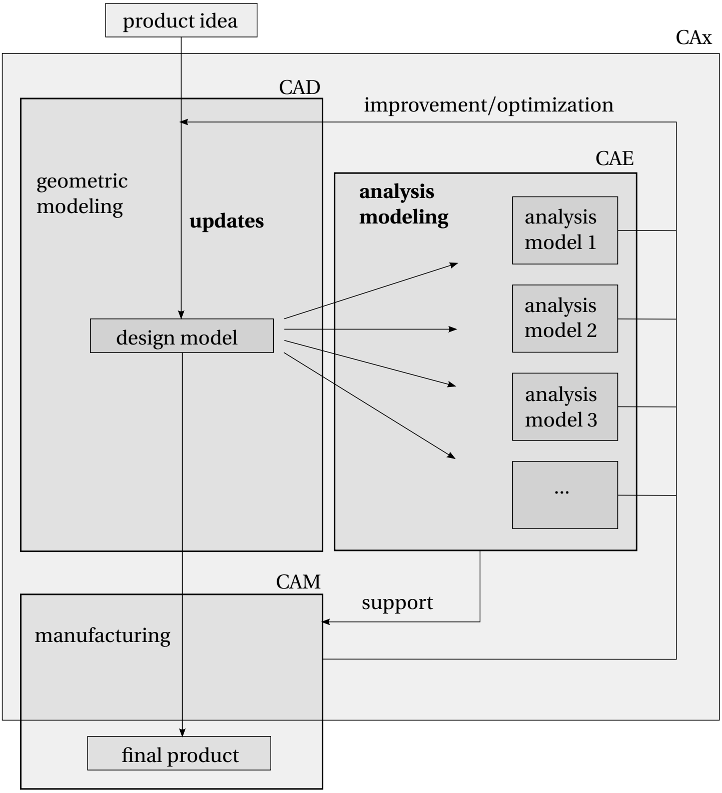
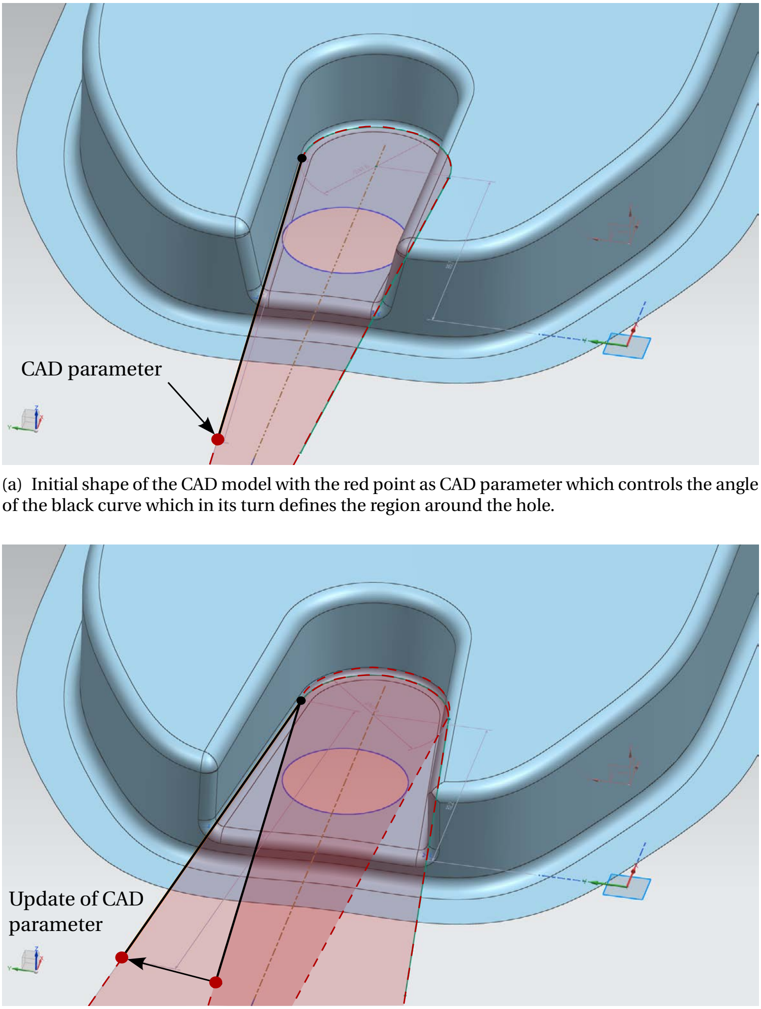


Figure 6.1: Simplified overview of a product development process using *computer-aided systems* (CAx) with special attention on design-through-analysis workflow (CAD-CAE) including CAD model updates



(a) Initial shape of the CAD model with the red point as CAD parameter which controls the angle of the black curve which in its turn defines the region around the hole.

(b) Updated CAD model: the angle of the line is changed by modifying the CAD parameter (position of the red point)

Figure 6.2: Example of a CAD parameter update on a fully parametrized CAD model within *Siemens NX*

the update of the entire CAD model. Within a design-through-analysis workflow, analysis models are therefore derived from NURBS-based B-Rep models.

6.1.2 Analysis modeling for classical FEA

In this thesis, analysis modeling refers to the preparation of a model for analysis itself. In industry, structural problems are solved predominantly with classical *finite element analysis* (FEA), which typically uses linear or quadratic polynomials defined over non-overlapping *parametric domains* (the elements) for describing the geometry of the analyzed model (see also Section 3.4). Thus, within the standard analysis workflow, illustrated in Figure 6.3, the NURBS-based B-Rep model needs to be converted into a finite element mesh. This change of geometry description is called meshing (see Section 3.4.2).

Another step to obtain the analysis model is the introduction of analysis-related data, e.g. material properties and boundary conditions. They can either be applied directly to the finite element mesh, e.g. by selecting the elements or nodes (option 1 in Figure 6.3), or to the CAD model (option 2). In option 2, properties are assigned as attributes to the B-Rep entities (e.g. *edges*) and after meshing they are transferred to the finite element mesh. This can be done automatically, option 2 is thus more convenient for the user, especially when re-meshing of the CAD model is necessary. The analysis itself is performed using a finite element (FE) solver. An example of a CAD model with analysis related data is illustrated in Figure 6.4.

The steps in the standard design-through-analysis workflow can be performed by using either different programs (CAD and FE program) or by using an integrated CAD/CAE system.

6.1.3 Analysis modeling for IBRA

In contrast to the standard analysis (CAD-CAE) workflow, in AiCAD the geometry representation of the design and the analysis models is the same. The design model just needs to be refined for the analysis (see Section 2.2.2), which does not change the representation of the model but only its description. The refinement is necessary to approximate the solution with adequate accuracy and can be assigned to the *B-Rep faces* as a property. In

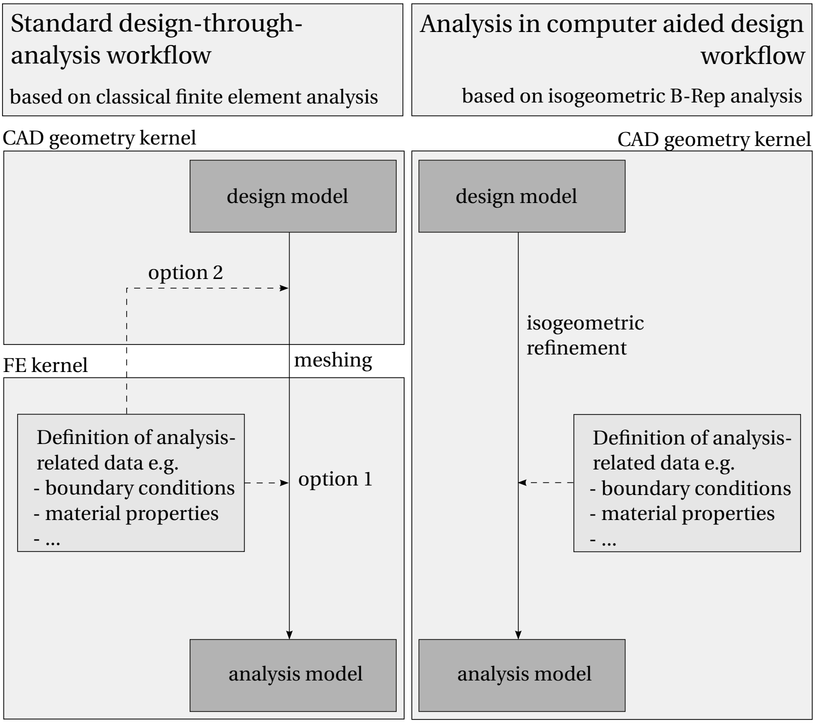

Standard design-through-analysis workflow
based on classical finite element analysis
Analysis in computer aided design workflow
based on isogeometric B-Rep analysis
CAD geometry kernel
design model
option 2
FE kernel
meshing
Definition of analysis-related data e.g.
- boundary conditions
- material properties
- ...
option 1
analysis model
CAD geometry kernel
design model
isogeometric refinement
Definition of analysis-related data e.g.
- boundary conditions
- material properties
- ...
analysis model

Figure 6.3: Difference between standard analysis (CAD-CAE) workflow based on classical *finite element analysis* (FEA) and proposed *analysis in computer aided design* (AiCAD) workflow based on *isogeometric B-Rep analysis (IBRA)*

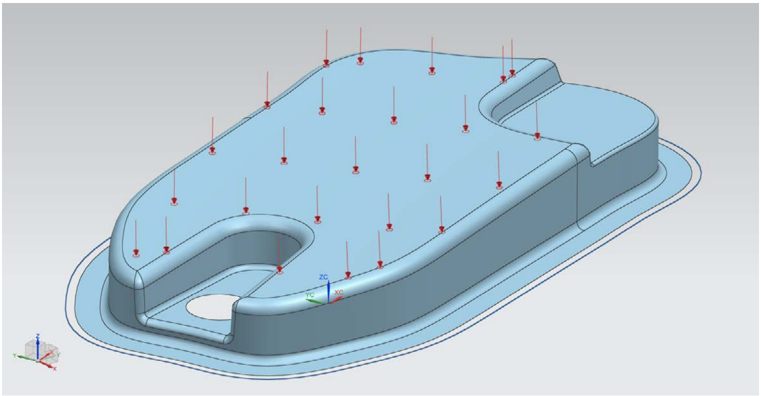


Figure 6.4: CAD model of an oil sump with analysis related data in the integrated CAD/CAE/CAM software *Siemens NX* (see Siemens PLM Software [84]). Analysis data like loads and supports can be visualized whereas material properties cannot. All data are saved as attributes at the CAD model.

AiCAD, not only the introduction of analysis-related data can be achieved in the CAD environment (option 2 in the standard analysis workflow) but also the postprocessing of the analysis. Here, of course, the CAD system needs to be extended to include the desired postprocessing functionalities, e.g. visualization of the stresses on the CAD model. This is usually done by using the API (*Application programming interface*) of the CAD software (see Figure 6.5).

The difference between standard analysis and the AiCAD workflow lies in the representation of the analysis model and its solution. In classical FEA, the (deformed and undeformed) geometries are represented by a finite element mesh, whereas in AiCAD, the geometries are represented by a NURBS-based B-Rep model.

6.2 Realization of an AiCAD worflow

In this section, the most important aspects for realizing an AiCAD workflow within an existing CAD system and using an existing FE solver are outlined. The AiCAD workflow has the following four main components:

- CAD system
- AiCAD interface software
- FE Solver
- eventually external postprocessing tool (e.g. HTML based)

The workflow can be seen in an abstract form independent of the implementation of AiCAD, as shown in Figure 6.5. In the following text, the various components and their tasks are briefly explained.

6.2.1 CAD system

The design component of an AiCAD workflow is realized in a CAD program, where the geometry kernel is used for the following tasks:

- geometric modeling (see section 6.1.1)
- pre- and eventually postprocessing for IBRA

Geometric modeling is the intrinsic function of the CAD program, whereas pre-and postprocessing may be seen as additional functionalities. The pre- and postprocessing in a CAD program require an API, which endows the CAD software with missing functions, such as

- assigning analysis related data to B-Rep entities (see section 6.1.3)
- assigning refinement parameters (see section 6.1.3)
- visualizing deformed geometries (see figure 6.6)
- ...

Depending on the CAD program, some of these functions are already available. To give an example, the integrated CAD/CAE/CAM software *Siemens NX* (see Siemens PLM Software [84]) can assign analysis related data to B-Rep entities (see figure 6.4), whereas in the CAD program *Rhinoceros* (see Rhinoceros [70]), this needs to be realized by the API of *Rhinoceros*.

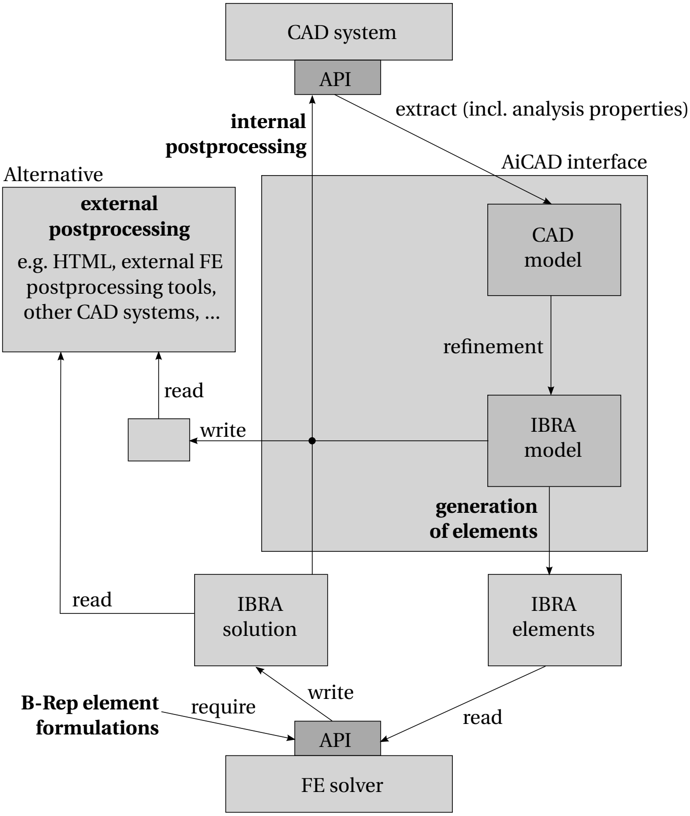
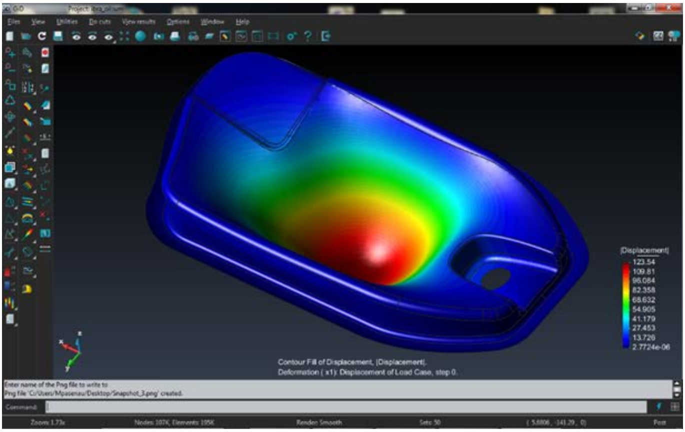


Figure 6.5: Overview of software components of an AiCAD workflow (see also Figure 5.1)

Note that in this thesis, it is assumed that a CAD program provides access to geometric data and functions such as the visualization of NURBS surfaces and space curve mapping (see section 2.2.3). From experiences, this assumption holds for almost all major CAD programs. At the author's chair, the AiCAD workflow has been realized using three different programs: *Rhinoceros* (Rhinoceros [70]), *Siemens NX* (Siemens PLM Software [84]), and *GiD* (GiD [34]).



Contour Fill of Displacement, |Displacement|.
Deformation (x1): Displacement of Load Case, step 0.

Enter name of the Png file to write to
Png file 'C:/Users/Mpasenau/Desktop/Snapshot_3.png' created.

Figure 6.6: IBRA postprocessing of an oil sump in the pre- and postprocessing software GiD (see GiD [34])

6.2.2 AiCAD interface software

The AiCAD interface software is used for communicating between the CAD program and the FE-solver. The interface software needs to provide, among others, the following functions:

- reading data from the CAD program
- generating *isogeometric elements* and *B-Rep elements* for the FE-solver (see Section 5.3)
- supporting pre- and postprocessing for IBRA in the CAD program

For this thesis, the interface is realized with the software TeDA (*Tool to Enhance Design by Analysis*), developed at the author's chair. TeDA is a C++ library, which can be used for the API in *Rhinoceros* and the integrated CAD/CAE/CAM software *Siemens NX*.

6.2.3 FE-solver

The FE-Solver is used for solving the IBRA problem described by *isogeometric* and *B-Rep elements*. Of course the corresponding element formulations

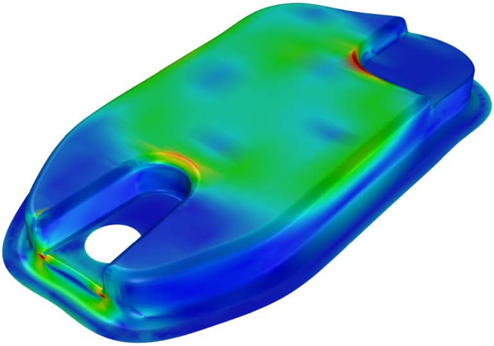


Figure 6.7: Visualization of vMises stresses on the CAD model of an oil sump (see also Figure 6.4) using the CAD software *Rhinoceros* (see Rhinoceros [70])

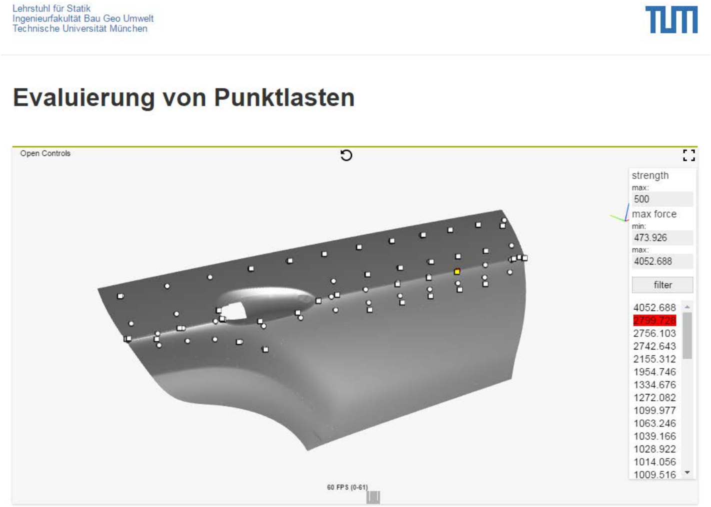
(see Section 4.4, Section 4.5, and Section 4.6) need to be implemented in the FE-solver like any other classical element formulation. Usually widely used FE-solver provide an API for implementing new elements. All the examples in this thesis have been computed with the FE-solver Carat++ developed at the authors' chair.

6.2.4 Postprocessing

The output of IBRA basically consists of the displacement field of *active* nodes, meaning the ones that influence the *trimmed domain*. The displacement field can be used for determining the deformed geometry and stresses. For visualizing deformed geometries, the following data are sufficient:

- description of trimmed and refined NURBS surfaces (see also Section 2.2.3)
- displacements values of *active* control points resp. nodes

Note that the *inactive* nodes do not have displacement because they have no influence on the *trimmed domain*. An example of a deformed CAD



Lehrstuhl für Statik
Ingenieurfakultät Bau Geo Umwelt
Technische Universität München

Evaluierung von Punktlasten

strength
max:
500
max force
min:
473.926
max:
4052.688

4052.688
2799.728
2756.103
2742.643
2155.312
1954.746
1334.676
1272.082
1099.977
1063.246
1039.166
1028.922
1014.056
1009.516

Figure 6.8: Visualization of a CAD model for special postprocessing functionalities within a website.

model in the pre- and postprocessing software GiD (see GiD [34]) is illustrated in Figure 6.6. In addition Figure 6.7 shows an example of a stress plot on a CAD model created with the CAD program *Rhinoceros* (see Rhinoceros [70]). For customized postprocessing functionalities also a HTML(web)-based visualization can be very useful. An example of such a website is shown in Figure 6.8

Summary and conclusion of Chapter 6

This chapter describes the steps required for realizing an CAD-integrated design-through-analysis workflow (AiCAD) within an existing CAD system. AiCAD has been realized within three different programs: *Rhinoceros* (Rhinoceros [70]), *Siemens NX* (Siemens PLM Software [84]), and *GiD* (GiD [34]). This demonstrates the flexibility of the developed concept.

In addition, the difference between AiCAD and a standard design-through-analysis workflow is explained.

Everything should be made as simple as possible, but not simpler.

Albert Einstein

Chapter 7

Examples and investigations

To highlight the accuracy, flexibility and robustness of IBRA, some well selected examples with increasing complexity are discussed. To demonstrate the correctness and to quantify the approximation errors of IBRA, the results are compared to solutions that are either computed analytically, with *isogeometric analysis* (IGA), or with commercial finite element software. The examples range from geometrically linear plates in membrane action to geometrically nonlinear shells, including weak boundary conditions (see also Section 4.5). For all examples, a linear elastic, isotropic material is used.

This chapter can be separated into three parts. The first part deals with problems that are discretized with one *trimmed NURBS surface* for investigating the *Adaptive Gaussian Integration Procedure* (AGIP) (see Section 5.5) and the influence of trimming tolerances on analysis results. The second part deals with *B-Rep edge elements* (according to Section 4.5) which are used to enforce different types of weak boundary conditions. The third part deals with the application of IBRA to industrial problems.

PART 1 starts with an example of a simple plate in membrane action to show the accuracy of the integration scheme AGIP. The second example deals with a geometrically linear (i.e. applying linear kinematics) plate in bending, which is used to show the robustness of AGIP against geometric distortions. The last example within the first block deals with a geometrically nonlinear shell problem to demonstrate the application of IBRA also to mechanically challenging problems and the flexibility of IBRA in terms of trimming tolerances.

The first example of PART 2 deals with a geometrically nonlinear cantilever subjected to an end moment, a so called mainspring. This example is used for discussing weak coupling boundary conditions for different discretizations, the influence of "dirty" geometries (gaps and overlaps) on the result, and the choice of appropriate penalty factors (according to Section 4.5.1). The second example deals with a geometrically nonlinear shell problem discretized with two trimmed surfaces focusing on weak coupling conditions.

PART 3 shows the application of IBRA to real industrial CAD models of an oil sump and an engine bonnet.

Moreover, all examples highlight the simplicity and straightforwardness of analysis model preparation using IBRA even for very complex geometries. All geometries for the examples are created either with the CAD program *Rhinoceros 5* (see Rhinoceros [70]) or the integrated CAD/CAE/CAM software *Siemens NX* (see Siemens PLM Software [84]). The trimming tolerance (according to Section 2.2.3) used for the examples is denoted separately for each example. For some few examples, it would be possible to represent the geometries exactly, i.e. without any trimming error, but this is not the goal of the proposed method. The goal of IBRA is to use the CAD model "as it is". For all presented examples, the geometries therefore are created in a CAD system, and they are analyzed directly. The analysis itself is performed with the in-house finite element program CARAT++. The results of the classical finite element analysis are computed in *ANSYS* (see ANSYS [2]) or *NASTRAN NX* (see Siemens PLM Software [84]) by using quadrilateral-dominant meshes with bi-linear quadrilaterals and triangular *Reissner-Mindlin* (RM) shell elements. For IBRA, the *NURBS-based Kirchhoff-Love (KL) shell element* (see Kiendl et al. [44]) is used. Note that KL theory is not always mentioned explicitly. The postprocessing for IBRA is realized with the interface software TeDA (see Section 6.2.2) used within the API of the CAD program *Rhinoceros 5*.

7.1 Definition of error measures

In order to verify the proposed technique, a few quantitative error measures need to be defined for the examples. The error over a *parametric domain* Ω can be expressed by the sum of the L^2-norms of each component of the stress tensor as follows:

$$e_r = \sum_i \| \sigma_{i\ \text{exact}} - \sigma_{i\ \text{num}} \|_{L^2(\Omega)}, \tag{7.1}$$

where i represents the components $\{xx, yy, xy\}$. In cases where a computed result (e.g. displacement) at a specific point is compared to a reference solution, the following relative error measure is used:

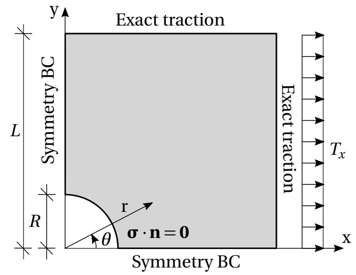
$$e_r = \frac{|u_{\text{ref}} - u_{\text{IBRA}}|}{|u_{\text{ref}}|} \tag{7.2}$$

For a general comparison of the relative difference between two results (res 1 and res 2) the following formula is used:

$$d_r = \frac{|u_{\text{res1}} - u_{\text{res2}}|}{|u_{\text{res1}}|} \tag{7.3}$$

7.2 Infinite plate with circular hole

The first example deals with the two-dimensional linear elastic problem shown in Figure 7.1. This setup is used to solve an infinite plate with a circular hole under in-plane tension at infinity. The exact solution to this problem is given in Gould [36], and is solved with IGA in Hughes et al. [40]. Moreover, the problem had already been solved by Kim et al. [47] using a trimmed NURBS surface. Since just another numerical integration scheme is applied (see AGIP in Section 5.5), one should obtain similar results as those published in Kim et al. [47].



Parameters:

Geometry: $L=4.0 \quad R=1.0$
Material: $E=10^5 \quad \nu=0.3$
Far-field traction: $T_x=10.0$

Analytical solution:

$$\sigma_{rr}(r,\theta)=\tfrac{T_x}{2}(1-\tfrac{R^2}{r^2})+\tfrac{T_x}{2}(1+3\tfrac{R^4}{r^4}-4\tfrac{R^2}{r^2})\cos(2\theta)$$

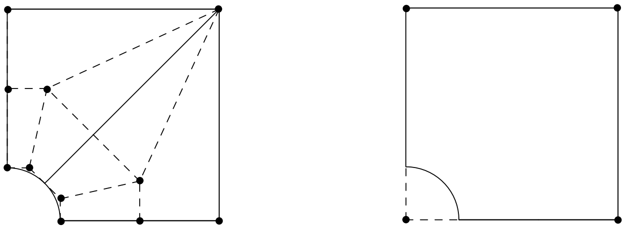
$$\sigma_{\theta\theta}(r,\theta)=\tfrac{T_x}{2}(1+\tfrac{R^2}{r^2})-\tfrac{T_x}{2}(1+3\tfrac{R^4}{r^4})\cos(2\theta)$$

$$\sigma_{r\theta}(r,\theta)=-\tfrac{T_x}{2}(1-3\tfrac{R^4}{r^4}+2\tfrac{R^2}{r^2})\sin(2\theta)$$

Figure 7.1: Example of infinite plate with hole - problem description and exact solution

To compare the results of IGA and IBRA, as well as the analytical solution, the stress distributions $\sigma_{\theta\theta}$ along the hole are plotted (see Figure 7.4), and the error as the sum of the L^2-norm of each stress tensor component (see Equation (7.1)) over the *parametric domain* is investigated (see Figure 7.6).

The different discretizations for IGA and IBRA are shown in Figure 7.2. The untrimmed surface used for IGA employs the same definition as proposed in Hughes et al. [40], and the trimmed surface used for IBRA is created using the CAD program *Rhinoceros* with a very low trimming tolerance of 10^{-8} units (see Section 2.2.3). For this specific example, one could define the trimmed NURBS surface without any trimming error by using an exact NURBS description of the trimming curve, but this is not common when using a CAD system, and doing so would contradict the proposed philosophy of integrating design and analysis (see Chapter 6). Note that due to the reduced modeling effort, the standard approach in practice would be to create a trimmed surface as it is shown in Figure 7.2(b) even if it does not represent the geometry in Figure 7.1 exactly. IBRA allows the usage of such trimmed surfaces directly for analysis without any modifications. In contrast to that IGA in its original form works on the exact geometry, however, requires complicated and costly geometric models with untrimmed surfaces suitable for analysis (see Figure 7.2(a)). Independent of the chosen geometry discretization one needs to refine the surface for analysis according to the desired solution accuracy. This is a requirement of mechanics, but not of the workflow. Different levels of refinement for the initial discretizations (see Figure 7.2) are shown in Figure 7.3. In both cases, existing



(a) Untrimmed NURBS surface description used for IGA

(b) Trimmed NURBS surface description used for IBRA

Figure 7.2: Different geometric descriptions (trimmed and untrimmed NURBS surfaces) of the geometry in the example, shown in Figure 7.1. The dashed lines show the control point polygons, and the black points represent the control points.

non-zero knot spans in the parameter space are subdivided sequentially in both directions. Note that in the step from the geometry in Figure 7.3(a) to the one in Figure 7.3(b), only one direction is divided in order to ensure that the number of elements in both directions is the same. Figure 7.3 graphically summarizes the refinement strategy for untrimmed and trimmed surfaces. These discretizations are the basis for a comparative evaluation of the analysis results of both approaches.

Figure 7.4 shows a comparison of the stress distribution $\sigma_{\theta\theta}$ at $r = R = 1$ along the boundary of the hole for different geometry refinements (see Figure 7.3). Each plot shows the analytical, IGA, and IBRA solution using a specific number of elements. The polynomial degree is p=q=3 for each plot. Owing to the global definition of the basis functions (see Figure 7.5(b)) along the hole boundary for the trimmed version with 4x4 elements, the IBRA solution oscillates slightly around the analytical solution (see Figure 7.4(a)). In contrast, the basis functions of the untrimmed discretization have local support (see Figure 7.5(a)), which allows for better approximation of the tangential stresses along the boundary of the hole. The same effect can be observed for geometries with 8x8 elements (see Figure 7.3(d) and Figure 7.3(h)). Again, for this level of refinement, the trimmed discretization has only three elements for approximating the stresses at the hole boundary, as opposed to eight elements for the untrimmed case. In the trimmed case result, one can again observe small oscillations around

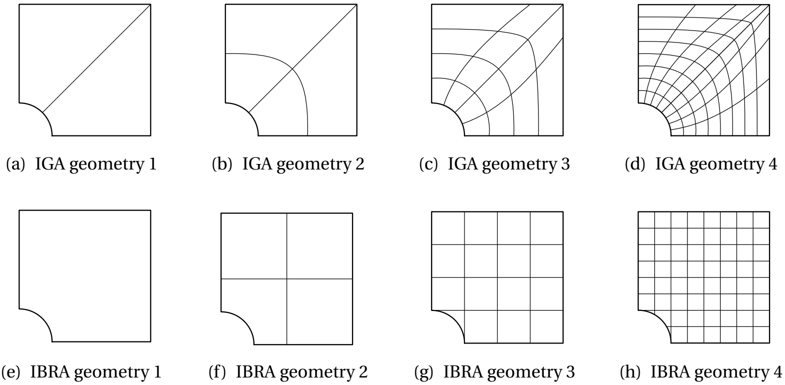
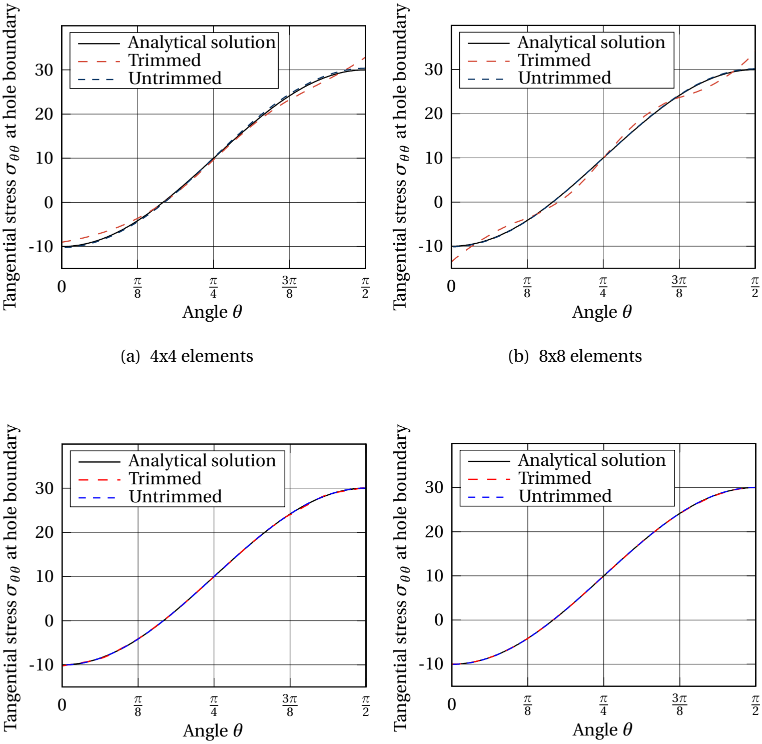

(a) IGA geometry 1
(b) IGA geometry 2
(c) IGA geometry 3
(d) IGA geometry 4
(e) IBRA geometry 1
(f) IBRA geometry 2
(g) IBRA geometry 3
(h) IBRA geometry 4

Figure 7.3: Different (refined) geometry descriptions of the example shown in Figure 7.1

the analytical solution. With a greater number of elements, the approximations become better and converge towards the exact solution. Similar plots obtained by using the B-Spline version of *finite cell method* (FCM) have been published by Schillinger et al. [76]. For this example the only difference between the B-Spline version of FCM and IBRA lies in different numerical integration procedures of the trimmed resp. visible domain.

The conclusion out of these observations is that the untrimmed discretization uses a better-suited parametrization for this specific problem. Thus, it is not surprising that the trimmed discretization does not perform as well as the untrimmed one. Nevertheless the convergence behavior with trimmed surfaces is remarkable which is also affirmed by the following investigations.

To confirm overall convergence to the exact solution, the sum of the L^2-error norm of each stress tensor component over the *parametric domain* (see Equation (7.1)) is plotted for different refinements in Figure 7.6. This figure shows the error norm versus the largest diameter found in the geometry and the number of *degrees of freedom* (DOF). It can be seen that the plots of the untrimmed and trimmed discretization cases are similar. As expected, the untrimmed discretization shows a slightly faster convergence. This is because the largest deviations for the stresses are in the



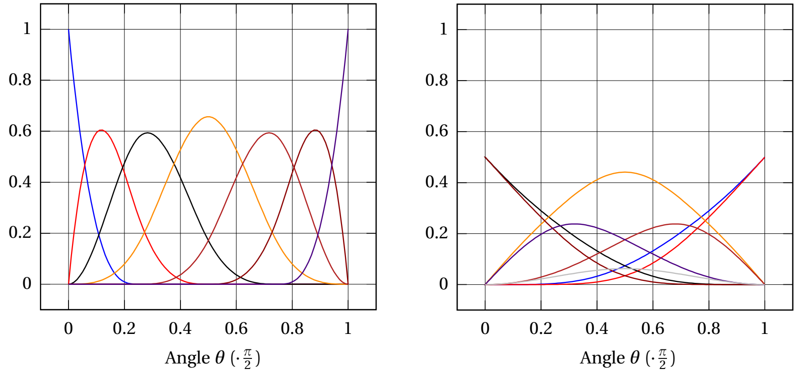
(a) 4x4 elements

(b) 8x8 elements

(c) 16x16 elements

(d) 32x32 elements

Figure 7.4: Stress distribution of $\sigma_{\theta\theta}$ at hole boundary $r = 1$ for different refinements. Comparison of results of untrimmed and trimmed discretizations.



(a) Basis functions of hole boundary for untrimmed discretization shown in Figure 7.3(c)

(b) Basis functions of hole boundary for trimmed discretization shown in Figure 7.3(g)

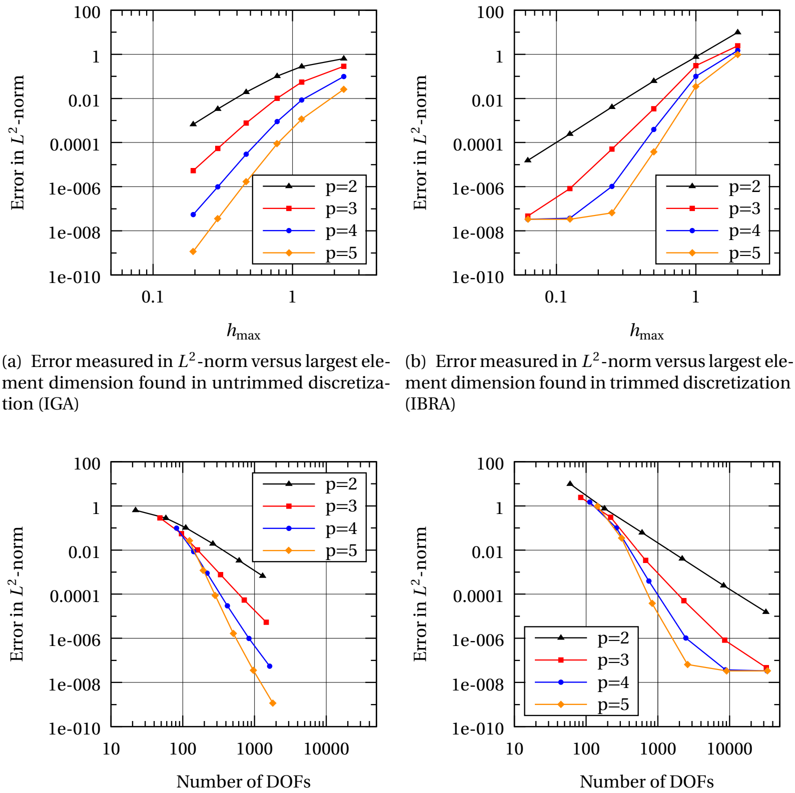
Figure 7.5: Comparison of the basis functions at the hole boundary for different discretizations

region around the hole, where the untrimmed version is optimally refined, as explained before.

The plot in Figure 7.6(a) shows the same behavior as the one presented in Hughes et al. [40], even though the geometry refinement is different. In the present work, the parameter space is refined uniformly, whereas in Hughes et al. [40], the geometry is refined uniformly. The results of the plots in Figure 7.6(b) and Figure 7.6(d) are similar to those published in Kim et al. [47] but with much more levels of refinement to underline the high accuracy of the proposed integration scheme. Note that the leveling off in the range of 10^{-8} is caused by the trimming tolerance of 10^{-8} units, which has been used for generating the geometry.

Summary and conclusions

From this example, it can be concluded that the untrimmed discretizations match the resolution requirement of the problem's mechanics slightly better than the trimmed discretizations. The price to be paid for this ideal discretization is greater geometrical modeling effort. In addition, it can be stated that restricting the *parameter domain* of a surface to a *trimmed*



(a) Error measured in L^2-norm versus largest element dimension found in untrimmed discretization (IGA)

(b) Error measured in L^2-norm versus largest element dimension found in trimmed discretization (IBRA)

(c) Error measured in L^2-norm versus number of degrees of freedom for IGA (untrimmed)

(d) Error measured in L^2-norm versus number of degrees of freedom for IBRA (trimmed)

Figure 7.6: Convergence of sum of L^2-norm of each stress tensor component in Cartesian coordinates x,y and xy (see Equation (7.1))

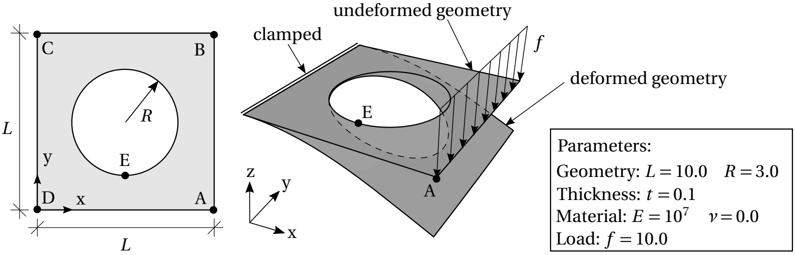


Figure 7.7: Example of a plate in bending - problem description

domain and using an accurate integration scheme neither influences the convergence properties nor the accuracy of the underlying functional basis for analysis purposes. Having the same slopes for the untrimmed and trimmed discretizations means that the proposed AGIP integration scheme (see Section 5.5) fulfills the moment fitting equation. IBRA only takes over the geometrical modeling error from CAD caused by the predefined trimming tolerance within the CAD system (see also Section 7.4) but this is done on purpose.

7.3 Plate in bending with circular hole

This example deals with a linear (i.e. linear kinematics) plate in bending. The setup is described in Figure 7.7. The goal of this example is mainly to investigate the robustness of IBRA against distortions in the geometric description. Note that the trimmed geometries of the previous example have no such distortions because the *parameter domain* and the geometry coincide i.e. the mapping between these two domains is one resp. constant. Independent of the initial parametrization of the geometry, the results of the refined geometries should converge to the same solution. Thus, the distortion of the geometric description must not influence the (converged) results, but only lead to a difference in the convergence rate.

The example described in Figure 7.7 is solved for different geometric descriptions, where one set of descriptions has a geometric distortion. This geometric distortion is described in Figure 7.8, where the red curve (CP 2 - CP 8) represents a parameter curve defined in the middle of the *pa-*

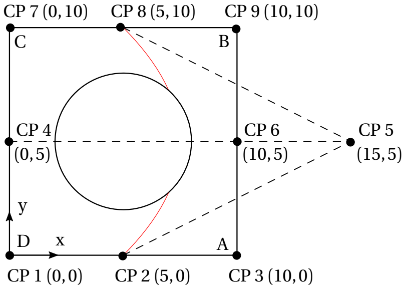


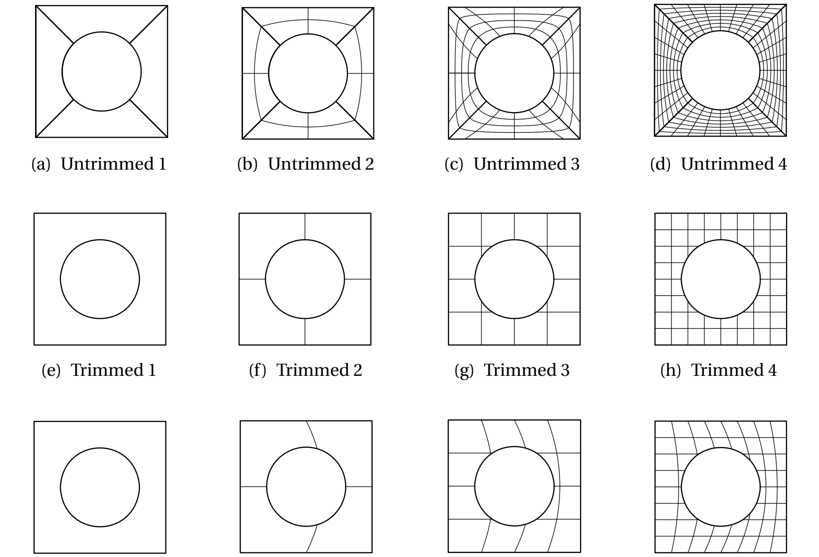
Figure 7.8: Distortion of geometric description of example shown in Figure 7.7

rameter domain. It is used to illustrate the distortion of the geometric description. By intention, the distortion is quite substantial, which allows for a demonstration of the robustness of the proposed integration scheme against numerical errors in the J_1 mapping.

The different geometric descriptions for this example are shown in Figure 7.9, where the trimming was performed with a tolerance of 10^{-8} units. The geometries in Figure 7.9 (a-d) are described by four untrimmed NURBS surfaces, those in (e-h) are described by simple trimmed NURBS surfaces, and those in (i-l) are described by trimmed NURBS surfaces with a distorted geometric description (see also Figure 7.8).

The problem is solved for the geometric descriptions shown in Figure 7.9 and for some further refinements following the same logic. For the NURBS-based KL shell element, a polynomial degree of $p = q = 3$ is used. For the untrimmed geometry bending strips have to be used (see Kiendl et al. [43]) even if the collinearity of the control points at the four corner points can not be provided (kink). In addition, the problem is solved with classical finite elements by using quadrilateral-dominant meshes with bi-linear RM elements.

Figure 7.10 shows the convergence of displacement in z-direction at point A (see Figure 7.7) for the four different types of discretizations. The displacement at point A converges for all types of discretizations using the KL theory to $u_z = -6.3499$ and for the RM theory to $u_z = -6.3748$. This results in a difference of 0.0249, which corresponds to a 0.3% discrepancy



(a) Untrimmed 1 (b) Untrimmed 2 (c) Untrimmed 3 (d) Untrimmed 4

(e) Trimmed 1 (f) Trimmed 2 (g) Trimmed 3 (h) Trimmed 4

(i) Trimmed geometry with a distorted geometric description 1 (j) Trimmed geometry with a distorted geometric description 2 (k) Trimmed geometry with a distorted geometric description 3 (l) Trimmed geometry with a distorted geometric description 4

Figure 7.9: Different geometry descriptions of geometry shown in Figure 7.7

which can be explained by the different mechanical models. The Dirichlet boundary condition for this example is applied strongly for all discretizations.

As expected the IBRA solution with geometric distortion shows slightly different convergence behavior than the simple trimmed discretizations but converges to exactly the same result. Moreover, it can be seen that the untrimmed descriptions with the bending strips initially behaves too soft, but also converge to the same solution. Just the influence of the bending strip method (penalty factor) and the trimming tolerance causes very little deviations. For the bending strip factor, a value of 5 is used, which corresponds to a penalty factor of 10^5 times E (Young's modulus). According to Kiendl et al. [43], this is within the optimal range.

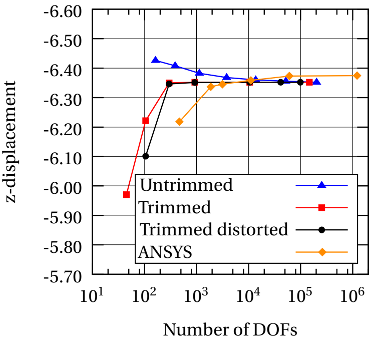


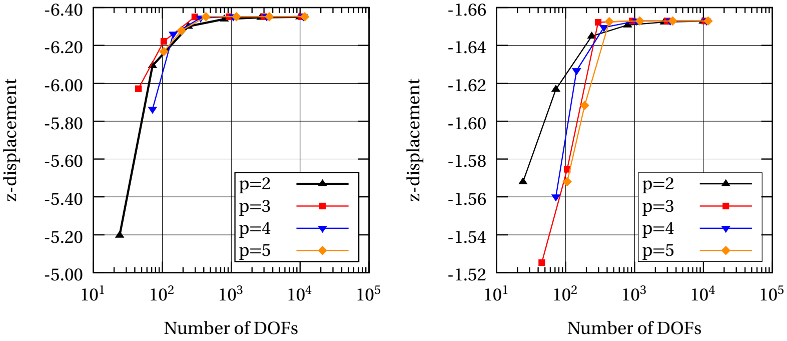
Figure 7.10: Displacement of point A in z-direction (see Figure 7.7) using different discretizations. For the NURBS-based discretizations a polynomial degree of p=q=3 is used.

For highlighting the good convergence behavior of IBRA, the convergence of the displacements at points A and E (see Figure 7.7) of the simple trimmed IBRA geometries are further investigated. The convergence plots of the displacement in z-direction for these two points are plotted in Figure 7.11 by using different polynomial degrees. The convergence behavior of point E is shown for illustrating the very good point-wise convergence achieved at points on internal hole boundaries as well.

A typical deformed geometry within IBRA is shown in Figure 7.12, where the result of the plate in bending example using a simple trimmed geometry is given. The points resp. nodes in the middle of the hole have no influence on the solution and thus they are called *inactive*.

Summary and conclusions

The conclusion of this example is that in contrast to the previous example, the trimmed geometries perform better than the untrimmed ones and yet can be modeled with greater ease. It is shown that IBRA is very robust against distortions in the geometric description owing to the use of the proposed integration scheme AGIP (see Section 5.5). In addition, the example confirms very good point-wise convergence achieve at points on



(a) Displacement in z-direction at point A

(b) Displacement in z-direction at point E

Figure 7.11: Convergence plots of displacement in z-direction for points A and E shown in Figure 7.7 for trimmed geometries.

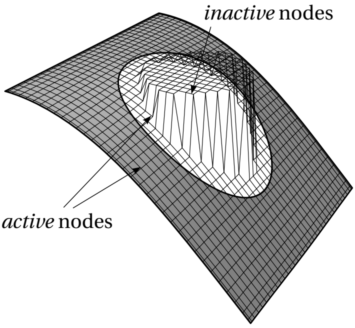


Figure 7.12: Deformed geometry of the example shown in Figure 7.7, including control point net and examples of *active* and *inactive* nodes resp. control points

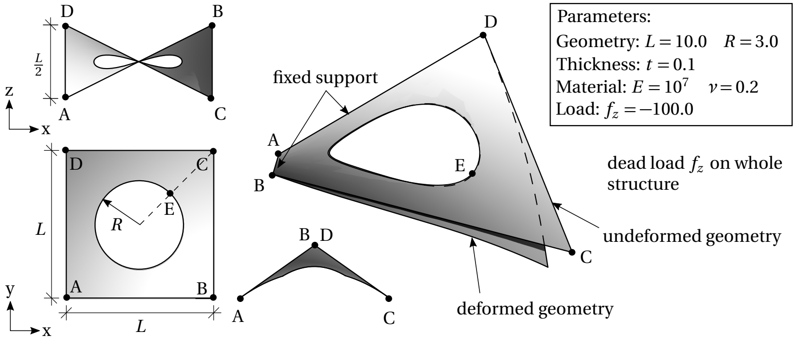


Figure 7.13: Example of a doubly curved trimmed shell with a circular hole - problem description

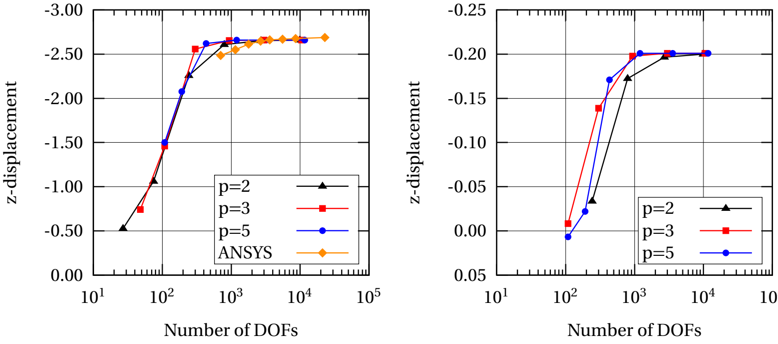
boundaries. Moreover, the difference between *active* and *inactive* nodes is shown.

7.4 Doubly curved shell with circular hole

The last example of the first part deals with a geometrically nonlinear doubly curved shell problem. This example illustrates that IBRA works for geometries with challenging mechanics as well. In addition, the influence of trimming tolerances (see Section 2.2.3) on the analysis result is discussed. Additionally the load is applied to the trimmed domain.

The example is described in Figure 7.13. Note that easy modeling of such a geometry with untrimmed NURBS surfaces is almost impossible. Thus only trimmed discretizations are used. The trimming is performed with a high tolerance of 10^{-2} units, which is higher than the default CAD tolerance for such an example, as well as with a very low tolerance of 10^{-7} units. The results for the different trimming tolerances are compared for judging from the analysis perspective the influence of trimming tolerances. The IBRA results are verified against solutions of classical finite element analysis computed by the commercial FE program *ANSYS*.

For a better understanding the example and illustration of the geometrical non-linearity of the problem, the load-displacement curve for the displacement in z-direction of point C is shown in Figure 7.16. It can be



(a) Displacement in z-direction of point C

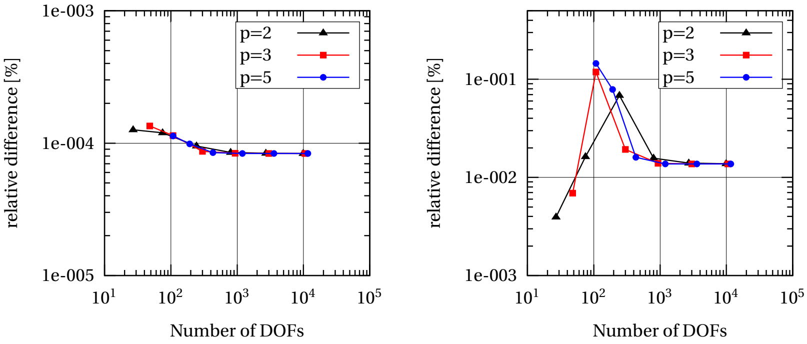
(b) Displacement in z-direction of point E

Figure 7.14: Displacement in z-direction of points C and E (see Figure 7.13) using trimming tolerance of 10^{-2} units

seen that the displacement is quite substantial and the structure softens after reaching a load factor of approximately 0.55.

To show the point-wise convergence also for this example, the two points C and E (see Figure 7.13) are investigated. Figure 7.14 shows the displacement in z-direction at points C and E for different polynomial degrees. For these plots, a trimming tolerance of 10^{-2} units is used. A comparison of the results obtained using different trimming tolerances will be shown later. Figure 7.14(a) additionally shows the convergence of classical FEA using quadrilateral-dominant meshes with bi-linear and triangular RM elements. It can be seen that classical FEA needs approximately 2000 DOFs for yielding an accurate (converged) solution, whereas with IBRA employing p=q=3 almost converged results can already be achieved with only 400 DOFs. Nevertheless all solutions converge to almost the same result (see Figure 7.14). For the displacement along the z-direction at point C, the IBRA results converge to −2.66029, whereas the FEA results converge to a slightly higher value of −2.6973. Again, the discrepancy can be explained by the use of different shell theories.

To show the influence of trimming tolerance on the IBRA results, the example is also solved for various geometries with two different trimming tolerances, namely 10^{-2} and 10^{-7}. Figure 7.15 shows the relative difference



(a) Relative difference for point C

(b) Relative difference for point E

Figure 7.15: Relative difference (see Equation (7.3)) in displacement of points C and E along z-direction in Figure 7.13 using trimming tolerance of 10^{-2} and 10^{-7} units

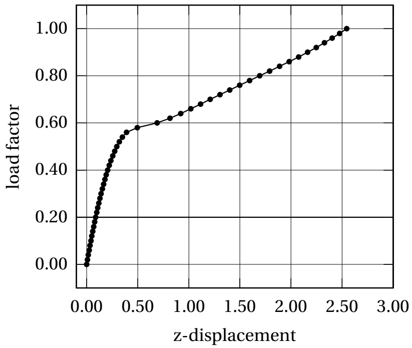


Figure 7.16: Load-displacement curve for point C, shown in Figure 7.13

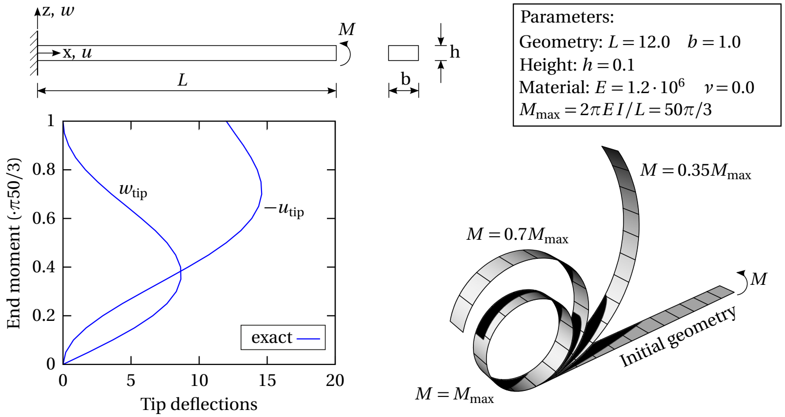


Figure 7.17: Example of mainspring - problem description

(see Equation (7.3)) between the results at points C and E by using two different trimming tolerances. For point C the relative difference is less than 10^{-4} and for point E it is less than $2 \cdot 10^{-2}$. This clearly indicates that the trimming tolerance, i.e., using a low resp. high trimming tolerance introduces the geometrical modeling error which at its turn influences the analysis result (see also Figure 7.6). For practical applications this influence can be neglected which is also done in industry using classical FEA which even worse introduces an additional meshing error.

Summary and conclusions

The conclusion of this example is that the analysis results slightly deviate for different trimming tolerances but from a practical point of view this effect can be neglected. Note that the trimming error is a geometrical modeling error which will be introduced to any analysis method which takes over the geometry from CAD. Within classical FEA the error coming from the mesh approximation, however, is an additional source of inaccuracies. This latter source of error is not introduced in an IBRA geometry. Moreover, this example demonstrates the good point-wise convergence behavior of IBRA also for mechanically challenging shell problems.

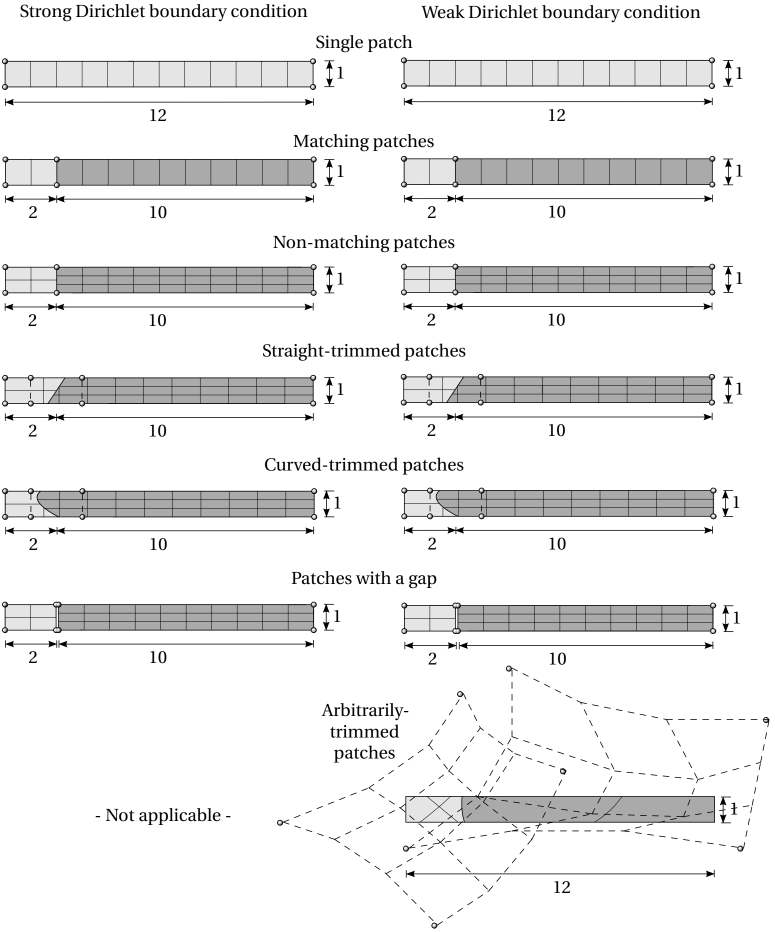


Figure 7.18: Different geometry descriptions of problem in Figure 7.17. The points represent characteristic control points. For all multi-patch geometries, a weak coupling boundary condition is used. Both types of Dirichlet boundary conditions, strong and weak, can be used.

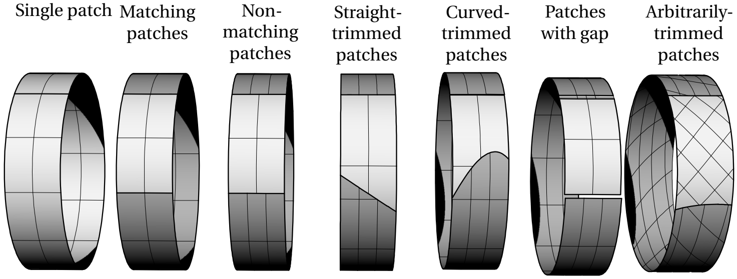


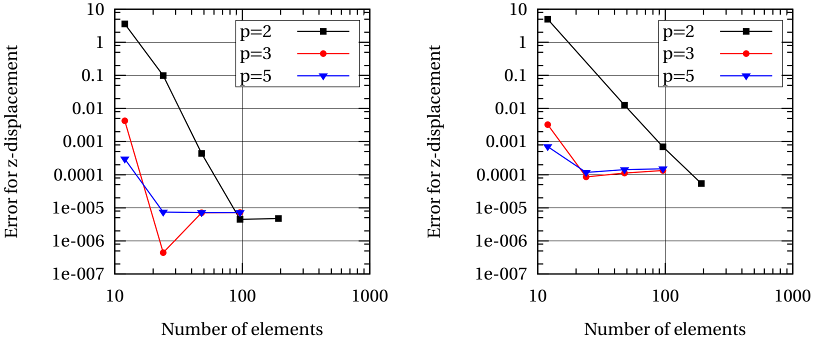
Figure 7.19: IBRA solutions of problem in Figure 7.17 with $M = M_{\text{max}}$ using different geometry descriptions from Figure 7.18

7.5 Cantilever subject to end moment

In the first example of the second part several aspects of weak Dirichlet and coupling boundary conditions are discussed. In addition, the effect of gaps and overlaps (see also *non-watertightness* in Section 2.4.1) for multi-patches is investigated.

Coupling of *parametric domains* and Dirichlet boundary conditions are enforced by *B-Rep edge elements* (see Section 4.3.3) following the element formulation from Section 4.5.1 resp. Section 4.5.2. The proposed formulations are purely geometry based and use a penalty approach. Thus, it requires the estimation of penalty factors. For this thesis, only one globally defined factor is used, which is applied for both constraining the displacements and rotations at common boundaries. Similar to bending-strips (see Kiendl et al. [43]), the proposed formulation achieves good results for a wide range of penalty factors. However, in contrast to the bending-strip method, the proposed formulation in Section 4.5.1 can handle non-matching patches along trimming curves including gaps and overlaps. For the following examples a relative penalty factor of $\alpha^{\text{rel}} = 100$ (see also Section 4.5.1) is used, if not indicated differently. This choice of the relative penalty factor will be explained later within this section.

The example itself treats the mainspring problem. The problem description is given in Figure 7.17 (see also Sze et al. [87]). This example is used for demonstrating the flexibility of IBRA from the perspective of the prob-



(a) Error of displacement in z-direction (middle of patch) for the geometry "non-matching patches" in Figure 7.18

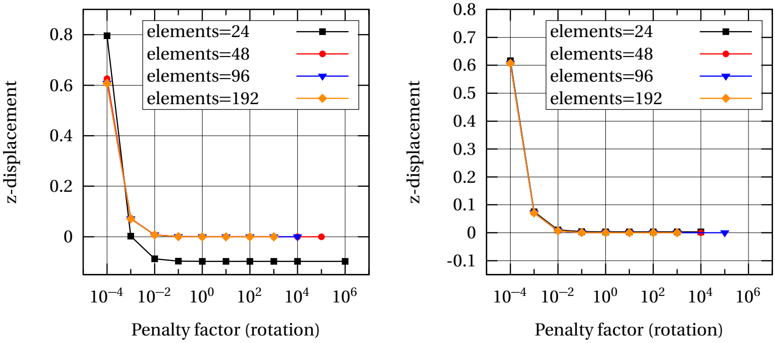
(b) Error of displacement in z-direction (middle of patch) for the geometry "straight-trimmed patches" in Figure 7.18

Figure 7.20: Error of displacement in z-direction (middle of tip) for the problem in Figure 7.17 by using different geometry descriptions with $M = M_{\text{max}}$

lem's geometry description. Thus, the problem is solved with IBRA for the different geometry descriptions shown in Figure 7.18. Here, for all geometries, a polynomial degree of $p{=}q{=}4$ is used. The multi-patch geometries use weakly enforced coupling boundary conditions (see Section 4.5.1). The Dirichlet boundary conditions can be enforced strongly or weakly, where strong boundary conditions are restricted to the patch boundaries (untrimmed surface). The geometry "arbitrarily-trimmed patches" demonstrates the necessity of weakly enforced boundary conditions along trimming curves. The acting moment is also applied with *B-Rep elements* referring to the rotation at the boundary.

The IBRA solutions corresponding to the different geometries are shown in Figure 7.19. All of them represent the expected solution of a closed ring for $M = M_{\text{max}}$.

For a detailed investigation of the solution quality, two convergence plots are discussed in Figure 7.20. The two plots show the displacement of the tip (point in the middle) in z-direction for the "non-matching" and "straight-trimmed" geometries (see Figure 7.18). It can be seen that both geometries have an absolute error of less than 10^{-4} in displacement along the



(a) Displacement in z-direction (middle of tip) using polynomial degree of p=q=2 for different elements

(b) Displacement in z-direction (middle of tip) using polynomial degree of p=q=3 for different elements

Figure 7.21: Displacement in z-direction for solution of problem in Figure 7.17 using "matching patches" geometry shown in Figure 7.18 with $M = M_{\text{max}}$ versus relative penalty factor for rotations. The displacement is strongly coupled for both investigations.

z-direction for an equilibrium accuracy of 10^{-6}. Note that a small discrepancy is introduced by using the penalty approach.

For discussing the influence of the penalty factor on the solution Figure 7.21 shows the z-displacement versus the relative penalty factor for rotations. Here, the displacements at the common boundary are coupled strongly to isolate the effect of rotation coupling. It can be observed that a relative penalty factor larger than 10 gives good results regardless of the polynomial degree. For quantifying the error of rotation at the boundary, the relative difference between the angles at the common edge versus the relative penalty factor is plotted in Figure 7.22. Here, again, the displacement coupling is enforced strongly. It can be observed that by increasing the relative penalty factor, the error decreases uniformly, independent of the polynomial degree. Thus, it can be stated that the higher the penalty factor, the smaller the error will be. Note that too high penalty factors lead to numerical problems (see also Kiendl [46]).

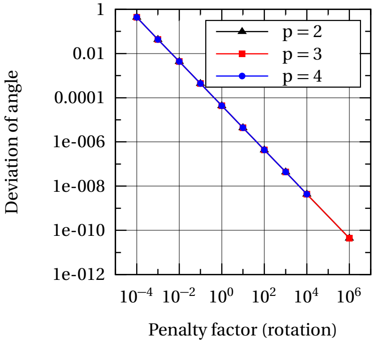


Figure 7.22: Difference of angles (between adjacent boundaries) for solution of the problem in Figure 7.17 using "matching patches" geometry (see Figure 7.18). As in Figure 7.21, in this study the displacement is strongly coupled.

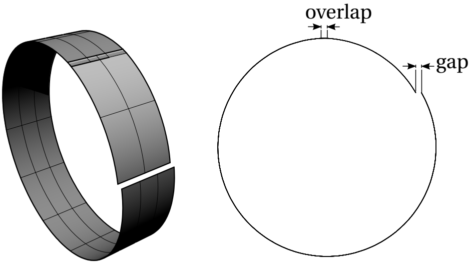


Figure 7.23: IBRA solution of problem in Figure 7.17 using extremely non-watertight model (see "patches with gap" geometry in Figure 7.18)

For the considered problem, Figure 7.22 shows that for a relative penalty factor of 100, the difference in the angles is less than 10^{-6}.

The same behavior can be observed for weak displacement coupling. Thus, a globally defined relative penalty factor of 100 is used within this thesis because it can be considered sufficiently high for achieving good results and not too high to avoid numerical problems.

For practical applications, IBRA needs to deal with *non-watertight* models because almost all CAD models are *non-watertight* as explained in Section 2.2.3. Figure 7.23 shows the solution of an extremely *non-watertight* model (see "patches with gap" geometry in Figure 7.18) for the problem shown in Figure 7.17. For the proposed B-Rep element formulation (see Section 4.5.1), the gaps are simply preserved, as can be seen in Figure 7.23. The displacements and rotations at the master and slave boundaries are the same. Thus, the overlap in Figure 7.23 corresponds to the gap size which at its turn is kept constant throughout the analysis. Note that such a *non-watertight* model is an extreme case and that all analysis techniques (FEA, IGA, FCM, ...) need to interpret such ("dirty") geometries somehow by e.g. adding material and/or preserving the gap.

Figure 7.24 investigates the influence of *non-watertightness* (see also Section 2.2.3) on the analysis result. It shows the "curved-trimmed patches" geometry shown in Figure 7.18 with the *gap-overlap* variable f. The variable is used to describe the size of the gap resp. overlap. A positive value expresses a gap, whereas a negative value indicates an overlap.

Figure 7.25 shows the influence of the *gap-overlap* variable f on the difference in the displacement along the z-direction, which is given by

$$d_1 = |u_{\mathrm{Tip}}(f) - u_{\mathrm{ref}}|, \tag{7.4}$$

where u_{ref} is the z-displacement when $f = 0$. For describing the function d_1, the total length $L = 1.37 + 10.63 + f$ is used as position for applying the moment which additionally introduces a small deviation caused by physics.

In Figure 7.25, it can be seen that the displacement difference in the final result for a small $|f|$, especially within the default CAD tolerance of 0.001 units (see also Section 2.2.3), is negligible. The reason for the different influence of gaps resp. overlaps is shown in Figure 7.24. In this example, given that the coupling technique is based on the *closest point projection*

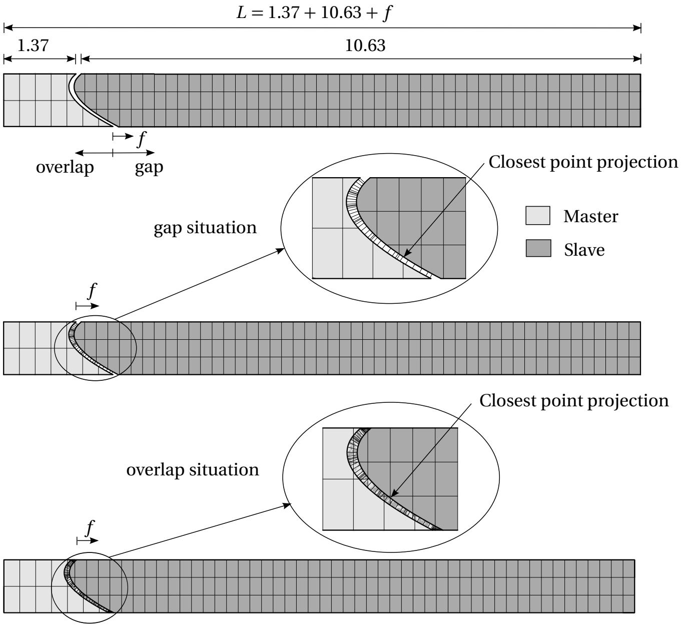


Figure 7.24: Problem description for investigating *non-watertight* (gaps and overlaps) models used in the problem shown in Figure 7.17 with the "curved-trimmed patch" geometry with additional gap description shown in Figure 7.18. The variable f expresses *non-watertightness* in the horizontal direction. For the coupling formulation, the closest point projection is used. The master-slave relationship for the gap and overlap situation is shown in detail.

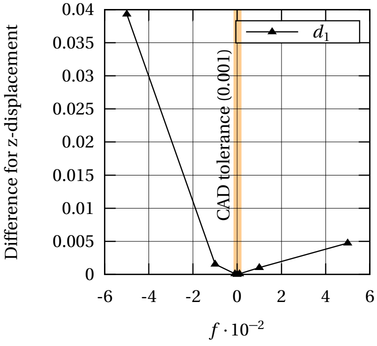


Figure 7.25: Influence of *non-watertightness* on analysis result of problem in Figure 7.17 with $M = M_{\max}$ using "curved-trimmed patch" geometry with additional gap shown in Figure 7.18 to a gap/overlap of f compared with a gap/overlap of $f = 0$. The default CAD tolerance of 0.001 units is represented by the light orange colored bar.

approach (master-slave), the behavior for gaps is better because there is a unique assignment of the quadrature points for the master and slave curves. For the overlaps, the assignment of quadrature points is not unique on the interface.

Summary and conclusions

The conclusion of this example is that IBRA is very flexible from the viewpoint of the geometric description of a problem, regardless of how "stupid" the geometric description may look like. Of course, a simple geometry description is generally better suited for analysis. Nevertheless, IBRA with its *B-Rep elements* provides the framework to handle complex composed and "dirty" geometries in a robust manner as well.

7.6 L-shaped cantilever

The second example of the second part is used for demonstrating that the proposed coupling formulation (see Section 4.5.1) can preserve kinks as well. Since kinks are treated in exactly the same way as the previous exam-

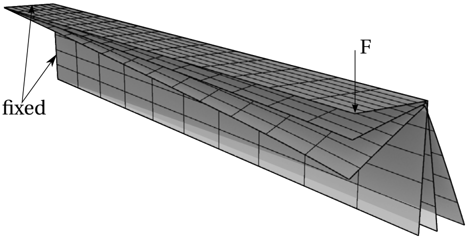


Figure 7.26: IBRA for shell structures with kinks. The kink is preserved by the proposed B-Rep element formulation (see Section 4.5.1)

ple this example is not evaluated quantitatively. An exemplary deformation of an L-shaped cantilever is shown in Figure 7.26.

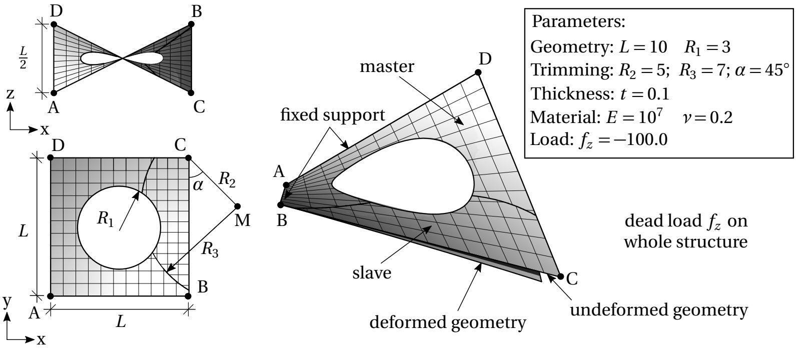
7.7 Doubly curved multi-patch shell with circular hole

The third example of the second part is used for testing the B-Rep element formulation (see Section 4.5.1) for a mechanically complex shell problem. The problem is exactly the same as that presented in Section 7.4, albeit with a different geometry representation. Instead of using a single-patch geometry, two weakly coupled non-matching patches are used (see Figure 7.27).

For this example, the slave patch uses two more elements per direction than the master patch. The displacement of point C in z-direction is shown in Figure 7.28. It can clearly be seen that the multi-patch results converge to the same solution as those for the single-patch (see Figure 7.14(a)).

Summary and conclusions

The conclusion of this example is that the proposed B-Rep element formulation works also for complex nonlinear (doubly curved) shell problems as well producing accurate results regardless of the discretization.



Parameters:
Geometry: $L = 10$ $R_1 = 3$
Trimming: $R_2 = 5$; $R_3 = 7$; $\alpha = 45°$
Thickness: $t = 0.1$
Material: $E = 10^7$ $\nu = 0.2$
Load: $f_z = -100.0$

Figure 7.27: Benchmark example of doubly curved trimmed shell including weak coupling boundary conditions - problem description

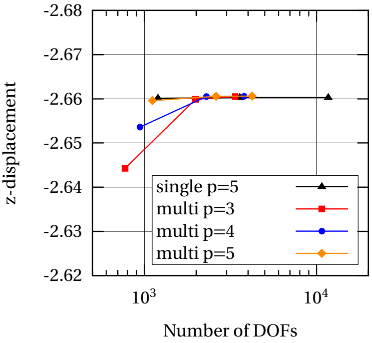


Figure 7.28: Displacement in z-direction for the problem shown in Figure 7.27 using different polynomial degrees. For the coupled multi-patches, the single-patch solution can be used as reference. Therefore, the figure contains the single-patch solution with p=q=5 from Figure 7.14(a) as well.

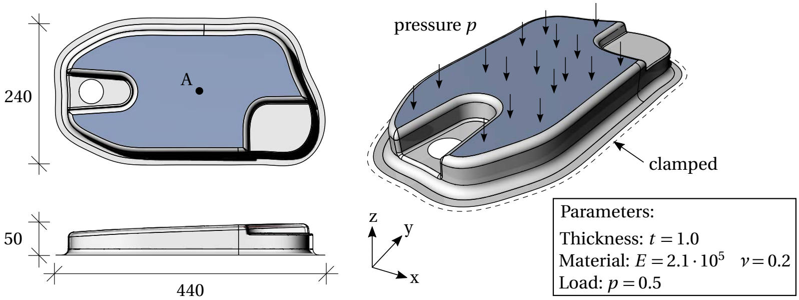
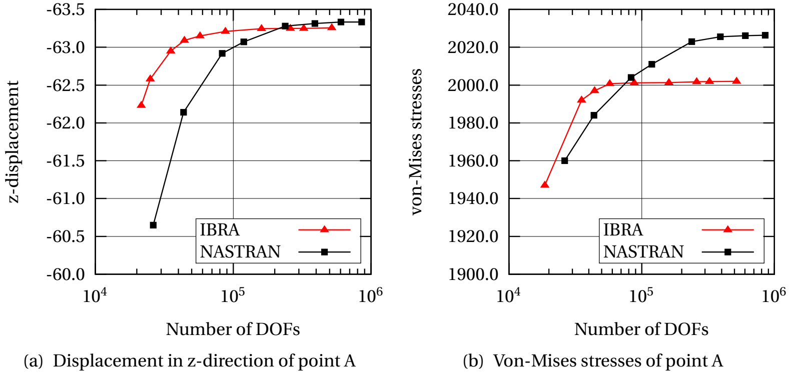


Figure 7.29: Problem description of oil sump example. The blue region is loaded. The Dirichlet boundary conditions is applied to the edge with the dashed curve. The model consist of 50 trimmed surfaces and 120 edges and has been generated with the software *Siemens NX*.

7.8 Industrial application: oil sump

The first industrial example, described in Figure 7.29, deals with the structural analysis of an oil sump. This example is used for demonstrating that IBRA can be used for industrial problems. In addition, a strategy for an uniform geometry refinement for complex geometries is highlighted which is required for complex CAD models. To validate the IBRA results they are compared to solutions obtained by the commercial FE program *NASTRAN NX*.

Figure 7.30 shows the convergence of the von-Mises stresses and the displacement in z-direction for IBRA and the classical FEA. It can clearly be seen that the IBRA solutions with a polynomial degree of $p = q = 3$ converge much faster than the linear basis functions used within classical FEA. As expected the RM shell, denoted with *NASTRAN*, converges towards a slightly higher value than the KL shell used for IBRA. The refinement strategy used for Figure 7.30 is demonstrated exemplary in Figure 7.31. Here the size *units per element* indicates the approximate maximum dimension of each element of the corresponding refined CAD model. Figure 7.32 shows a qualitative comparison of the von-Mises stresses using IBRA and classical FEA. One can see a clear match between the two stress plots.



(a) Displacement in z-direction of point A

(b) Von-Mises stresses of point A

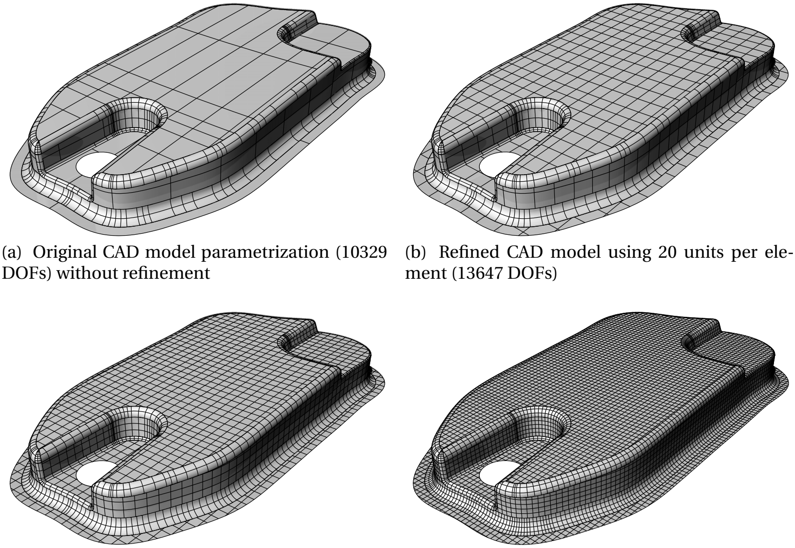
Figure 7.30: Convergence plots for the displacement in z-direction and von-Mises stresses at point A for the problem given in Figure 7.29. The IBRA solution uses a polynomial degree of $p = q = 3$.

Summary and conclusions

IBRA is able to deliver highly accurate solutions also for complex industrial CAD models even with less DOFs. This can be really advantageous for problems where the solution process of the linear system of equation represents the bottleneck. A key feature to be competitive with classical FEA is an uniform refinement strategy i.e. creating elements with uniform dimensions, to avoid unnecessary DOFs. Moreover the robustness of IBRA applied to real CAD models is demonstrated.

7.9 Industrial application: engine bonnet

The last example is used to demonstrate that IBRA can already be used for CAD models which are used as geometrical basis for production lines in industry. The example is described in Figure 7.33 and the corresponding result is illustrated demonstratively in Figure 7.34. The CAD model consist of 428 trimmed surfaces (see also Figure 2.2(b)) and 1224 edges and is created with the CAD program *CATIA* (see Dassault Systemes [28]). The geometry is kindly provided by Daimler AG [27].



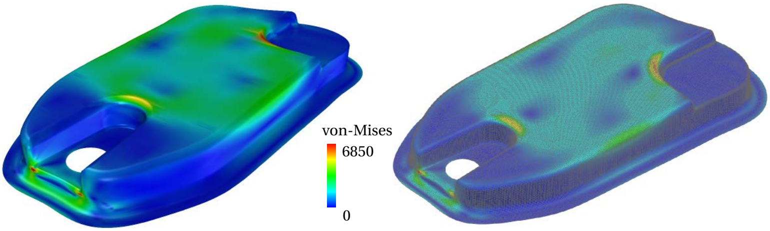
(a) Original CAD model parametrization (10329 DOFs) without refinement

(b) Refined CAD model using 20 units per element (13647 DOFs)

(c) Refined CAD model using 10 units per element (21561 DOFs)

(d) Refined CAD model using 5 units per element (44199 DOFs)

Figure 7.31: Different refinement levels for the CAD model used in Figure 7.29. The size *units per element* indicates the approximate dimensions of each element for the corresponding refined CAD model. In addition, the number of DOFs considering the KL shell formulation is indicated.



(a) IBRA solution (Visualization with TeDA in the CAD program *Rhinoceros*)

(b) Solution obtained by the commercial FE program *NASTRAN NX* within *Siemens NX.*

Figure 7.32: Comparison of von-Mises stress results for the problem given in Figure 7.29 using IBRA and classical FEA.

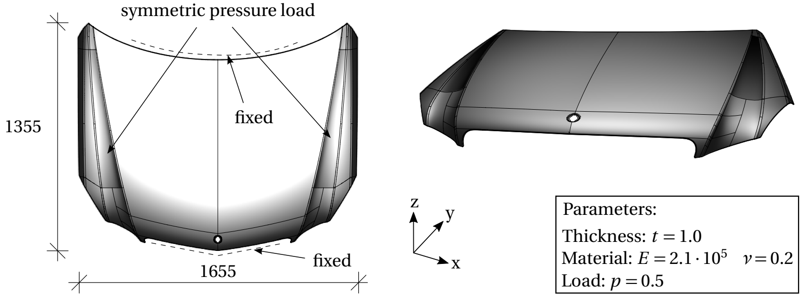


Figure 7.33: Problem description of engine bonnet example. The dark grey patches are loaded. The Dirichlet boundary conditions are applied to the edges with the dashed curves. (CAD model: by courtesy of Daimler AG [27])

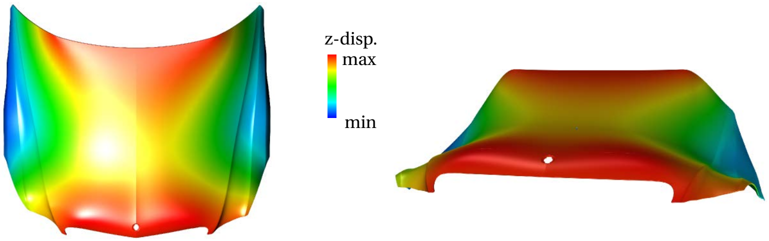


Figure 7.34: Displacement in z-direction for the problem given in Figure 7.33

Summary and conclusion of Chapter 7

The following list contains some important aspects elaborated within this chapter:

- Section 7.2 shows that restricting the *parameter domain* of a surface to a *trimmed domain* and applying an accurate integration scheme neither influences the convergence properties nor the accuracy of the underlying functional basis for analysis purposes. The accuracy is just limited by the trimming tolerance (geometrical modeling error).
- Section 7.3 demonstrates the robustness of IBRA against distortions in the geometric description as well as the good point-wise convergence even on trimmed boundaries.
- Section 7.4 demonstrates the influence of trimming tolerances on analysis results (geometrical modeling error) for nonlinear shell problems.
- Section 7.5 illustrates that IBRA is very flexible and able to deal with "dirty" geometries.
- Section 7.7 shows the accuracy of B-Rep coupling elements for geometrically nonlinear shell problems.
- The examples in Section 7.8 and Section 7.9 demonstrate that IBRA can compete with established analysis approaches.

The various numerical examples including real industrial problems presented in this chapter confirm the accuracy, flexibility, and robustness of IBRA.

There is no reason anyone would want a computer in their home.

Ken Olsen, founder of Digital Equipment Corporation, 1977

Chapter 8
Summary and Conclusion

A new concept of a CAD-integrated design-through-analysis workflow for *Kirchhoff-Love* (KL) shell structures, named *Analysis in Computer Aided Design* (AiCAD), has been developed. AiCAD is realized within the commercial CAD Systems *Siemens NX* (see Siemens PLM Software [84]), *Rhinoceros* (see Rhinoceros [70]), and *GiD* (see GiD [34]). In contrast to other workflows AiCAD uses the CAD geometry description, to be more specific *NURBS-based B-Rep models* (see Section 2.2), throughout the entire workflow.

For analyzing *NURBS-based B-Rep models*, the *isogeometric B-Rep analysis*, a new finite element technique, has been developed as well. IBRA can be seen as an extension of *isogeometric analysis* (IGA) introduced by Hughes et al. [40] by considering the B-Rep topology of the CAD model for the analysis. Thus, it provides the framework for creating a direct and complete analysis model from CAD in a consistent finite-element-like manner. The new type of finite elements called *(isogeometric) B-Rep elements* are used to handle discontinuous and trimmed geometries with gaps and overlaps for structural analysis and to enforce weak Dirichlet and Neumann boundary conditions. Corresponding *B-Rep element* formulations are introduced.

Appropriate solutions for local refinement and the application of arbitrarily located loads within IBRA are presented as well. In addition, a new *adaptive Gaussian integration procedure* (AGIP) for integrating trimmed surfaces is introduced. For applying AGIP the *parameter domain* of trimmed surfaces is subdivided into *integration domains* such that all of them can be handled by following one single procedure. The accuracy of this approach is shown by numerical examples.

Moreover a concept for implementing IBRA within an existing FE-program and realizing AiCAD within an existing CAD system is outlined and thus a new concept of bridging the gap between CAD and FEM is presented.

Various examples clearly demonstrate the power of IBRA. IBRA is flexible for trimming tolerances, robust against geometric distortion, and can handle *non-watertight* CAD models. These features have been demonstrated by the examples, which cover a gamut of problems ranging from simple linear shells to real industrial problems.

The newly developed *isogeometric B-Rep analysis* seems to be general and robust enough to be competitive with established analysis approaches used in industry. Although the general concept has been established successfully, further research is necessary. Some interesting fields are the following:

- development of alternative *B-Rep element formulations* for shell coupling better suited for structural dynamics
- development of *B-Rep element formulations* for further applications (contact, shape optimization, FSI,...)
- defeaturing i.e. neglecting small details of the CAD model by treating them e.g. as rigid objects in oder to concentrate on the mechanically relevant effects and thus to avoid unnecessary computational cost

There is still a long way to go until the proposed techniques are mature enough to use them in daily engineering life but the ingredients for a *CAD-integrated design and analysis workflow for shell structures* are provided by this thesis.

Bibliography

[1] Abel Coll Sans. "Robust volume mesh generation for non-watertight geometries." PhD thesis. Barcelona: CIMNE, May 2014.

[2] ANSYS. *http://www.ansys.com*.

[3] B. M. Irons. "Engineering applications of numerical integrationin stiffness methods." In: *AIAA J.* 4 (1966), pp. 2035–2037.

[4] Y. Basar and W. Krätzig. *Mechanik der Flächentragwerke. Theorie, Berechnungsmethoden, Anwendungsbeispiele.* Braunschweig: Vieweg, 1985. ISBN: 3-528-08685-8.

[5] A. M. Bauer, M. Breitenberger, B. Philipp, R. Wüchner, and K.-U. Bletzinger. "Nonlinear isogeometric spatial Bernoulli beam." In: *Computer Methods in Applied Mechanics and Engineering* 303 (2016), pp. 101–127. DOI: `10.1016/j.cma.2015.12.027`.

[6] Y. Bazilevs, V. M. Calo, Y. Zhang, and T. J. R. Hughes. *Isogeometric Fluid–structure Interaction Analysis with Applications to Arterial Blood Flow*. 2006. DOI: `10.1007/s00466-006-0084-3`.

[7] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, and T. Sederberg. "Isogeometric analysis using T-splines." In: *Computer Methods in Applied Mechanics and Engineering* 199.5-8 (2010), pp. 229–263. DOI: `10.1016/j.cma.2009.02.036`.

[8] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, and G. Scovazzi. "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows." In: *Computer Methods in Applied Mechanics and Engineering* 197.1-4 (2007), pp. 173–201. DOI: `10.1016/j.cma.2007.07.016`.

[9] Y. Bazilevs, M.-C. Hsu, and M. Scott. “Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines.” In: *Computer Methods in Applied Mechanics and Engineering* 249-252 (2012), pp. 28–41. DOI: 10.1016/j.cma.2012.03.028.

[10] T. Belytschko, R. W. Lewis, D. Perić, and R. L. Taylor. “Isogeometric analysis of nearly incompressible solids.” In: *International Journal for Numerical Methods in Engineering* 87.1-5 (2011), pp. 273–288. DOI: 10.1002/nme.3048.

[11] D. J. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, and T. Hughes. “Blended isogeometric shells.” In: *Computer Methods in Applied Mechanics and Engineering* 255 (2013), pp. 133–146. DOI: 10.1016/j.cma.2012.11.020.

[12] D. Benson, Y. Bazilevs, M. Hsu, and T. Hughes. “Isogeometric shell analysis: The Reissner–Mindlin shell.” In: *Computer Methods in Applied Mechanics and Engineering* 199.5-8 (2010), pp. 276–289. DOI: 10.1016/j.cma.2009.05.011.

[13] D. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, and T. Hughes. “Blended isogeometric shells.” In: *Computer Methods in Applied Mechanics and Engineering* 255 (2013), pp. 133–146. DOI: 10.1016/j.cma.2012.11.020.

[14] W. Boehm. “Inserting new knots into B-spline curves.” In: *Computer-Aided Design* 12.4 (1980), pp. 199–201. DOI: 10.1016/0010-4485(80)90154-2.

[15] C. de Boor. *A practical guide to splines.* 5. print. Vol. 27. Applied mathematical sciences. New York: Springer, 1991. ISBN: 0-387-90356-9.

[16] C. de Boor. *A practical guide to splines: With 32 figures.* Rev. ed. Vol. 27. Applied mathematical sciences. New York: Springer, 2001. ISBN: 0-387-95366-3.

[17] M. Breitenberger, A. Apostolatos, B. Philipp, R. Wüchner, and K.-U. Bletzinger. “Analysis in computer aided design: Nonlinear isogeometric B-Rep analysis of shell structures.” In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015), pp. 401–457. DOI: 10.1016/j.cma.2014.09.033.

[18] D. Burkhart, Hamann B., and Umlauf G. “Iso-geometric Finite Element Analysis Based on Catmull-Clark Subdivision Solids.” In: *Comput. Graph. Forum* 29.5 (2010), pp. 1575–1784.

[19] B. K. Choi. *Surface modeling for CAD/CAM*. Vol. 11. Advances in industrial engineering. Amsterdam, New York: Elsevier, 1991. ISBN: 0444884823.

[20] F. Cirak, M. Ortiz, and P. Schröder. “Subdivision surfaces: a new paradigm for thin-shell finite-element analysis.” In: *International Journal for Numerical Methods in Engineering* 47 (2000), pp. 2039–2072.

[21] F. Cirak and Q. Long. *Subdivision shells with exact boundary control and non-manifold geometry*. 2011. DOI: 10.1002/nme.3206.

[22] E. Cohen, T. Lyche, and L. L. Schumaker. “Degree raising for splines.” In: *Journal of Approximation Theory* 46.2 (1986), pp. 170–181. DOI: 10.1016/0021-9045(86)90059-6.

[23] E. Cohen, R. F. Riesenfeld, and G. Elber. *Geometric modeling with splines: An introduction*. Natick and Mass: AK Peters, 2001. ISBN: 978-1568811376.

[24] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric analysis: Toward integration of CAD and FEA*. Chichester et al.: Wiley, 2009. ISBN: 0470748737.

[25] J. Cottrell, T. Hughes, and A. Reali. “Studies of refinement and continuity in isogeometric structural analysis.” In: *Computer Methods in Applied Mechanics and Engineering* 196.41-44 (2007), pp. 4160–4183. DOI: 10.1016/j.cma.2007.04.007.

[26] J. Cottrell, A. Reali, Y. Bazilevs, and T. Hughes. “Isogeometric analysis of structural vibrations.” In: *Computer Methods in Applied Mechanics and Engineering* 195.41-43 (2006), pp. 5257–5296. DOI: 10.1016/j.cma.2005.09.027.

[27] Daimler AG. *https://www.daimler.com/*.

[28] Dassault Systemes. *http://www.3ds.com*.

[29] R. Dimitri, L. de Lorenzis, M. Scott, P. Wriggers, R. Taylor, and G. Zavarise. “Isogeometric large deformation frictionless contact using T-splines.” In: *Computer Methods in Applied Mechanics and Engineering* 269 (2014), pp. 394–414. DOI: `10.1016/j.cma.2013.11.002`.

[30] W. Dornisch, S. Klinkel, and B. Simeon. “Isogeometric Reissner–Mindlin shell analysis with exactly calculated director vectors.” In: *Computer Methods in Applied Mechanics and Engineering* 253 (2013), pp. 491–504. DOI: `10.1016/j.cma.2012.09.010`.

[31] W. Dornisch, R. Müller, and S. Klinkel. “An efficient and robust rotational formulation for isogeometric Reissner–Mindlin shell elements.” In: *Computer Methods in Applied Mechanics and Engineering* 303 (2016), pp. 1–34. DOI: `10.1016/j.cma.2016.01.018`.

[32] A. Düster, J. Parvizian, Z. Yang, and E. Rank. “The finite cell method for three-dimensional problems of solid mechanics.” In: *Computer Methods in Applied Mechanics and Engineering* 197.45-48 (2008), pp. 3768–3782. DOI: `10.1016/j.cma.2008.02.036`.

[33] R. Echter, B. Oesterle, and M. Bischoff. *A hierarchic family of isogeometric shell finite elements.* 2013. DOI: `10.1016/j.cma.2012.10.018`.

[34] GiD. *http://www.gidhome.com/.*

[35] W. J. Gordon and R. F. Riesenfeld. “Bernstein-Bézier Methods for the Computer-Aided Design of Free-Form Curves and Surfaces.” In: *Journal of the ACM* 21.2 (1974), pp. 293–310. DOI: `10.1145/321812.321824`.

[36] P. L. Gould. *Introduction to linear elasticity.* 2nd ed. New York: Springer-Verlag, 1994. ISBN: 0387941002.

[37] G. A. Hansen, R. W. Douglass, and A. Zardecki. *Mesh enhancement: Selected elliptic methods, foundations and applications.* London: Imperial College Press, 2005. ISBN: 1-86094-487-6.

[38] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. “High-performance computing of wind turbine aerodynamics using isogeometric analysis.” In: *Computers & Fluids* 49.1 (2011), pp. 93–100. DOI: `10.1016/j.compfluid.2011.05.002`.

[39] M.-C. Hsu, C. Wang, F. Xu, A. J. Herrema, and A. Krishnamurthy. “Direct immersogeometric fluid flow analysis using B-rep CAD models.” In: *Computer Aided Geometric Design* (2016). DOI: `10.1016/j.cagd.2016.02.007`.

[40] T. Hughes, J. Cottrell, and Y. Bazilevs. “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement.” In: *Computer Methods in Applied Mechanics and Engineering* 194.39-41 (2005), pp. 4135–4195. DOI: `10.1016/j.cma.2004.10.008`.

[41] ISO 10303. *www.WikiSTEP.org: STEP (Standard for the Exchange of Product model data).*

[42] L. Kantorovich and V. Krylov. *Approximate methods of higher analysis.* Interscience Publishers, 1958.

[43] 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.” In: *Computer Methods in Applied Mechanics and Engineering* 199.37-40 (2010), pp. 2403–2416. DOI: `10.1016/j.cma.2010.03.029`.

[44] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. “Isogeometric shell analysis with Kirchhoff–Love elements.” In: *Computer Methods in Applied Mechanics and Engineering* 198.49-52 (2009), pp. 3902–3914. DOI: `10.1016/j.cma.2009.08.013`.

[45] J. Kiendl, R. Schmidt, R. Wüchner, and K.-U. Bletzinger. “Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting.” In: *Computer Methods in Applied Mechanics and Engineering* 274 (2014), pp. 148–167. DOI: `10.1016/j.cma.2014.02.001`.

[46] M. J. Kiendl. *Isogeometric analysis and shape optimal design of shell structures.* Aachen: Shaker, 2011. ISBN: 978-3-8440-0104-4.

[47] H.-J. Kim, Y.-D. Seo, and S.-K. Youn. "Isogeometric analysis for trimmed CAD surfaces." In: *Computer Methods in Applied Mechanics and Engineering* 198.37-40 (2009), pp. 2982–2995. DOI: 10.1016/j.cma.2009.05.004.

[48] E. Klingbeil. *Tensorrechnung für Ingenieure*. Vol. 197/197a. BI-Hochschultaschenbücher. Mannheim [u.a.]: BI-Wiss.-Verl., 1966. ISBN: 978-3411001972.

[49] P. M. Knupp and S. Steinberg. *The fundamentals of grid generation*. Boca Raton [u.a.]: CRC Press, 1994. ISBN: 0-8493-8987-9.

[50] S. Krishnan and D. Manocha. "An efficient surface intersection algorithm based on lower-dimensional formulation." In: *ACM Transactions on Graphics (TOG)* 16.1 (1997), pp. 74–106.

[51] L. Kudela, N. Zander, T. Bog, S. Kollmannsberger, and E. Rank. "Efficient and accurate numerical quadrature for immersed boundary methods." In: *Advanced Modeling and Simulation in Engineering Sciences* 2.1 (2015), p. 763. DOI: 10.1186/s40323-015-0031-y.

[52] L. Kudela, N. Zander, S. Kollmannsberger, and E. Rank. "Smart octrees: Accurately integrating discontinuous functions in 3D." In: *Computer Methods in Applied Mechanics and Engineering* 306 (2016), pp. 406–426. DOI: 10.1016/j.cma.2016.04.006.

[53] G. Kuru, C. Verhoosel, K. van der Zee, and E. van Brummelen. "Goal-adaptive Isogeometric Analysis with hierarchical splines." In: *Computer Methods in Applied Mechanics and Engineering* 270 (2014), pp. 270–292. DOI: 10.1016/j.cma.2013.11.026.

[54] L. de Lorenzis, İ. Temizer, P. Wriggers, and G. Zavarise. *A large deformation frictional contact formulation using NURBS-based isogeometric analysis*. 2011. DOI: 10.1002/nme.3159.

[55] Y. L. Ma and W. Hewitt. "Point inversion and projection for NURBS curve and surface: Control polygon approach." In: *Computer Aided Geometric Design* 20.2 (2003), pp. 79–99. DOI: 10.1016/S0167-8396(03)00021-9.

[56] M. Mäntylä. *An introduction to solid modeling*. Vol. 13. Principles of computer science series. Rockville and Md: Computer Science Press, 1988. ISBN: 0-88175-108-1.

[57] M. E. Mortenson. *Geometric modeling*. New York: Wiley, 1985. ISBN: 0471882798.

[58] A. P. Nagy and D. J. Benson. “On the numerical integration of trimmed isogeometric elements.” In: *Computer Methods in Applied Mechanics and Engineering* (2014). DOI: `10.1016/j.cma.2014.08.002`.

[59] K. L. Narayan, K. M. Rao, and Sarcar, M. M. M. *Computer aided design and manufacturing*. New Delhi: Prentice-Hall of India, 2008. ISBN: 812033342X.

[60] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wüchner, K. Bletzinger, Y. Bazilevs, and T. Rabczuk. “Rotation free isogeometric thin shell analysis using PHT-splines.” In: *Computer Methods in Applied Mechanics and Engineering* 200.47-48 (2011), pp. 3410–3424. DOI: `10.1016/j.cma.2011.08.014`.

[61] J. Parvizian, A. Düster, and E. Rank. *Finite cell method.* 2007. DOI: `10.1007/s00466-007-0173-y`.

[62] N. M. Patrikalakis. “Surface-to-surface intersections.” In: *IEEE Computer Graphics and Applications* 13.1 (1993), pp. 89–95.

[63] B. Philipp, M. Breitenberger, I. D’Auria, R. Wüchner, and K.-U. Bletzinger. “Integrated design and analysis of structural membranes using the Isogeometric B-Rep Analysis.” In: *Computer Methods in Applied Mechanics and Engineering* 303 (2016), pp. 312–340. DOI: `10.1016/j.cma.2016.02.003`.

[64] L. Piegl and W. Tiller. “Geometry-based triangulation of trimmed NURBS surfaces.” In: *Computer Aided Design* 1.30 (1998), pp. 11–18.

[65] L. A. Piegl and W. Tiller. *The NURBS book*. 2nd ed. Berlin, New York: Springer, op. 1997. ISBN: 3-540-61545-8.

[66] X. Qian and O. Sigmund. “Isogeometric shape optimization of photonic crystals via Coons patches.” In: *Computer Methods in Applied Mechanics and Engineering* 200.25-28 (2011), pp. 2237–2255. DOI: `10.1016/j.cma.2011.03.007`.

[67] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. "Geometric modeling, isogeometric analysis and the finite cell method." In: *Computer Methods in Applied Mechanics and Engineering* 249-252 (2012), pp. 104–115. DOI: `10.1016/j.cma.2012.05.022`.

[68] K. Reed, D. Harrod, and W. Conroy. *The Initial Graphics Exchange Specification (IGES) version 5.0*. Vol. 4412. NISTIR. Gaithersburg, MD: National Institute of Standards and Technology, 1990. ISBN: 9781871530049.

[69] G. Renner and V. Weiß. "Exact and approximate computation of B-spline curves on surfaces." In: *Computer-Aided Design* 36.4 (2004), pp. 351–362. DOI: `10.1016/S0010-4485(03)00100-3`.

[70] Rhinoceros. *http://www.rhino3d.com*.

[71] R. F. Riesenfeld, E. Cohen, and Lyche T. "Discrete B-splines und subdivision techniques in computer-aided geometric design and computer graphics." In: *Elsevier* 1980 ().

[72] D. F. Rogers. *An introduction to NURBS: With historical perspective*. San Francisco: Morgan Kaufmann Publishers, 2001. ISBN: 1-55860-669-6.

[73] J. Rooney and P. Steadman. *Computer-aided Design*. Milton Keynes: Open University Press, 1987. ISBN: 0273026720.

[74] Scan&Solve. *http://www.intact-solutions.com*.

[75] D. Schillinger. "The p- and B-spline versions of the geometrically nonlinear finite cell method and hierarchical refinement strategies for adaptive isogeometric and embedded domain analysis." Dissertation. München: Technischen Universität München, 22.May 2012.

[76] D. Schillinger, L. Dedè, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, and T. J. Hughes. "An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces." In: *Computer Methods in Applied Mechanics and Engineering* 249-252 (2012), pp. 116–150. DOI: `10.1016/j.cma.2012.03.017`.

[77] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, and E. Rank. *Small and large deformation analysis with the p- and B-spline versions of the Finite Cell Method.* 2012. DOI: 10.1007/s00466-012-0684-z.

[78] R. Schmidt. *Trimming, mapping, and optimization in isogeometric analysis of shell structures: Zugl.: München, Techn. Univ., Diss., 2013.* Vol. 20. Schriftenreihe des Lehrstuhls für Statik TU München. Aachen: Shaker, 2013. ISBN: 978-3-8440-2310-7.

[79] M. Scott, X. Li, T. Sederberg, and T. Hughes. "Local refinement of analysis-suitable T-splines." In: *Computer Methods in Applied Mechanics and Engineering* 213-216 (2012), pp. 206–222. DOI: 10.1016/j.cma.2011.11.022.

[80] T. W. Sederberg, G. T. Finnigan, X. Li, H. Lin, and H. Ipson. "Watertight trimmed NURBS." In: *ACM Transactions on Graphics* 27.3 (2008), p. 1. DOI: 10.1145/1360612.1360678.

[81] T. W. Sederberg and R. J. Meyers. "Loop detection in surface patch intersections." In: *Computer Aided Geometric Design* 5.2 (1988), pp. 161–171.

[82] T. Sederberg, D. L. Cardon, J. Zheng, and Lyche T. "T-spline Simplication and Local Refinement." In: *ACM Trans. Graphics 22* 23.276-283 (2004).

[83] T. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. "T-splines and T-NURCCs." In: *ACM Trans. Graphics 22* 22.477-484 (2003).

[84] Siemens PLM Software. *http://www.plm.automation.siemens.com.*

[85] Spatial Corporation. *http://www.spatial.com/.*

[86] I. Stroud. *Boundary representation modelling techniques.* London: Springer, 2006. ISBN: 978-1-84628-312-3.

[87] K. Sze, X. Liu, and S. Lo. "Popular benchmark problems for geometric nonlinear analysis of shells." In: *Finite Elements in Analysis and Design* 40.11 (2004), pp. 1551–1569. DOI: 10.1016/j.finel.2003.11.001.

[88] İ. Temizer, P. Wriggers, and T. Hughes. *Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS.* 2012. DOI: 10.1016/j.cma.2011.10.014.

[89] B. H. V. Topping. *Finite element mesh generation.* Saxe-Coburg publications on computational engineering. Stirling and Scotland: Saxe-Coburg Publications, 2004. ISBN: 1-874672-10-5.

[90] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. “A hierarchical approach to adaptive local refinement in isogeometric analysis.” In: *Computer Methods in Applied Mechanics and Engineering* 200.49-52 (2011), pp. 3554–3567. DOI: 10.1016/j.cma.2011.09.004.

[91] W. A. Wall, M. A. Frenzel, and C. Cyron. *Isogeometric structural shape optimization.* 2008. DOI: 10.1016/j.cma.2008.01.025.

[92] Y.-W. Wang, Z.-D. Huang, Y. Zheng, and S.-G. Zhang. “Isogeometric analysis for compound B-spline surfaces.” In: *Computer Methods in Applied Mechanics and Engineering* 261-262 (2013), pp. 1–15. DOI: 10.1016/j.cma.2013.04.001.

[93] Y. Zhang, Y. Bazilevs, S. Goswami, C. L. Bajaj, and T. J. Hughes. *Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow.* 2007. DOI: 10.1016/j.cma.2007.02.009.

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

Bisherige Titel der Schriftenreihe

Band	Titel
1	Frank Koschnick, *Geometrische Lockingeffekte bei Finiten Elementen und ein allgemeines Konzept zu ihrer Vermeidung,* 2004.
2	Natalia Camprubi, *Design and Analysis in Shape Optimization of Shells,* 2004.
3	Bernhard Thomee, *Physikalisch nichtlineare Berechnung von Stahlfaserbetonkonstruktionen,* 2005.
4	Fernaß Daoud, *Formoptimierung von Freiformschalen - Mathematische Algorithmen und Filtertechniken,* 2005.
5	Manfred Bischoff, *Models and Finite Elements for Thin-walled Structures,* 2005.
6	Alexander Hörmann, *Ermittlung optimierter Stabwerkmodelle auf Basis des Kraftflusses als Anwendung plattformunabhängiger Prozesskopplung,* 2006.
7	Roland Wüchner, *Mechanik und Numerik der Formfindung und Fluid-Struktur-Interaktion von Membrantragwerken,* 2006.
8	Florian Jurecka, *Robust Design Optimization Based on Metamodeling Techniques,* 2007.
9	Johannes Linhard, *Numerisch-mechanische Betrachtung des Entwurfsprozesses von Membrantragwerken,* 2009.
10	Alexander Kupzok, *Modeling the Interaction of Wind and Membrane Structures by Numerical Simulation,* 2009.
11	Bin Yang, *Modified Particle Swarm Optimizers and their Application to Robust Design and Structural Optimization,* 2009.

Band	**Titel**
12	Michael Fleischer, *Absicherung der virtuellen Prozesskette für Folgeoperationen in der Umformtechnik*, 2009.
13	Amphon Jrusjrungkiat, *Nonlinear Analysis of Pneumatic Membranes - From Subgrid to Interface*, 2009.
14	Alexander Michalski, *Simulation leichter Flächentragwerke in einer numerisch generierten atmosphärischen Grenzschicht*, 2010.
15	Matthias Firl, *Optimal Shape Design of Shell Structures*, 2010.
16	Thomas Gallinger, *Effiziente Algorithmen zur partitionierten Lösung stark gekoppelter Probleme der Fluid-Struktur-Wechselwirkung*, 2011.
17	Josef Kiendl, *Isogeometric Analysis and Shape Optimal Design of Shell Structures*, 2011.
18	Joseph Jordan, *Effiziente Simulation großer Mauerwerksstrukturen mit diskreten Rissmodellen*, 2011.
19	Albrecht von Boetticher, *Flexible Hangmurenbarrieren: Eine numerische Modellierung des Tragwerks, der Hangmure und der Fluid-Struktur-Interaktion*, 2012.
20	Robert Schmidt, *Trimming, Mapping, and Optimization in Isogeometric Analysis of Shell Structures*, 2013.
21	Michael Fischer, *Finite Element Based Simulation, Design and Control of Piezoelectric and Lightweight Smart Structures*, 2013.
22	Falko Hartmut Dieringer, *Numerical Methods for the Design and Analysis for Tensile Structures*, 2014.
23	Rupert Fisch, *Code Verification of Partitioned FSI Environments for Lightweight Structures*, 2014.
24	Stefan Sicklinger, *Stabilized Co-Simulation of Coupled Problems Including Fields and Signals*, 2014.

Band	Titel
25	Madjid Hojjat, *Node-based parametrization for shape optimal design*, 2015.
26	Ute Israel, *Optimierung in der Fluid-Struktur-Interaktion - Sensitivitätsanalyse fu ür die Formoptimierung auf Grundlage des partitionierten Verfahrens*, 2015.
27	Electra Stavropoulou, *Sensitivity analysis and regularization for shape optimization of coupled problems*, 2015.
28	Daniel Markus, *Numerical and Experimental Modeling for Shape Optimization of Offshore Structures*, 2015.
29	Pablo Suárez, *Design Process for the Shape Optimization of Pressurized Bulkheads as Components of Aircraft Structures*, 2015.
30	Armin Widhammer, *Variation of Reference Strategy - Generation of Optimized Cutting Patterns for Textile Fabrics*, 2015.
31	Helmut Masching, *Parameter Free Optimization of Shape Adaptive Shell Structures*, 2016.
32	Hao Zhang, *A General Approach for Solving Inverse Problems in Geophysical Systems by Applying Finite Element Method and Metamodel Techniques*, 2016.
33	Tianyang Wang, *Development of Co-Simulation Environment and Mapping Algorithms*, 2016.