Technische Universität München

Lehrstuhl für Computation in Engineering

The *p*- and B-spline versions of the geometrically nonlinear finite cell method and hierarchical refinement strategies for adaptive isogeometric and embedded domain analysis

Dominik Schillinger

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

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Kai-Uwe Bletzinger

Prüfer der Dissertation:

1. Univ.-Prof. Dr.rer.nat. Ernst Rank
2. Prof. Manolis Papadrakakis, Ph.D.,
   National Technical University of Athens / Griechenland
3. Univ.-Prof. Dr.-Ing. habil. Manfred Bischoff,
   Universität Stuttgart

Die Dissertation wurde am 9. Dezember 2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingenieur- und Vermessungswesen am 22. Mai 2012 angenommen.

Gewidmet in Liebe und Dankbarkeit

meiner Partnerin Anna
meiner Mutter Rosa

Zusammenfassung

Die vorliegende Dissertation befasst sich mit zwei neuen numerischen Verfahren, die die Anbindung geometrischer Modelle an die Finite-Elemente-Analyse erleichtern sollen: die Methode der Finiten Zellen und die Isogeometrische Analyse.

Im Kontext der *Methode der Finiten Zellen* (FCM) wird der bisherige Entwicklungsstand um vier Aspekte erweitert: Erstens wird die FCM unabhängig von einem bestimmten Ansatzraum als allgemeines Konzept zur Geometrie-Analyse-Integration vorgestellt. Zweitens wird die B-spline Version der FCM eingeführt, die B-Splines hoher Ordnung und hoher Glattheit im Rahmen des allgemeinen FCM-Konzepts anwendet. Drittens wird die Idee des "Deformation resetting" motiviert, das die Lösung geometrisch nichtlinearer Probleme im Rahmen der FCM erlaubt. Viertens wird die FCM mit dem Ansatz der hierarchischen *hp-d*-Adaptivität kombiniert, um lokale Netzverfeinerung zu ermöglichen.

Im Kontext der *Isogeometrischen Analyse* (IGA) wird aufbauend auf dem Subdivisionsprinzip ein hierarchisches Verfeinerungsschema für nicht-uniforme rationale B-Splines (NURBS) motiviert, das volle Analysetauglichkeit in 2D und 3D mit pragmatisch-intuitiver Implementierung in Baumstrukturen verbindet und somit eine geeignete Basis für lokale Netzverfeinerung in der NURBS-basierten IGA darstellt. Es wird gezeigt, dass die Kombination von hierarchischer Verfeinerung und der B-Spline Version der FCM eine nahtlose CAD-Analyse-Integration von komplexen T-Spline basierten Bauteilen erlaubt, deren lokale Eigenschaften hinsichtlich Geometrie und Lösungsfeldern adaptiv berücksichtigt werden können.

Abstract

The thesis at hand addresses two recently introduced concepts intended to support an efficient interaction between geometrical models and finite element analysis: the finite cell method and isogeometric analysis.

In the context of the *finite cell method* (FCM), the state of the art is complemented by four new aspects: First, the FCM is generalized as a seamless geometry-through-analysis procedure, independent of a special high-order basis. Second, the B-spline version of the FCM is introduced, which applies high-order and high-continuity B-spline bases within the FCM concept. Third, the idea of deformation resetting is motivated, which allows for the extension of the finite cell method to geometrically nonlinear problems. Fourth, the FCM is combined with hierarchical *hp-d* adaptivity for local mesh refinement.

In the context of *isogeometric analysis* (IGA), the principle of subdivision is used to derive a hierarchical refinement procedure for non-uniform rational B-splines (NURBS). It combines full analysis suitability in 2D and 3D with straightforward implementation in tree data structures and thus constitutes a suitable basis for local refinement in NURBS based IGA. It is furthermore demonstrated that the combination of the B-spline version of the FCM and hierarchical refinement of NURBS opens the door for a seamless IGA design-through-analysis procedure for complex T-spline based engineering parts and assemblies, which is able to adaptively resolve local features in geometry and solution fields.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit am Lehrstuhl für Computation in Engineering der Technischen Universität München im Zeitraum Januar 2009 bis November 2011. Sie wurde ermöglicht durch ein Promotionsstipendium im Rahmen des interdisziplinären Forschungsclusters MAC (Munich Centre of Advanced Computing) sowie der Graduiertenschule IGSSE (International Graduate School of Science and Engineering).

Mein besonderer Dank gilt zuerst meinem geschätzten Doktorvater Prof. Ernst Rank. Seine rückhaltlose und großzügige Unterstützung aller meiner Vorhaben und Ideen sowie der beträchtliche Freiraum, den ich während der letzten drei Jahre genießen durfte, haben ganz wesentlich zum Gelingen dieser Arbeit und zu meiner persönlichen Entwicklung als Wissenschaftler beigetragen.

Besonders bedanken möchte ich mich auch bei Prof. Alexander Düster, mit dem ich nach seiner Berufung an die Technische Universität Hamburg-Harburg kurz nach Beginn meiner Tätigkeit in engem Kontakt bleiben durfte. Von seinem stetigen Interesse, seiner großen Erfahrung mit der FCM und seinen wertvollen Anregungen haben die Kapitel 2 bis 5 ganz wesentlich profitiert. Dank gebührt ihm auch für seine herzliche Gastfreundschaft während meines Forschungsaufenthalts in seiner Hamburger Gruppe im November 2010.

Besonderer Dank geht auch an Prof. Tom Hughes für die selbstverständliche Aufnahme in seine Gruppe während meines dreimonatigen Forschungsaufenthalts am Institute for Computational Engineering and Sciences der University of Texas at Austin im Sommer 2011. Der Einfluss von Tom und seiner Gruppe findet sich ganz deutlich in den Kapiteln 6 und 7, die während dieser Zeit entstanden sind.

Weiterer Dank geht an Prof. Yuri Bazilevs, Prof. Zohar Yosibash und Prof. Manolis Papadrakakis, die während der letzten drei Jahre wertvolle Anregungen zur vorliegenden Arbeit gegeben haben. Letzterem gebührt ebenfalls besonderer Dank für seine herzliche Gastfreundschaft während des Institutsseminars auf Rhodos im Sommer 2009 und während meines Forschungsaufenthalts an der NTU Athen im Februar 2011 sowie für die Übernahme eines der Gutachten zu dieser Arbeit.

Mein herzlicher Dank gilt ebenfalls meinem früheren Lehrer Prof. Manfred Bischoff für das Interesse an dieser Arbeit und die Übernahme eines der Gutachten. Es freut mich sehr, einen "Stuttgarter" dabei zu haben, denn ohne die hervorragende Diplomausbildung in Mechanik, Mathematik und Numerik, die ich dort am Institut für Baustatik und Baudynamik, am Institut für Mechanik im Bauwesen, am Lehrstuhl für Hydromechanik und Hydrosystemmodellierung sowie am Institut für Angewandte Analysis und Numerische Simulation genießen durfte, wäre die vorliegende Arbeit so nicht möglich gewesen. Nicht vergessen möchte ich hier Prof. Kai-Uwe Bletzinger, dem ich für die Übernahme des Prüfungsvorsitzes danke.

Bedanken möchte ich mich auch bei all den Kollegen an der TUM, der UT Austin und der NTU Athen, mit denen ich während der letzten drei Jahre in verschiedener Form zusammen-

arbeiten durfte. Besonders zu nennen sind hier Janos Benk, Tino Bog, Mike Borden, Quanji Cai, Dr. Luca Dedè, Dr. John Evans, David Franke, Jérôme Frisch, Dr.-Ing. Stefan Kollmannsberger, Jovana Knežević, Panayiotis Metsis, Dr. rer.nat. Ralf-Peter Mundani, Dr. Vissarion Papadopoulos, Iason Papaioannou, Dr.-Ing. Martin Ruess, Dr. Mike Scott, Christian Sorger, Atanas Stavrev, Dimitar Stefanov, Vasco Varduhn, Dr.-Ing. Zhengxiong Yang und Nils Zander. Nicht vergessen möchte ich Hanne Cornils und Judith O'Meara, die mir während der letzten drei Jahre mit großem persönlichen Engagement über zahlreiche organisatorische Hürden hinweggeholfen haben.

Schließlich möchte ich mich beim gesamten Lehrstuhl für Computation in Engineering sowie dem Fachgebiet für Computergestützte Modellierung und Simulation für das angenehme Arbeitsklima und die freundliche Atmosphäre bedanken, die den 3. Stock im Gebäude 1 der TUM immer zu einem angenehmen Ort gemacht haben.

Dominik Schillinger
München, im November 2011

Contents

1 Introduction **1**

2 The finite cell method for high-order embedded domain analysis **5**
2.1 A generalized geometry-through-analysis concept 5
2.1.1 The fictitious domain approach 6
2.1.2 Adaptive integration 7
2.1.3 Weak imposition of unfitted Dirichlet constraints 8
2.1.4 High-order approximation bases 9
2.2 High-order p-version and B-spline bases 9
2.2.1 The p-version of the finite cell method 9
2.2.2 The B-spline version of the finite cell method 12
2.3 Characteristic solution behavior 13
2.3.1 Smooth extension of solution fields 14
2.3.2 High rates of convergence in energy norm 15
2.4 Numerical examples at small strains 16
2.4.1 Plate with a circular hole 16
2.4.2 Ring with unfitted Dirichlet constraints 22
2.4.3 A proximal femur bone with inhomogeneous material 25

3 The finite cell method for geometrically nonlinear problems **28**
3.1 A geometrically nonlinear formulation in principal directions 28
3.1.1 Kinematics 29
3.1.2 Constitutive equations 30
3.1.3 Discretization and linearization 30
3.2 The standard geometrically nonlinear finite cell formulation 32
3.3 A modified formulation based on repeated deformation resetting 34
3.3.1 Motivation and implementation 34
3.3.2 High rates of convergence in energy measure 36
3.4 The penalty method for the weak imposition of Dirichlet constraints 37
3.4.1 Penalty vs. Nitsche's method 37
3.4.2 Discretization and linearization 38

4 Large deformation analysis with the finite cell method **40**
4.1 Standard benchmarks in two and three dimensions 40
4.1.1 Plate with a circular hole 40
4.1.2 Cube with a spherical hole 44

4.2 Unfitted Dirichlet constraints and robustness under severe mesh distortion . . 45
4.2.1 Ring with weakly enforced boundary conditions 45
4.2.2 Beam under axial torsion 48
4.3 FCM as a seamless geometry-through-analysis procedure for voxel models . . . 53
4.3.1 Efficient generation of adaptive sub-cells from voxels 53
4.3.2 Nonlinear analysis of a metal foam sample 56
4.4 Software implementation and computational performance 58
4.5 Comparison of the p- and B-spline versions of the FCM 60

5 Hierarchical *hp-d* adaptivity in the finite cell method 61
5.1 The hierarchical *hp-d* method 62
5.1.1 Linear overlays for the p-version basis 62
5.1.2 Subdivision overlays for high-order B-splines 65
5.2 Analysis with the *hp-d* adaptive p-version of the FCM 69
5.2.1 Plate with a circular inclusion 70
5.2.2 The L-shaped domain 71
5.3 Analysis with the *hp-d* adaptive B-spline version of the FCM 73
5.3.1 Plate with a circular inclusion 73
5.3.2 Cube with a spherical inclusion 75
5.3.3 An inter-penetrating phase composite 77

6 Hierarchical refinement of non-uniform rational B-splines (NURBS) 80
6.1 B-spline and NURBS basis functions 81
6.1.1 Univariate B-splines 81
6.1.2 Multivariate B-splines 82
6.1.3 Non-uniform rational B-splines 83
6.2 A concept for hierarchical refinement based on B-spline subdivision 84
6.2.1 Refinability of B-spline basis functions by subdivision 84
6.2.2 Construction of adaptive hierarchical approximation spaces 86
6.2.3 A simple model problem in 1D 90
6.2.4 Condition number and sparsity of the stiffness matrix 92
6.3 Hierarchical refinement of axis-aligned B-splines in multiple dimensions 93
6.3.1 Transition from the 1D concept to multiple dimensions 93
6.3.2 Geometry representation and hierarchical mapping 95
6.3.3 Efficient implementation in tree data structures 96
6.4 Hierarchical refinement of NURBS for arbitrary geometries 98

7 Adaptive analysis based on local hierarchical refinement of NURBS 103
7.1 Error estimation and automatic refinement 103
7.2 Adaptive isogeometric analysis based on cuboidal geometries 104
7.2.1 Advection skew to the mesh in 2D 104
7.2.2 Advection skew to the mesh in 3D 107
7.2.3 Lid driven cavity 108
7.3 Adaptive isogeometric analysis based on NURBS geometries 109
7.3.1 The pinched cylindrical shell 110
7.3.2 Advection-diffusion in an annular section 113

7.3.3 Advection-diffusion in a rotating cylinder 114
7.4 Hierarchical refinement, the finite cell method, and T-spline CAD surfaces . . 114
7.4.1 Free vibration analysis of a ship propeller 116
7.4.2 Stress-displacement analysis of a rim of a car wheel 121

8 Summary and conclusions **124**

Bibliography **127**

Chapter 1

Introduction

The finite element method (FEM) is one of the most important tools for the numerical solution of partial differential equations, which are used in many engineering disciplines to simulate technical systems and processes. Finite element analysis (FEA) with standard tools requires the discretization of the physical domain into a finite element mesh, whose boundaries conform to the boundaries of the geometry. While this constraint can be tackled today with the help of powerful mesh generators, it still constitutes a severe bottleneck, when highly complex geometries are at issue. This is illustrated by a recent statement of Ted Blacker, Manager of Simulation Sciences at Sandia National Laboratories, who reports in [56] that in the context of computer aided design (CAD), common industrial experience attributes 80% of the overall analysis time to the *transfer* of a complex geometrical model into an analysis suitable discretization and only 20% to finite element analysis itself. For geometrical models beyond CAD such as volumetric voxel representations derived from medical imaging technologies, the effort to create a suitable finite element discretization can be even much larger. Decreasing the high ratio between discretization effort and actual analysis time would considerably reduce the cost of FEA, which could help to further promote and establish large-scale numerical simulation for complex physical problems in many industrial, medical or scientific applications. Therefore, the question of a more efficient interaction between geometrical models and finite element analysis has gained tremendous interest over the last decade, and several promising concepts to alleviate the discretization challenge for complex geometries have recently been developed in the computational mechanics community. The thesis at hand focuses on two of them: the finite cell method and isogeometric analysis.

The finite cell method (FCM) introduced by Parvizian, Düster and Rank [150] belongs to the class of embedded domain methods [129, 141], also known as immersed boundary [138, 152] or fictitious domain methods [59, 94, 155]. Their main idea consists of the extension of the physical domain of interest beyond its potentially complex boundaries into a larger embedding domain of simple geometry, which can be meshed more easily. The fictitious domain approach and very similar ideas have been widely used in conjunction with penalty methods [33, 155], the mortar approach [10], Lagrange multipliers [39, 81, 92, 94, 101], Nitsche's method [40], extended finite elements [24, 91, 102, 195], discontinuous Galerkin [16] and spectral methods [124, 130, 149] as well as in the immersed [167, 173, 219] and unaligned finite element methods [225] to address problems of structural analysis, acoustics, fluid and heat flow, fluid-structure interaction, topology and shape optimization. The finite cell method combines the

fictitious domain approach with the p-version of the finite element method [197, 198], adaptive integration and weak imposition of unfitted Dirichlet boundary conditions [84, 207, 218]. For problems with a smooth solution, it offers the following key advantages:

- It circumvents the whole mesh generation process by maintaining a structured grid of cells, which are disconnected from the geometry. The geometry information is taken into account implicitly during numerical integration by penalizing material parameters at integration points that are located outside the physical geometry.
- It can accommodate geometries of arbitrary complexity, while the corresponding algorithms are simple and allow for reliable automation of the discretization process.
- Despite its simplified geometry handling, the finite cell method maintains high rates of convergence and thus guarantees high accuracy at a moderate computational effort.

Within the framework of the finite cell method, the following aspects have been examined so far: Analysis of complex structures [162, 168], fluid-structure interaction [209], topology optimization [151], thin-walled structures [163], weak boundary conditions [168, 207, 218], elasto-plasticity [1] and advection-diffusion problems [42]. The present work contributes four main new aspects: First, the finite cell method is characterized as a general geometry-through-analysis procedure, which is independent of a special high-order approximation basis. Second, this viewpoint is corroborated by the application of high-order and high-continuity B-splines within the general finite cell concept, coined the B-spline version of the FCM, for which equivalent overall solution characteristics with respect to the original p-version of the FCM are demonstrated. Third, the formulation of the finite cell method is extended to geometrically nonlinear problems, based on the deformation resetting idea. Fourth, the finite cell method is combined with the concept of *hp-d* adaptivity, which provides the possibility for local mesh refinement, while preserving the simple meshing property of the finite cell concept.

Isogeometric analysis (IGA) was introduced in a pioneering contribution [109] by Hughes and co-workers to bridge the gap between computer aided design and finite element analysis. Its core idea is to use the same basis functions for the representation of geometry in CAD and the approximation of solutions fields in FEA. This strategy bypasses the whole mesh generation process and supports a tightly connected interaction between CAD and FEA [54, 115, 183]. In addition, analysis based on geometric representations of higher-order smoothness exhibits many advantages over standard C^0-continuous finite element representations [79]. This has been shown in many applications such as structural vibrations [55, 56], incompressibility [9, 77, 199], shells [27, 75, 117], fluid-structure interaction [20, 23], turbulence [6, 18, 21], phase-field analysis [96, 35], contact [200, 201], fracture [206] and optimization [60, 211].

Due to their relative simplicity and ubiquity in today's CAD tools, isogeometric analysis has been largely based on non-uniform rational B-splines (NURBS). However, a multivariate NURBS basis does not provide a natural possibility for truly local mesh refinement [22, 56]. Due to its rigid tensor product structure, refinement of NURBS is a global process that propagates throughout the domain. The demand for adaptivity has recently expedited a shift from NURBS to T-spline based isogeometric analysis, where local mesh refinement can be naturally achieved [22, 69, 126, 185]. At the same time, prominent CAD tools such as Rhino or SolidWorks [166, 191, 205] have started to introduce T-spline geometry representations, since

T-splines exhibit further important geometric advantages over NURBS, such as the capability of joining patches without gaps and the preservation of higher-order continuity throughout a multi-patch model [186, 187]. A further geometry representation, which naturally accommodates local mesh refinement, are the so-called PHT-splines [63, 125], whose application for local refinement in isogeometric analysis has been recently presented in [143, 144].

While T-splines are recent developments, the concept of hierarchical refinement of B-splines dates back to the late 80's, when it was introduced by Forsey and Bartels for local surface refinement in CAD [88, 89] and later adopted by Höllig and co-workers for local mesh refinement in B-spline finite elements [105, 106, 119]. In the framework of isogeometric analysis, local hierarchical refinement of NURBS combines several important advantages:

- Hierarchical B-splines rely on the intuitive principle of B-spline subdivision [212, 226], which automatically ensures linear independence of the refined basis.
- Since hierarchical B-splines still rely on a local tensor product structure, they can be easily generalized to arbitrary dimensions. Particular interest lies in 3D, where T-splines become extremely complicated. The rigidity and simplicity of the tensor product structure also considerably facilitates automation of the refinement process.
- A hierarchical organization of a basis can be directly transferred into tree-like data structures, which are a well-established concept in computer science [14, 41, 171, 172] and allow for a straightforward implementation with manageable coding efforts.

Exploiting the benefits of hierarchical refinement, the present work contributes to the design of suitable approximation bases and corresponding algorithms for adaptive B-spline and NURBS based analysis. In particular, the subdivision split of B-spline basis functions is combined with the concept of *hp-d* adaptivity, which leads to a simple, easy-to-implement, yet efficient hierarchical refinement strategy for B-splines and NURBS. Its versatility and flexibility is demonstrated in adaptive isogeometric and embedded domain analysis.

The present work is organized as follows: **Chapter 2** provides a concise introduction to the general finite cell concept as a seamless geometry-through-analysis procedure and establishes the *p*- and B-spline versions of the FCM, highlighting the different derivation and characteristics of their high-order bases. Using linear elastic problems in one, two and three dimensions, the equivalent solution behavior of the *p*- and B-spline versions of the FCM is demonstrated, comprising for example the smooth extension of solution fields, high rates of convergence and simple handling of very complex geometries. Turning to geometrically nonlinear problems, **Chapter 3** introduces a FCM formulation based on the logarithmic strain measure and demonstrates that a standard FCM approach applying the same kinematics over the complete domain leads to the loss of uniqueness of the nonlinear deformation map within the fictitious domain. It motivates its modification by the idea of deformation resetting, which assumes the initial reference configuration within the fictitious domain after each Newton iteration. Furthermore, it provides a geometrically nonlinear formulation of the penalty method for the imposition of unfitted Dirichlet constraints in elements cut by the geometric boundary. **Chapter 4** presents a range of numerical benchmarks, which demonstrate stability, accuracy, physical consistency and high rates of convergence for the modified geometrically nonlinear FCM formulation applied within the *p*- and B-spline versions. It addresses the versatility of the

geometrically nonlinear penalty method, the behavior of the FCM under severe mesh distortion, and presents an application oriented large deformation analysis of a complex metal foam sample. The latter demonstrates the capability of the FCM to directly operate on volumetric voxel representations, leading to a seamless geometry-through-analysis procedure for complex voxel based geometrical models. Furthermore, aspects of software implementation, hardware requirements and computational performance are discussed and a concise comparison between the *p*- and B-spline versions of the FCM is drawn.

Chapter 5 introduces the concept of *hp-d* adaptivity in the framework of the finite cell method. It allows for local mesh refinement to take into account non-smooth properties of a solution, for example discontinuities along material interfaces or singular points. While the *hp-d*-adaptive *p*-version is based on overlays of linear hierarchical functions, the *hp-d*-adaptive B-spline version uses overlays of successively contracted subdivision B-splines. The latter is an important prerequisite for the efficient and easy-to-handle local refinement of NURBS discussed in **Chapter 6**. After a brief review of B-spline and NURBS basis functions, a hierarchical refinement scheme for axis-aligned B-splines is presented, drawing on ideas of subdivision B-splines, the *hp-d* adaptivity and established hierarchical refinement of standard nodal based FEA. Several aspects, such as extension to arbitrary dimensions, efficient implementation and algorithms and generalization to the hierarchical refinement of NURBS for the representation of arbitrary geometries are discussed in detail. **Chapter 7** demonstrates the validity, versatility and efficiency of hierarchical refinement of NURBS for a range of two- and three-dimensional examples from fluid and structural mechanics. In addition, the potential of embedded domain analysis as a seamless IGA design-through-analysis procedure for complex engineering parts and assemblies is outlined for the examples of a ship propeller and a rim of a car wheel, analyzed with the B-spline version of the finite cell method. In this context, hierarchical refinement is demonstrated to considerably increase the flexibility of embedded domain analysis by adaptively resolving local features in geometry and solution fields. **Chapter 8** closes the discussion with a concise summary and some conclusions.

Chapter 2

The finite cell method for high-order embedded domain analysis

The fundamental ideas of the finite cell method (FCM) were introduced in [74, 150] as an extension to the p-version of the finite element method. In this work, the finite cell method is characterized as an embedded domain method, which combines the fictitious domain approach with a high-order approximation basis, adaptive integration, and weak imposition of unfitted Dirichlet boundary conditions. Its core benefit is the possibility to maintain a structured, axis-aligned grid of cuboidal elements irrespective of the geometric complexity involved, while its high-order approximation basis offers high rates of convergence for smooth problems. In contrast to the pioneering FCM contributions [74, 150], we thus perceive herein the finite cell method as a generalized geometry-through-analysis procedure, which is not rigidly connected to the integrated Legendre basis of the p-version of the finite element method, but can be combined with any high-order basis favored by the analyst.

The present chapter starts with a brief review of the components of the finite cell method in the framework of linear elasticity. We then shortly outline the basics of high-order approximation spaces based on the p-version of the finite element method and high-order B-splines, highlighting the different derivation and characteristics of their basis functions, and subsequently combine them with the generalized FCM concept. In the context of this work, the two resulting FCM schemes are coined the p- and B-spline versions of the finite cell method. Using linear elastic problems first, we demonstrate the equivalent solution behavior of the two FCM versions in terms of the smooth extension of solution fields, exponential rates of convergence, in conjunction with Nitsche's method and for a complex three-dimensional proximal femur bone with inhomogeneous material.

2.1 A generalized geometry-through-analysis concept

In the following, we introduce the main components of the finite cell method in the framework of linear elasticity, i.e. the fictitious domain approach, adaptive integration, weak imposition of Dirichlet constraints and high-order approximation bases. We keep the discussion concise and provide links to the pertinent literature for further details beyond the FCM context.

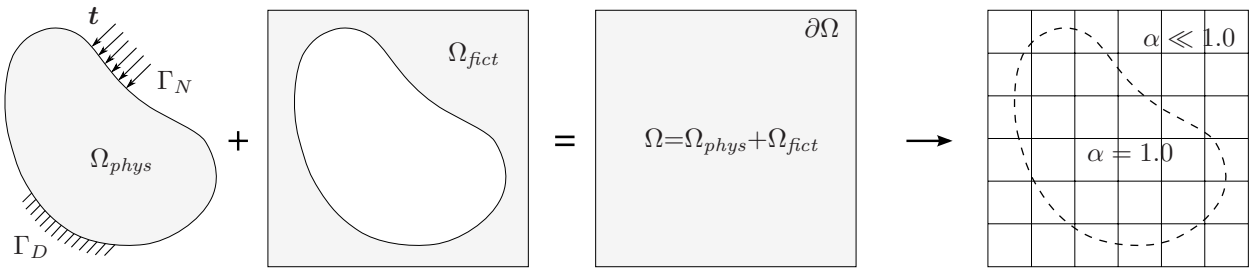


Figure 2.1: The fictitious domain approach: The physical domain Ω_{phys} is extended by the fictitious domain Ω_{fict} into an embedding domain Ω to allow easy meshing of complex geometries. The influence of Ω_{fict} is penalized by α.

2.1.1 The fictitious domain approach

Figure 2.1 provides a concise illustration of the fictitious domain concept. The embedding domain Ω consists of the physical domain of interest Ω_{phys} and the fictitious domain extension Ω_{fict}. Analogous to standard finite element methods, the finite cell method for linear elastic problems is derived from the principle of virtual work

$$\delta W\left(\boldsymbol{u}, \delta\boldsymbol{u}\right) = \int_{\Omega} \boldsymbol{\sigma} \;:\; \left(\nabla_{sym}\, \delta\boldsymbol{u}\right)\; dV - \int_{\Omega_{phys}} \delta\boldsymbol{u} \cdot \boldsymbol{b}\; dV - \int_{\Gamma_N} \delta\boldsymbol{u} \cdot \boldsymbol{t}\; dA \;\; = \;\; 0 \tag{2.1}$$

where $\boldsymbol{\sigma}$, $\boldsymbol{b}$, $\boldsymbol{u}$, $\delta\boldsymbol{u}$ and ∇_{sym} denote the Cauchy stress tensor, body forces, displacement vector, test function and the symmetric part of the gradient, respectively [17, 83, 108, 222]. Neumann boundary conditions are specified over the boundary of the embedding domain $\partial\Omega$, where tractions are zero by definition, and over Γ_N of the physical domain by traction vector $\boldsymbol{t}$ (see Fig. 1). The elasticity tensor $\boldsymbol{C}$ [17, 108, 222] relating stresses and strains

$$\boldsymbol{\sigma} = \alpha\boldsymbol{C} \;:\; \boldsymbol{\varepsilon} \tag{2.2}$$

is complemented by a scalar factor α, which leaves the material parameters unchanged in the physical domain, but penalizes the contribution of the fictitious domain

$$\alpha\left(\boldsymbol{x}\right) = \begin{cases} 1.0 & \forall \boldsymbol{x} \in \Omega_{phys} \\ 10^{-q} & \forall \boldsymbol{x} \in \Omega_{fict} \end{cases} \tag{2.3}$$

In Ω_{fict}, α must be chosen as small as possible, but large enough to prevent extreme ill-conditioning of the stiffness matrix [74, 150]. Typical values of α range between 10^{-4} and 10^{-15}. The idea of applying a penalized material for void regions of a domain has also been frequently used in optimization, see for example [7, 141].

Using a structured grid of high-order elements (see Figure 2.1), which will be called finite cells in the following, kinematic quantities are discretized as

$$\boldsymbol{u} = \sum_{a=1}^{n} N_a \boldsymbol{u}_a \tag{2.4}$$

$$\delta\boldsymbol{u} = \sum_{a=1}^{n} N_a \delta\boldsymbol{u}_a \tag{2.5}$$

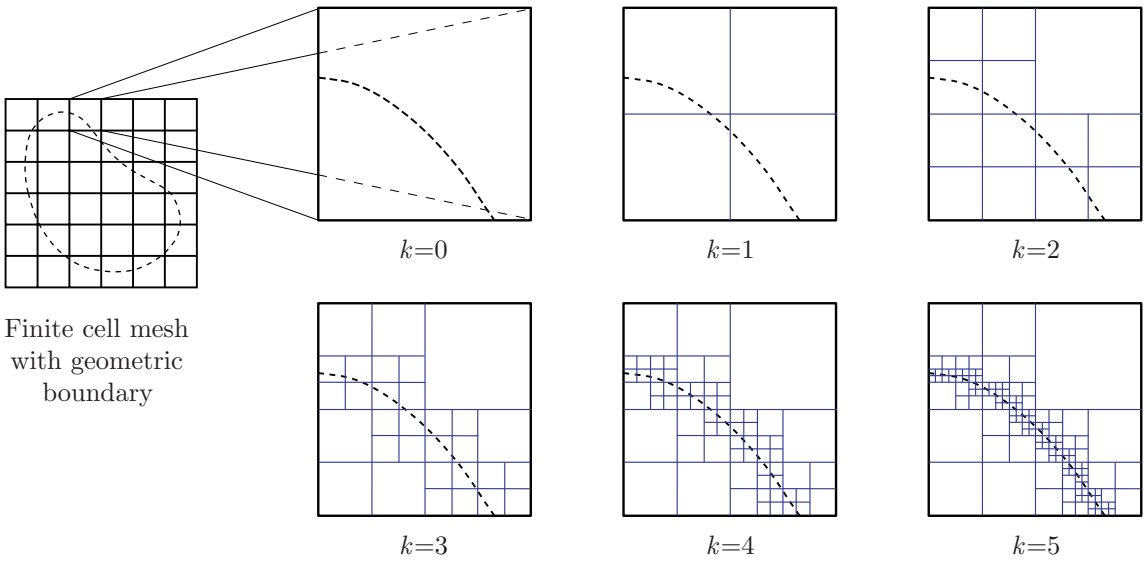


Figure 2.2: 2D sub-cell structure (thin blue lines) for adaptive integration of finite cells (bold black lines) that are cut by the geometric boundary (dashed line).

The sum of N_a denotes a finite set of n high-order basis functions, and $\boldsymbol{u}_a$ and $\delta\boldsymbol{u}_a$ the corresponding vectors of unknown coefficients [34, 108]. Following the standard Bubnov-Galerkin approach [17, 83, 108, 222], the substitution of Equations (2.4) and (2.5) into the weak form Equation (2.1) leads to a discrete finite cell representation

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \tag{2.6}$$

with stiffness matrix $\boldsymbol{K}$ and load vector $\boldsymbol{f}$. Due to the similarity to the standard finite element method [17, 83, 108, 197, 222], the implementation of the FCM can exploit existing finite element techniques to the full.

2.1.2 Adaptive integration

The accuracy of numerical integration by Gauss quadrature [108, 194], which assumes smoothness of the integrands, is considerably influenced by discontinuities within cells introduced by the penalization parameter α of Equation (2.3) [74, 150]. Therefore, the finite cell method uses composed Gauss quadrature to improve the integration accuracy in cells cut by geometric boundaries, based on a hierarchical decomposition of the original cell into integration sub-cells [74]. Figure 2.2 illustrates the generation of the sub-cell structure in two dimensions, where it can be built up in the sense of a quadtree [171, 172]. Starting from the original finite cell of level k=0, each sub-cell of level k=i is first checked whether it is cut by a geometric boundary. If true, it is replaced by four equally spaced cells of level k=i+1, each of which is equipped with $(p+1)\times(p+1)$ Gauss points. Partitioning is repeated for all cells of current level k, until a predefined maximum depth k=m is reached. The quadtree approach can be easily adjusted to one or three dimensions by binary trees or octrees, respectively [74, 172]. To clearly distinguish between finite cell and sub-cell meshes, finite cells are plotted in black and integration sub-cells are plotted in blue lines throughout this work (see Figure 2.2).

The adaptive integration scheme is easy to implement, keeps the regular grid structure of the FCM, and requires considerably less computational effort than non-adaptive schemes such

as the Gauss point method [150, 224]. However, the major part of the computational cost of the finite cell method still stems from linear algebra operations, which need to be repeated many times due to the large number of Gauss points in the sub-cell structure during the integration of the stiffness matrix. Note that a reduction of the $(p{+}1)$ Gauss points in each sub-cell direction is not recommended, since this introduces an additional integration error, which for instance dramatically decreases the convergence rate of the Newton-Raphson procedure or can provoke a random failure of the nonlinear deformation mapping in a geometrically nonlinear context.

2.1.3 Weak imposition of unfitted Dirichlet constraints

The fictitious domain concept inherently satisfies Neumann boundary conditions of zero traction, since stresses cannot be transferred beyond Ω_{phys} due to the stiffness penalization in Ω_{fict} [74, 150]. Non-zero Neumann boundary conditions can be simply imposed by integrating over the Neumann boundary Γ_N (see Figure 2.1), irrespective of whether the geometric boundary coincides with a cell boundary or not. For Dirichlet boundary conditions, the situation is more involved. In case of cuboidal domains, where boundaries coincide with cell boundaries of the structured grid, they can be imposed strongly by standard finite element techniques [56, 83, 108, 198]. In case of more complex domains, Dirichlet constraints are defined along boundaries of arbitrary geometry cutting through finite cells, which require an imposition in a weak sense by variational techniques such as the penalty method [11, 13, 84, 220], the Lagrange multiplier method [39, 84, 87, 91, 92, 94, 221] or Nitsche's method [19, 40, 66, 78, 80, 99, 100].

In the framework of the FCM, the latter is usually preferred [168, 218], since it does not introduce additional unknowns, preserves a symmetric, positive definite stiffness matrix and satisfies variational consistency in the sense that solutions of the weak form can be shown to be solutions of the original boundary value problem [78, 84]. In linear elasticity, Nitsche's method extends the weak form of Equation (2.1) by additional terms as follows

$$\begin{aligned} \delta W_{\boldsymbol{K}}\left(\boldsymbol{u},\delta\boldsymbol{u}\right) \;=\; & \int_{\Omega} \boldsymbol{\sigma} : \left(\nabla_{sym}\,\delta\boldsymbol{u}\right) dV \;+\; \beta \int_{\Gamma_D} \boldsymbol{u}\cdot\delta\boldsymbol{u}\; dA \\ & - \int_{\Gamma_D} \delta\left(\boldsymbol{\sigma}\cdot\boldsymbol{n}\right)\cdot\boldsymbol{u}\; dA \;-\; \int_{\Gamma_D} \left(\boldsymbol{\sigma}\cdot\boldsymbol{n}\right)\cdot\delta\boldsymbol{u}\; dA \end{aligned} \tag{2.7}$$

$$\begin{aligned} \delta W_{\boldsymbol{f}}\left(\boldsymbol{u},\delta\boldsymbol{u}\right) \;=\; & \int_{\Omega_{phys}} \delta\boldsymbol{u}\cdot\boldsymbol{b}\; dV \;+\; \int_{\Gamma_N} \delta\boldsymbol{u}\cdot\boldsymbol{t}\; dA \\ & + \beta \int_{\Gamma_D} \hat{\boldsymbol{u}}\cdot\delta\boldsymbol{u}\; dA \;-\; \int_{\Gamma_D} \delta\left(\boldsymbol{\sigma}\cdot\boldsymbol{n}\right)\cdot\hat{\boldsymbol{u}}\; dA \end{aligned} \tag{2.8}$$

where $\delta W_{\boldsymbol{K}}{=}\delta W_{\boldsymbol{f}}$. Function $\hat{\boldsymbol{u}}$ denotes the prescribed displacements along the Dirichlet boundary Γ_D, scalar β is a stabilization parameter, which can be chosen empirically or according to a generalized Eigenvalue problem [78, 84], and $\boldsymbol{n}$ is the outward unit normal vector on Γ_D. Evaluation of Equations (2.7) and (2.8) leads to the stiffness matrix $\boldsymbol{K}$ and the force vector $\boldsymbol{f}$, respectively.

2.1.4 High-order approximation bases

So far, the finite cell method has been closely linked to the high-order p-version basis derived from integrated Legendre polynomials [74, 150, 163], which is well-known from its application in the p-version of the finite element method [70, 71, 197, 198]. Convergence is achieved by increasing the polynomial degree p of the basis functions to reduce the approximation error, while the structured high-order mesh remains unchanged. For smooth problems, the finite cell method with p-version basis functions has been shown to achieve exponential rates of convergence in energy norm [74, 150, 163]. Further advantages of approximations based on the p-version basis comprise the locking-free behavior of high-order discretizations of thin-walled structures [71, 161] or incompressible materials [49, 196, 104] and an increased level of robustness against severe mesh distortion in geometrically nonlinear problems [72, 104].

Similar benefits are provided by alternative high-order approximation methods, such as spectral methods [43, 68] or B-spline based finite elements [56, 106, 109], and can be expected to apply also for corresponding finite cell approaches, which use their respective high-order basis functions in the framework of the FCM concept. In the following, this is demonstrated for the example of high-order B-splines.

2.2 High-order *p*-version and B-spline bases

The present work unhinges the finite cell method from a special high-order basis and demonstrates for Ansatz spaces based on high-order B-splines that in the framework of the FCM, a solution behavior comparable to that obtained with the established p-version basis can be achieved. The two resulting FCM schemes will be coined the p- and B-spline versions of the FCM [175, 176, 178, 180] and are briefly introduced in the following.

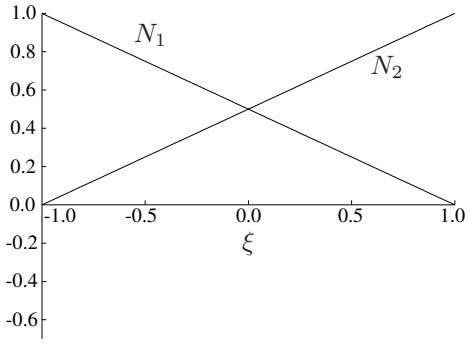
2.2.1 The *p*-version of the finite cell method

The high-order basis originally applied in the finite cell method [74, 150] uses a regular mesh of elements of the p-version of the finite element method, introduced by Szabó and Babuška [70, 71, 197, 198]. Its formulation is based on integrated Legendre polynomials of the form

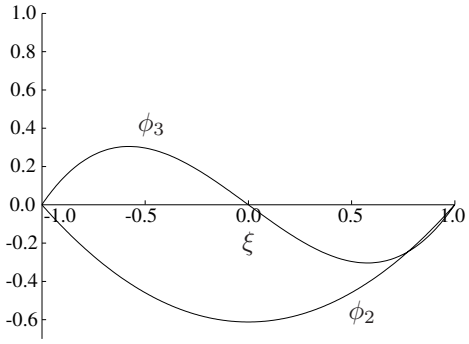
$$\begin{aligned} \phi_j(\xi) &= \sqrt{\frac{2j-1}{2}} \int_{-1}^{\xi} P_{j-1}(t)\, dt \\ &= \frac{1}{\sqrt{4j-2}} \left(P_j(\xi) - P_{j-2}(\xi) \right), \quad j = 2, 3, ... \end{aligned} \tag{2.9}$$

where $P_j(\xi)$ are standard Legendre polynomials in local cell coordinates [194, 197]. As illustrated in Figure 2.3, the one-dimensional basis is constructed by combining the hat functions of the standard linear basis (see for example [17, 83, 108, 197, 198, 221]) with higher-order functions provided by Equation (2.1). The resulting basis is hierarchic, so that an increase of the polynomial degree p of the basis by 1 is achieved by the addition of another ϕ_j [70, 197].

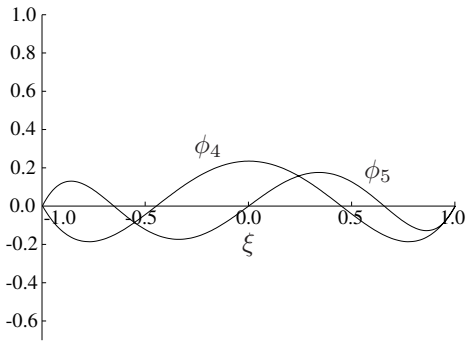
Corresponding higher-dimensional bases can be constructed by tensor products of the 1D case. Figure 2.4 illustrates basis functions for high-order quadrilateral elements in 2D, which are usually grouped as follows:



Linear nodal modes N_j, j=1, 2

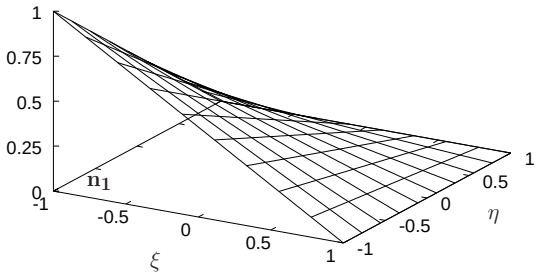


Quadratic and cubic integrated Legendre basis functions ϕ_j, j=2, 3

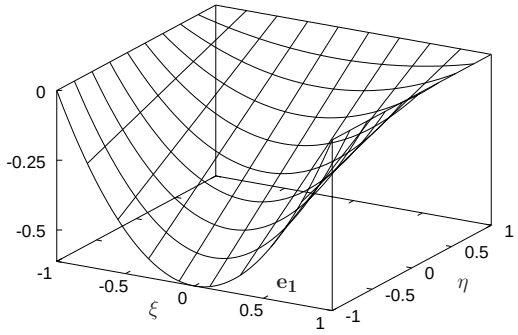


Quartic and quintic integrated Legendre basis functions ϕ_j, j=4, 5

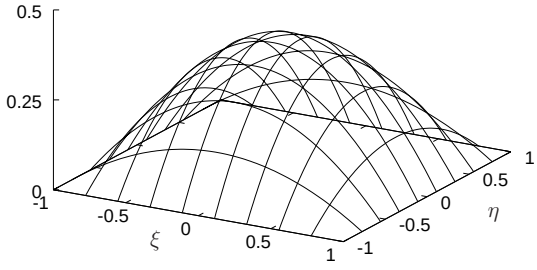
Figure 2.3: The 1D p-version basis in the parameter space ξ. The combination of the displayed functions yields a 1D finite cell of p=5.



Nodal mode $N_1^{\mathrm{n}_1}$ at node n_1



Quadratic edge mode $N_2^{\mathrm{e}_1}$ at edge e_1



Quadratic internal mode $N_{2,2}^{\mathrm{int}}$

Figure 2.4: Examples of p-version basis functions for 2D quadrilateral cells in the parameter space $\{\xi, \eta\}$ (adapted from [70]).

- *Nodal modes* are the standard bilinear basis functions. The corresponding mode for node n_1 of Figure 2.4 reads

$$N_1^{\mathrm{n}_1}(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta) \tag{2.10}$$

- *Edge modes* are defined separately for each individual edge and vanish at all other edges.

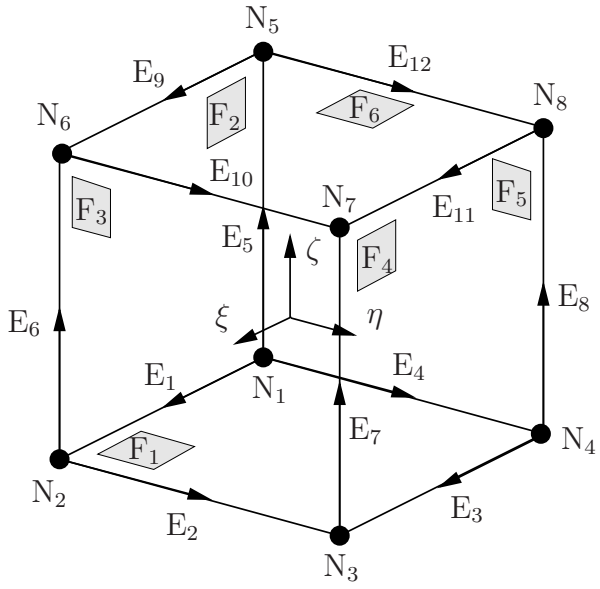


Figure 2.5: Definition of nodes, edges and faces for a 3D hexahedral cell (adapted from [70]).

The corresponding modes for edge e_1 of Figure 2.4 read

$$N_i^{e_1}(\xi,\eta) = \frac{1}{2}(1-\eta)\phi_i(\xi) \tag{2.11}$$

- *Internal modes* are purely local and vanish at all element edges. They read

$$N_{i,j}^{\text{int}}(\xi,\eta) = \phi_i(\xi)\phi_j(\eta) \tag{2.12}$$

For a high-order hexahedral cell in three dimensions as shown in Figure 2.5, one can generalize the two-dimensional concept and obtain three-dimensional basis functions, classified in the following four groups:

- *Nodal modes* are the standard trilinear basis functions, which read

$$N_{1,1,1}^{N_i}(\xi,\eta,\zeta) = \frac{1}{8}(1+\xi_i\,\xi)(1+\eta_i\,\eta)(1+\zeta_i\zeta), \quad i=1,...,8 \tag{2.13}$$

  They are well-known from the isoparametric eight-node brick element. (ξ_i,η_i,ζ_i) are the local coordinates of the i-th node.

- *Edge modes* are defined separately for each individual edge. If we consider for example edge E_1 (see Figure 2.5), the corresponding edge modes read

$$N_{i,1,1}^{E_1}(\xi,\eta,\zeta) = \frac{1}{4}(1-\eta)(1-\zeta)\phi_i(\xi) \tag{2.14}$$

- *Face modes* are defined separately for each individual face. If we consider for example face F_1, the corresponding face modes read

$$N_{i,j,1}^{F_1}(\xi,\eta,\zeta) = \frac{1}{2}(1-\zeta)\phi_i(\xi)\phi_j(\eta) \tag{2.15}$$

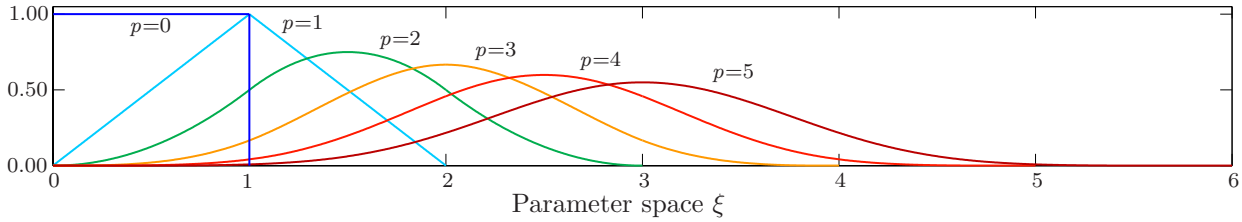


Figure 2.6: B-spline basis functions of increasing polynomial degree p, starting from a piecewise constant at p=0 that are defined over knot spans in the parameter space.

- *Internal modes* are purely local and vanish at the faces of the hexahedral cell. They read

$$N^{\text{int}}_{i,j,k}(\xi,\eta,\zeta) \quad = \quad \phi_i(\xi)\phi_j(\eta)\phi_k(\zeta) \tag{2.16}$$

The indices i, j, k of the basis functions denote the polynomial degrees in local directions ξ, η, ζ. To limit the number of additional unknowns in 2D and 3D, the so-called *trunk space* is used instead of the full tensor product basis [70, 198].

The geometry of the structured finite cell mesh in terms of the position vector $\boldsymbol{X}$ can be represented exactly by the nodal part alone

$$\boldsymbol{X} = \sum_{i=1}^{n_{\text{vert}}} N^{\text{n}}_i \; \boldsymbol{X}_i \tag{2.17}$$

where $\boldsymbol{X}_i$ and N^{n}_i denote the location of the n_{vert} cell vertices and the corresponding nodal modes, respectively. The p-version of the FCM can thus be regarded as a sub-parametric finite element scheme [17, 156]. The resulting FCM stiffness matrix Equation (2.6) inherits all beneficial properties of the p-version FEM, such as hierarchy of the modal contributions and a tremendous improvement of the condition number under p-refinement with respect to standard nodal finite element schemes [70, 197, 198].

2.2.2 The B-spline version of the finite cell method

The B-spline version of the FCM has been recently established as a suitable alternative to the p-version [175, 176, 178, 180]. Its formulation is based on high-order B-spline basis functions $N_{i,p}$ of polynomial degree p. We mention only the most important concepts for the construction of B-spline basis functions here and refer to Section 6.1 for details, where a more comprehensive review of B-splines and corresponding geometric aspects is provided in the framework of isogeometric analysis.

B-spline basis functions are defined by p+2 knots $\xi_1 \leq \xi_2 \leq ... \leq \xi_{p+2}$ in the parameter space ξ [56, 109, 154, 165]. The resulting p+1 knot spans contain piecewise polynomials of degree p, which join smoothly up to a continuous differentiability of C^{p-1}. A number of n basis functions constitute a patch, defined by a so-called knot vector [56, 154, 165]

$$\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}, \quad \xi_1 \leq \xi_2 \leq ... \leq \xi_{n+p+1} \tag{2.18}$$

The knots of each individual basis function $N_{i,p}$ with patch index i can be identified as the consecutive entries $\{i, i{+}1, ..., i{+}p{+}1\}$ in Ξ. Figure 2.6 illustrates B-spline basis functions $N_{i,p}$ of arbitrary polynomial degree p, starting from piecewise constants $N_{i,0}$ [154, 165].

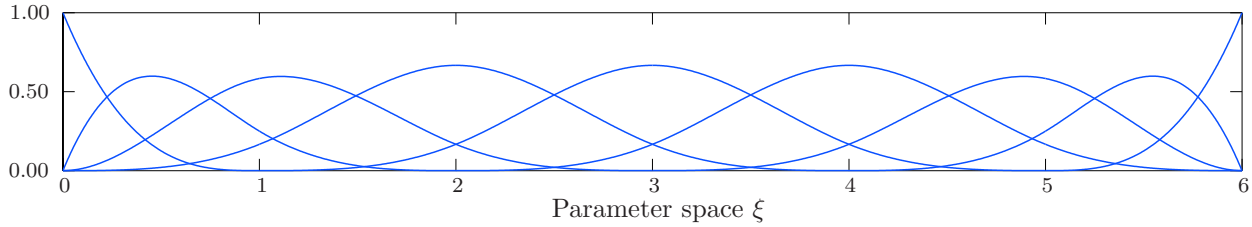


Figure 2.7: Example of an open cubic B-spline patch consisting of three uniform B-splines and six knot spans. At the boundaries, knots are repeated four times to obtain interpolatory B-splines.

A uniform B-spline patch with open knot vectors as illustrated in Figure 2.7 guarantees optimal approximation for smooth problems due to maximum continuity within the patch, but also allows for the imposition of boundary conditions by standard FE techniques [56, 109]. Its basis functions away from the boundary of the patch consist of uniform B-splines constructed from equidistant knots, which can be interpreted as translated copies of each other [106]. At the boundaries, knots are repeated $p+1$ times in order to make the basis interpolatory.

The two-dimensional B-spline basis uses two independent 1D open knot vectors with indices $\{i, j\}$ for local coordinates $\{\xi, \eta\}$, respectively. The parameter space is then discretized by a structured grid of knot spans, whose knot positions can be obtained by permuting all entries $\{i, j\}$. Corresponding multivariate B-spline basis functions are obtained by taking the tensor product of its univariate components

$$N_{i,j,p}(\xi, \eta) = N_{i,p}(\xi) \cdot N_{j,p}(\eta) \tag{2.19}$$

The three-dimensional B-spline basis can be constructed analogously, using three independent 1D knot vectors with indices $\{i, j, k\}$ for local coordinates $\{\xi, \eta, \zeta\}$, respectively, and taking the corresponding tensor product

$$N_{i,j,k,p}(\xi, \eta, \zeta) = N_{i,p}(\xi) \cdot N_{j,p}(\eta) \cdot N_{k,p}(\zeta) \tag{2.20}$$

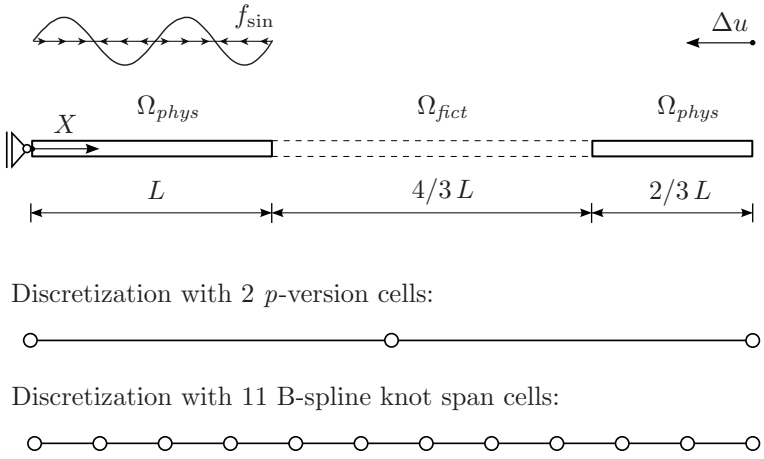
In the framework of the finite cell method, the discretization is accomplished with a regular grid of knot spans of width h in the sense of the fictitious domain concept as illustrated in Figure 2.1. In two and three dimensions, each knot span can be identified as a quadrilateral or hexahedral finite cell, respectively, with full Gaussian integration [176, 178, 180]. The physical coordinates $\boldsymbol{X}$ of the finite cell grid can be generated from a simple linear transformation of the parameter space $\boldsymbol{\xi}$

$$\boldsymbol{X} = \boldsymbol{X}_0 + h\,\boldsymbol{\xi} \tag{2.21}$$

where $\boldsymbol{X}_0$ denotes the origin of the physical coordinate system in the parameter space. The B-spline version of the FCM is thus also a sub-parametric scheme [17, 156].

2.3 Characteristic solution behavior

In the finite cell method, the polynomial degree p of the basis functions is increased to reduce the approximation error, while the structured high-order mesh remains unchanged. For the illustration of the typical solution behavior, a linear elastic uni-axial rod is examined, for



Parameters:

Young's modulus E=*1.0*
Poisson's ratio $\nu = 0.0$
Penalization with $\alpha = 10^{-q}$
Area A=*1.0*
Length L=*1.0*
Displacement load Δu=0.02
Sine load $f_{\sin} = 1/20 \sin(4\pi X)$

Figure 2.8: Uni-axial rod example. Geometric boundaries are located at $X = 1.0$ and $X = 2\frac{1}{3}$.

which geometry, material and boundary conditions are specified in Figure 2.8. Its middle part represents the fictitious domain Ω_{fict}, whose Young's modulus E is penalized with parameter α=10^{-8}. The example approximates the situation of two separate rods. The right one undergoes a rigid body movement Δu and the left one is subjected to a sine load $f_{\sin}$. The FCM discretizations considered consist of 2 p-version finite cells and 11 knot span cells as shown in Figure 2.8. Due to the different construction of the bases, the B-spline version requires a denser knot span grid than the p-version in order to achieve a comparable amount of degrees of freedom (dofs). For all computations of this section, adaptive sub-cells of depth k=20 are used to minimize the integration error in cells cut by geometric boundaries (see Section 2.2).

2.3.1 Smooth extension of solution fields

The corresponding analytical solution fields with finite α=10^{-8} exhibit kinks and jumps in displacements and strains, respectively, at geometric boundaries X=L and X=7/3. The analytical solution is approximated by an overkill solution with 1,800 equally-spaced standard cubic finite elements, where the geometric boundaries coincide with nodes, so that discontinuities can be represented. The p- and B-spline versions of the FCM, however, produce solution fields, which extend smoothly into the fictitious domain despite the discontinuities of the analytical solution. This is illustrated in Figure 2.9, which compares the analytical strains to the numerical strains of the p- and B-spline versions. The importance of the smooth extension of the FCM solution into the fictitious domain for the overall convergence behavior of the finite cell method can be explained with the help of the penalty parameter α in conjunction with the total strain energy

$$U = \int_\Omega \Psi \, dV = \frac{1}{2} \int_\Omega \boldsymbol{\sigma} : \boldsymbol{\varepsilon} \, dV \tag{2.22}$$

where Ψ represents the strain energy function, defined over the complete domain Ω. The best approximation property to the total strain energy U states that the solution of a Galerkin finite element scheme represents a least-squares best fit to the exact solution in terms of Equation (2.22) [108, 221]. Due to the penalization with parameter α of Equation (2.3), deviations from

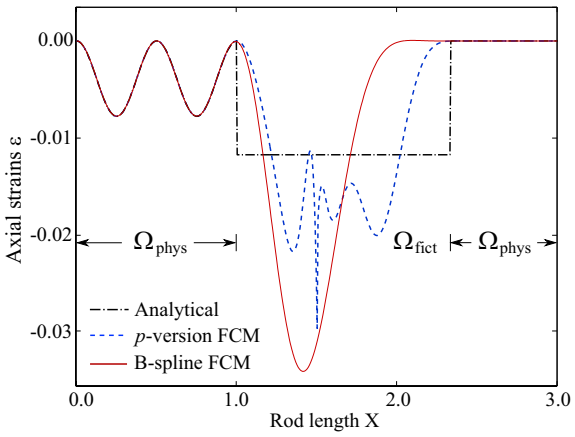


Figure 2.9: Smooth extension of the FCM solutions (p=15) vs. discontinuous analytical solution (α=10^{-8}) for the linear elastic strains of the 1D example.

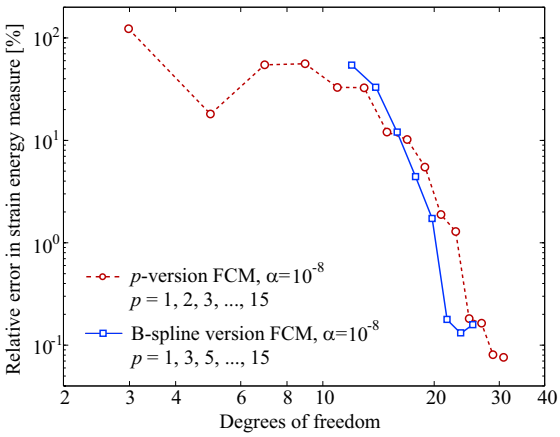


Figure 2.10: Convergence in energy norm obtained by p-refinement in the given p-version and B-spline discretizations of the linear elastic 1D example.

the exact solution in Ω_{fict} have a considerably smaller impact on the strain energy Equation (2.22) than deviations in Ω_{phys}. Therefore, a minimization of the strain energy error by the high-order basis of the FCM scheme results in an accurate approximation in Ω_{phys}, where potential deviations lead to considerable error contributions. In Ω_{fict}, a largely inaccurate approximation is allowed, since potential deviations lead to negligible error contributions due to penalization. In particular, this implies a smooth extension of the solution into the fictitious domain, so that its gradients in Ω_{phys} remain unaffected up to the geometric boundary, as illustrated in Figure 2.9 for the 1D example.

2.3.2 High rates of convergence in energy norm

We will assess accuracy and convergence of the FCM versions by the relative error in terms of the total strain energy U of Equation (2.22) [70, 108, 197, 198, 221], which is obtained by the following expression

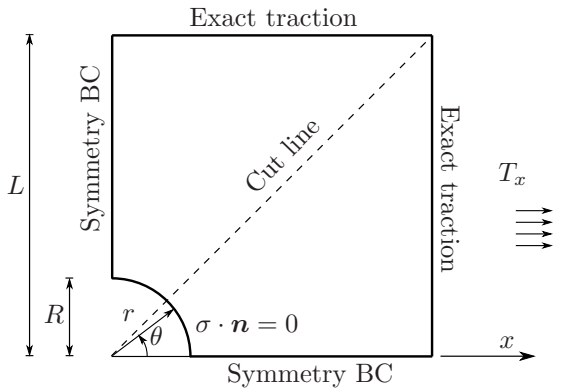
$$e_r = \sqrt{\frac{|U_{ex} - U_{FCM}|}{U_{ex}}} \times 100\% \tag{2.23}$$

U_{ex} represents the exact strain energy of the original problem defined over the physical domain Ω_{phys}. Accordingly, U_{FCM} denotes the strain energy contribution from Ω_{phys}, obtained numerically with an FCM scheme. In the linear elastic case, Equation (2.23) is equivalent to the relative error in energy norm [70, 197, 221].

The corresponding rate of convergence q, with which e_r decreases from the i^{th} to the $(i+1)^{th}$ p-refinement step compared to the number of degrees of freedom n_{dof} is

$$q = -\frac{\log_{10}\left(e_r^{i+1}/e_r^{i}\right)}{\log_{10}\left(n_{dof}^{i+1}/n_{dof}^{i}\right)} \tag{2.24}$$

The physical reference strain energy U_{ex} for the example of Figure 2.8 consists of the energy contribution by $f_{\sin}$ only. An overkill discretization with 1,000 cubic finite elements



Parameters:
Geometry: L=4.0; R=1.0
Material: E=10^5; $\nu = 0.3$
Far-field traction: $T_x = 10.0$
Penalization parameter $\alpha = 10^{-12}$

Analytical solution:

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

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

$$\tau_{r\theta}(r,\theta) = -\frac{T_x}{2}\left(1 - \frac{3R^4}{r^4} + \frac{2R^2}{r^2}\right)\sin(2\theta)$$

Figure 2.11: Linear elastic plate with a circular hole: problem definition and exact solution.

taking into account the left rod only yields U_{ex}=$1.1873576208 \cdot 10^{-5}$. Figure 2.10 compares the convergence behavior of the p- and B-spline versions of the FCM, if the polynomial degree of the discretizations given in Figure 2.8 is increased from p=1 to 15. Both the p-version and the B-spline version of the FCM converge exponentially with a maximum rate of *q=23.56* and *q=23.74*, respectively. The penalization parameter α=10^{-8} cannot completely erase the influence of the fictitious domain, and the corresponding error takes control at a value around 0.1% in both the p- and B-spline versions, leading to a flattening of the convergence curves. The present example shows that the p- and B-spline bases applied in the framework of the finite cell concept exhibit an equivalent solution behavior and achieve comparable performance in terms of error level, rates of convergence and flattening of the convergence curve, although their high-order approximation bases are very different.

2.4 Numerical examples at small strains

Having observed the equivalent general solution behavior of the p- and B-spline versions for the simple one-dimensional example, we would like to corroborate that statement by considering more complex cases of linear elasticity in two and three dimensions. The standard example of a two-dimensional plate with a circular hole demonstrates the FCM discretization procedure and the convergence behavior with p-version and B-spline basis functions. The 2D ring problem and the 3D proximal femur shown in the following outline the imposition of unfitted Dirichlet boundary conditions and the versatility of the FCM for the analysis of very complex structures.

2.4.1 Plate with a circular hole

Figure 2.11 provides geometry, boundary conditions and the analytical solution [170] for the classical example of an infinite plate with a circular hole under in-plane tension [56, 109]. Due to symmetry, only one quarter of the problem is considered. Figure 2.12 illustrates the discretization procedure in the framework of the finite cell concept introduced in Section 2.1. Basis functions are defined over an axis-aligned grid, which embeds the physical domain Ω_{phys} in a simple square irrespective of its geometry. The circular hole constitutes the fictitious domain Ω_{fict}, whose influence is extenuated by penalizing the elasticity matrix at integration point level with a finite value α=10^{-12} according to Equations (2.2) and (2.3). The integration

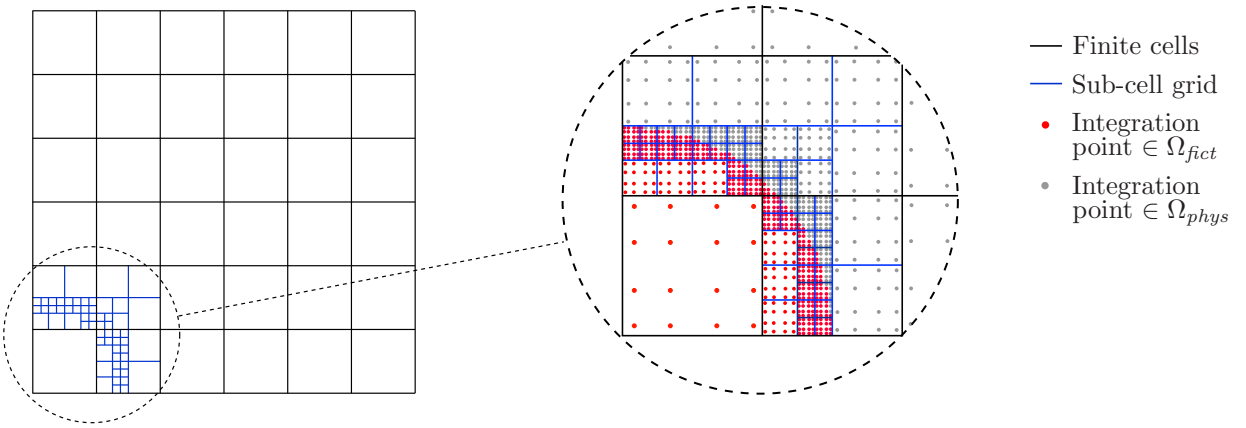


Figure 2.12: Basis functions are defined over an axis-aligned grid (black lines). A sub-cell grid (blue lines) leads to an aggregation of Gauss points around the boundary, where the elasticity matrix is penalized according to Equation (2.3) at all points outside the physical domain Ω_{phys} (marked in red).

accuracy in cut elements is ensured by adaptive integration sub-cells, which lead to an aggregation of Gauss points around the boundary of the circular hole. It is important to note that integration sub-cells do not affect basis functions, but only increase the number of integration points around the geometric boundary. The foundation of the FCM discretization procedure is a point location query, which is able to determine for each integration point with global coordinates $\{x, y\}$, if it is located in Ω_{phys} or Ω_{fict}. In the present example, the geometric boundary of the circular hole can be used to set up an inequality of the form

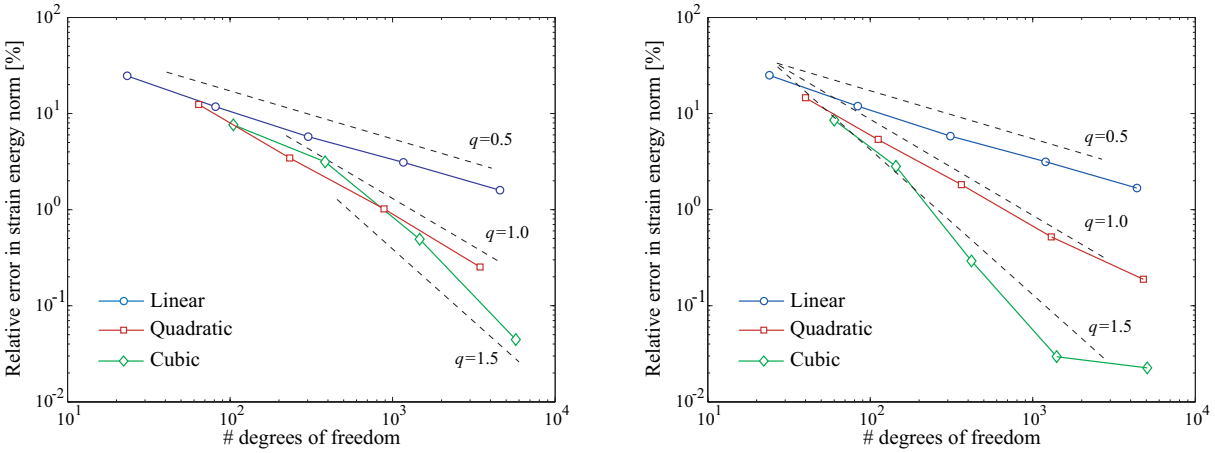
$$x^2 + y^2 \leq R^2 \tag{2.25}$$

with R being the radius of the hole. A simple evaluation of Equation (2.25) is sufficient to decide whether the point is located in Ω_{phys} (inequality is true) or Ω_{fict} (inequality is false). For more complex arbitrarily shaped geometries, more general point location queries based on voxel models or space-partitioning data structures are addressed later on in Sections 4.4.1 and 7.4.1. Finally, it should be noted that Dirichlet boundary conditions can be incorporated strongly via standard finite element techniques, since the boundary of cell edges coincide with the geometric boundary of the physical domain Ω_{phys}.

2.4.1.1 FCM based on *h*-refinement

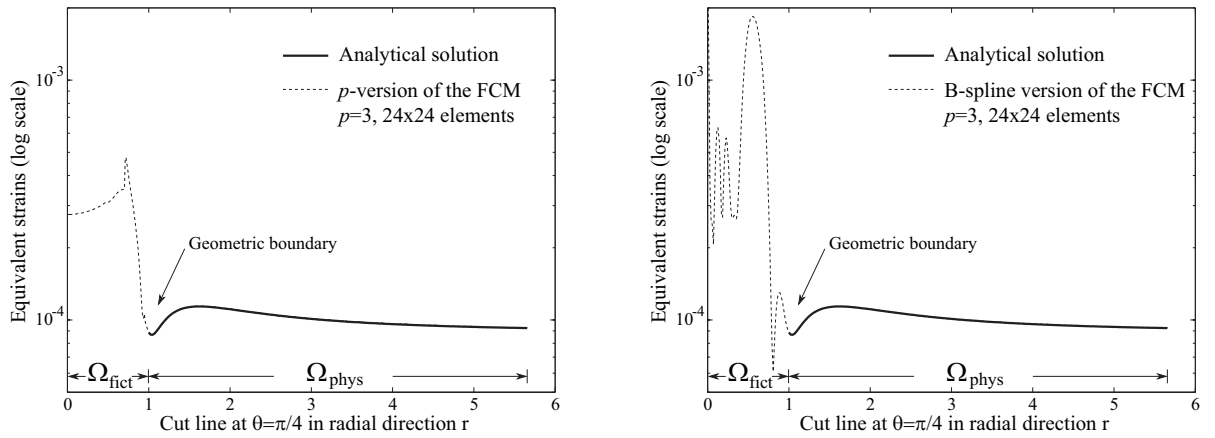
In a first step, the finite cell method is tested in the framework of classical h-refinement [108, 197, 221]. We therefore discretize the plate with basis functions of moderate polynomial degree p and quadrisect the finite cell mesh in each refinement step to reduce the approximation error. Each discretization is supplemented by adaptive sub-cells for integration of cells cut by the geometric boundary. We apply a depth of the sub-cell quadtree of m=4, placing an overly large number of 20 integration points along each local sub-cell direction, in order to exclude any effect from a potential integration error in the following convergence study.

Figure 2.13 shows the resulting convergence for both the p-version and B-spline version in terms of the relative error in energy norm Equation (2.23). Both FCM schemes operate on the same series of quadrisected grids, starting from a discretization with 3×3 finite cells. Depending on the polynomial degree p and the length of a cell edge h, the mathematically



(a) p-version. **(b)** B-spline version.

Figure 2.13: Convergence in energy norm for uniform h-refinement of the finite cell mesh.

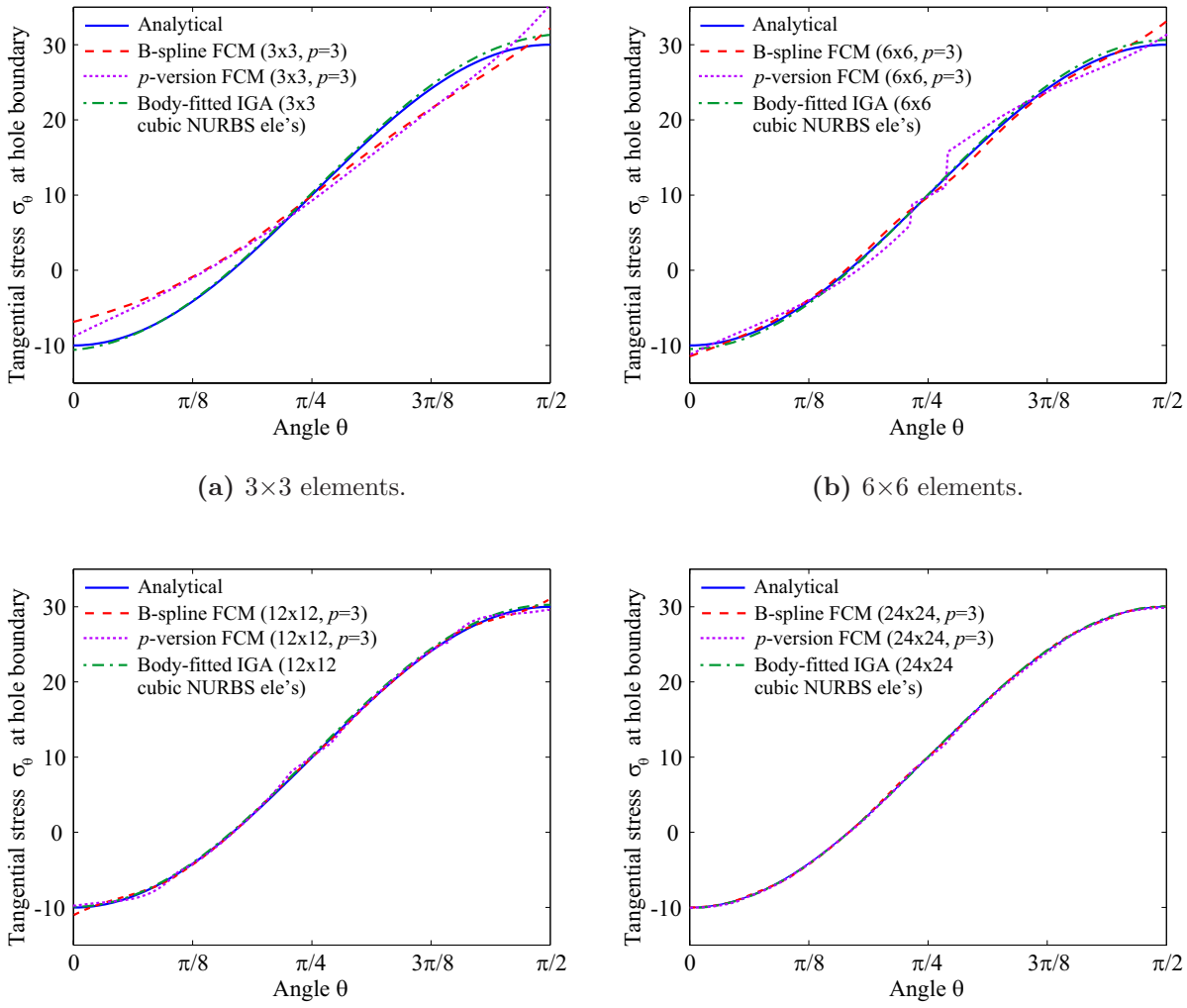


(a) p-version. **(b)** B-spline version.

Figure 2.14: Smooth extension of solution fields beyond the physical domain, illustrated for von Mises equivalent strains plotted along the diagonal cutline (see Figure 2.11).

optimal rate of convergence is of order $O(h^p)$, and of order $O(n_{dof})^{-p/2}$ with respect to the number of degrees of freedom n_{dof}, which is approximately inversely proportional to h^2 for a two-dimensional problem [108, 221]. From Figure 2.13, we can observe that for linear, quadratic and cubic basis functions, optimal rates of convergence are achieved with both the p-version and the B-spline basis. In addition, we do not erase basis functions from the discretization, which are lying outside the physical domain, although they obviously do not contribute to the accuracy of the finite cell method. Thus, their removal could even slightly improve the reported constants. The removal of basis functions with no support in Ω_{phys} is discussed in more detail in Sections 7.3, where it is applied in the framework of adaptive analysis of complex CAD models with the B-spline version of the FCM.

As discussed in Section 2.3.1, the accuracy of the finite cell method relies on the smooth extension of the solution fields into the fictitious domain. This important behavior of the



(a) 3×3 elements. **(b)** 6×6 elements.

(c) 12×12 elements. **(d)** 24×24 elements.

Figure 2.15: Convergence of circumferential stresses $\sigma_\theta(r,\theta)$ at the circular boundary of the hole (r=1.0) for the finite cell method and standard body-fitted isogeometric analysis [56] under uniform h-refinement of corresponding 3×3 cubic meshes.

FCM is illustrated in Figure 2.14 by von Mises equivalent strains plotted along the diagonal cut line. While the exact solution is only defined over the physical domain Ω_{phys}, the finite cell solution extends smoothly into the fictitious domain Ω_{fict}, although the elasticity matrix is discontinuous due to the penalization factor α as defined by Equations (2.2) and (2.3). Due to the best approximation property of a Galerkin finite element scheme [108, 221], largely inaccurate solutions in the fictitious domain Ω_{fict} are allowed, which permit an accurate solution in the physical domain Ω_{phys} up to the geometric boundary. Of particular interest in this respect is the stress accuracy that can be achieved directly on the geometric boundary in cut cells. We therefore plot the circumferential stresses on the circular boundary of the hole obtained with different cubic finite cell meshes in Figure 2.15a through 2.15d and compare them to corresponding results obtained analytically (see Figure 2.11) and with body-fitted isogeometric discretizations given on page 118 of [56]. The plots illustrate that the FCM stresses converge

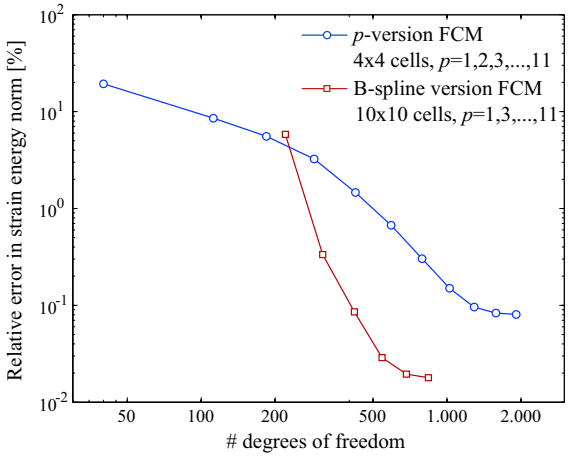


Figure 2.16: Convergence in energy norm for uniform p-refinement of the finite cell mesh.

to the analytical reference. However, standard body-fitted meshes lead to stresses that hit the reference faster.

2.4.1.2 FCM based on p-refinement

In contrast to standard nodal functions based on Lagrange polynomials [17, 108, 197, 221], integrated Legendre and B-spline basis functions still produce well conditioned system matrices, when the polynomial degree p of the basis functions is large [56, 70, 109, 197]. Therefore, convergence of the p- and B-spline versions of the finite cell method can alternatively be achieved by increasing the polynomial degree p of the basis functions to reduce the approximation error, while the structured high-order mesh remains unchanged. In fact, this is the standard procedure for global refinement in the finite cell method (see [150, 74] and Section 2.1.4), which we will justify in the following by considering the convergence behavior of the present plate example, discretized with 4×4 p-version cells and 10×10 knot span cells. Note that the difference in mesh size results from the fact that we want to consider discretizations with a comparable amount of degrees of freedom. We carry out all computations with a large sub-cell depth of m=6, placing an overly large number of 50 integration points in each local sub-cell direction, in order to make sure that the integration error does not play any role in the following convergence study.

Figure 2.16 illustrates the convergence of the relative error in energy norm Equation (2.23) under p-refinement of the considered FCM discretizations We observe that for the p-version of the FCM, the rate of convergence measured by Equation (2.24) improves exponentially from an initial q=0.79 to a maximum of q=2.74. For the B-spline version of the FCM, rates of convergence turn out to be high from the beginning with q=6.65 and q=3.13 in the first and fourth refinement step, respectively. The high rates of convergence obtained with global p-refinement leads to a specifically low ratio between the relative error and the corresponding number of degrees of freedom (dofs). Let us consider a target error in energy norm of 0.1 % that is to be obtained with the p-version of the finite cell method. With linear basis functions, extrapolation of the corresponding curve in Figure 2.13a indicates that we require approx. 100,000 dofs. With quadratic and cubic basis functions, we are able to decrease the number to approx. 10,000 and 3,500 dofs, respectively. With p-refinement, however, we require only

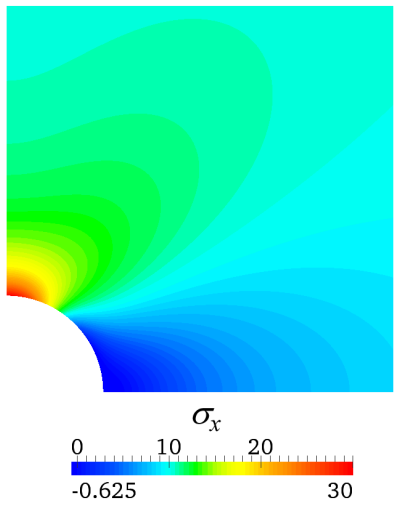
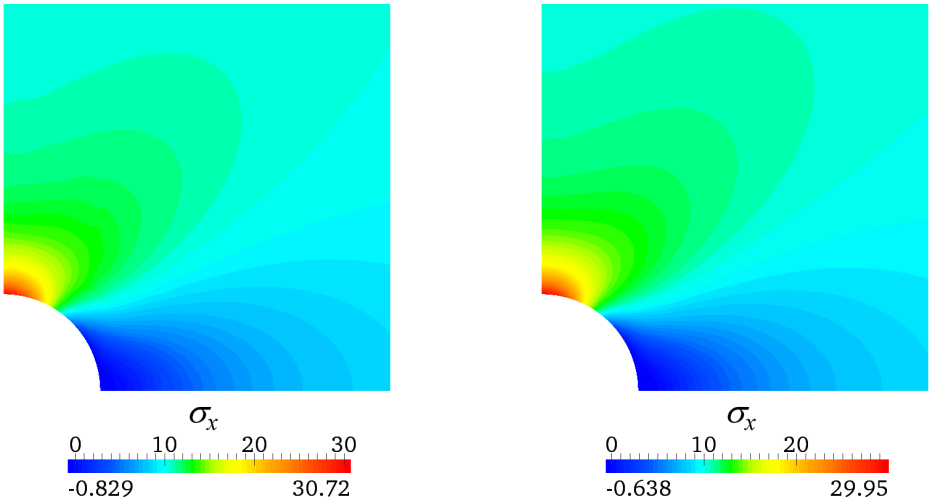


Figure 2.17: Exact normal stresses σ_x.



(a) 4×4 p-version cells. **(b)** 10×10 knot span cells.

Figure 2.18: Normal stresses σ_x, plotted over Ω_{phys}, for FCM discretization of polynomial degree p=7.

about 1,000 dofs. For the B-spline version of the finite cell method, a similar relation can be observed. While basis functions with p=1,2 and 3 require 100,000 dofs, 10,000 dofs and 800 dofs, respectively, p-refinement falls below the target error with only about 400 dofs. Similar to the 1D example discussed in Section 2.3.2, both curves of Figure 2.16 level off at some finite error, where the influence of the fictitious domain extension takes control due to the finite penalization parameter α.

Comparing the two FCM versions in Figure 2.16, we observe that the lowest error of the B-spline version is about one order of magnitude smaller than the lowest error in the p-version and obtained with about half the number of degrees of freedom. This is also reflected in Figures 2.17 and 2.18, which compare the analytical stress plot for σ_x with corresponding stress plots obtained with the p- and B-spline versions of the FCM at polynomial degree p=7. Whereas both FCM versions accurately reproduce the general stress distribution, the B-spline version comes closer to the reference in terms of maximum and minimum values. These observations

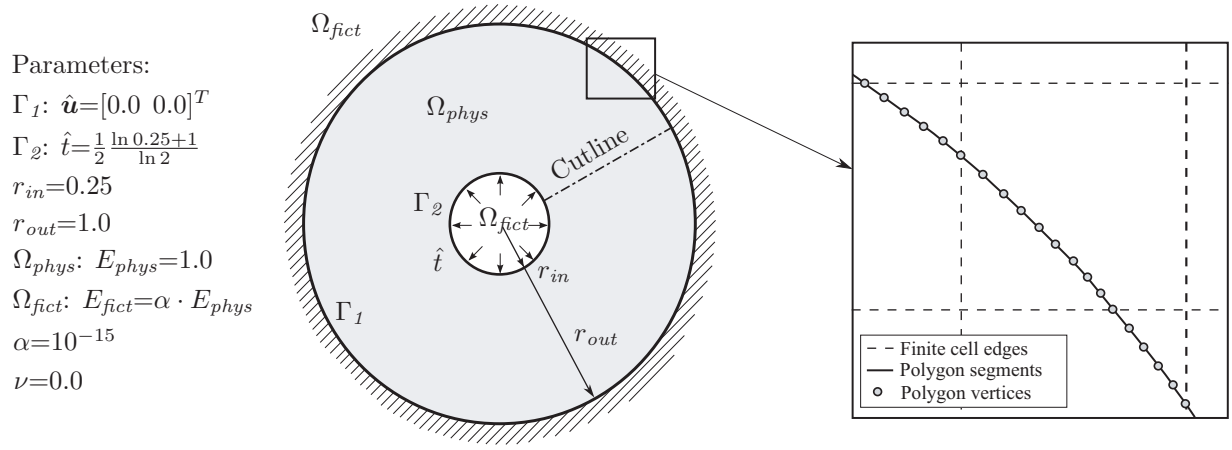


Figure 2.19: Ring under internal radial pressure and fixed outer displacements.

indicate an advantage of the B-spline version over the p-version in terms of the ratio between a relative error and the corresponding number of degrees of freedom, which will be corroborated more or less throughout the following examples and can be largely attributed to the high-order continuity of the B-splines. However, a more sophisticated comparison will be presented in Section 4.5, taking into account additional aspects such as computational time and required hardware resources, and will lead to a more balanced picture, highlighting specific strengths for each FCM version.

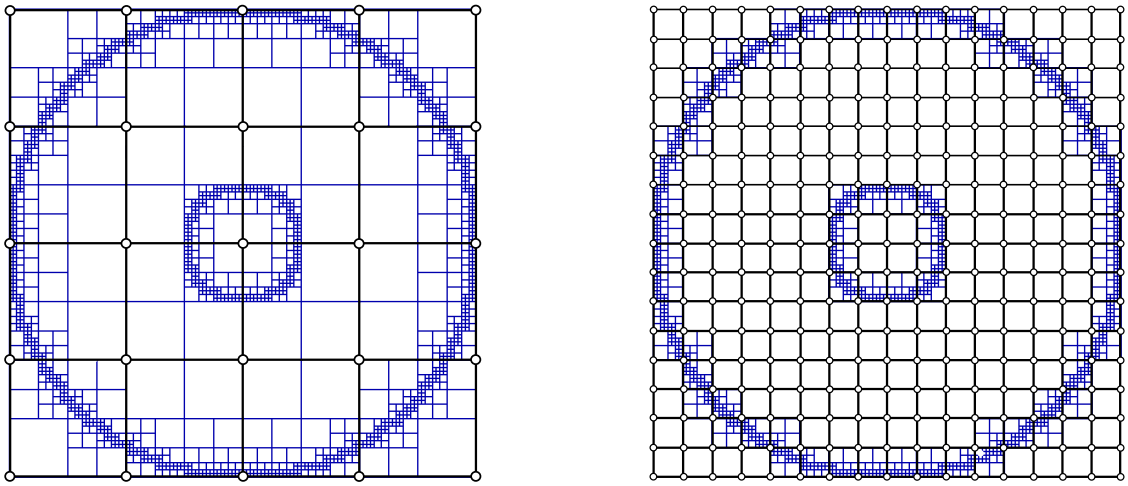
The results for the example of a plate with a spherical hole demonstrate that the finite cell method works very well with both integrated Legendre and B-spline basis functions. Both p- and B-spline versions of the FCM lead to equivalent solution characteristics. In particular, we highlighted optimal rates of convergence under uniform h-refinement and the high quality of stresses on the geometric boundaries in cut cells. Moreover, we saw that p-refinement achieves a given accuracy level with fewer degrees of freedom than the classical h-refinement strategy with low-order polynomials.

2.4.2 Ring with unfitted Dirichlet constraints

The imposition of unfitted Dirichlet boundary conditions with Nitsche's method in the p- and B-spline versions of the FCM is illustrated by the example of a ring in plane strain as shown in Figure 2.19. For the present example, curved boundaries are approximated with arbitrary precision by a polygon

$$\boldsymbol{s}: \begin{bmatrix} X^s \\ Y^s \end{bmatrix} = \sum_{i=1}^{n_{vert}} N_i^{lin}(\vartheta) \cdot \begin{bmatrix} X_i^s \\ Y_i^s \end{bmatrix} \tag{2.26}$$

X_i^s and Y_i^s denote the physical coordinates of the n_{vert} vertices of the polygon in the reference configuration, which lie on Γ. Nodal basis functions $N_i^{lin}(\vartheta)$ defined over parameter space $\vartheta \in [-1, 1]$ linearly interpolate Γ between vertices. Assuming that wherever polygon $\boldsymbol{s}$ crosses a cell edge, a vertex is placed (see Figure 2.19), the surface integrals in Equations (2.7) to (2.8) can be evaluated by placing $p+1$ Gauss points in the parameter space ϑ of each polygon segment without introducing an additional error from the discontinuities of the basis functions at cell edges. This parameterization concept can be analogically applied to three-dimensional



(a) 4×4 p-version finite cells with integration sub-cells of depth m=5.

(b) 16×16 knot span finite cells with integration sub-cells of depth m=3.

Figure 2.20: Finite cell discretizations with adaptive sub-cells for the ring example. Open circles denote element nodes and B-spline knots in the p-version and B-spline meshes, respectively, while the lines in blue denote integration sub-cells.

problems by introducing for example a triangular approximation for boundary surfaces of arbitrary geometry [74, 168].

The present example specifies radial and circumferential displacements along the Dirichlet boundary Γ_1

$$\hat{u}_r = 0.0;\ \hat{u}_\theta = 0.0 \tag{2.27}$$

and radial traction along the Neumann boundary Γ_2

$$\hat{t}_r = \frac{1}{2}\ \frac{\ln 0.25 + 1}{\ln 2} \tag{2.28}$$

In addition, radial body forces are applied

$$b_r = \frac{1}{r\ \ln 2} \tag{2.29}$$

over the physical domain Ω_{phys}. Assuming the compatible displacement solution in polar coordinates (r, θ) with origin in the center of the circular ring

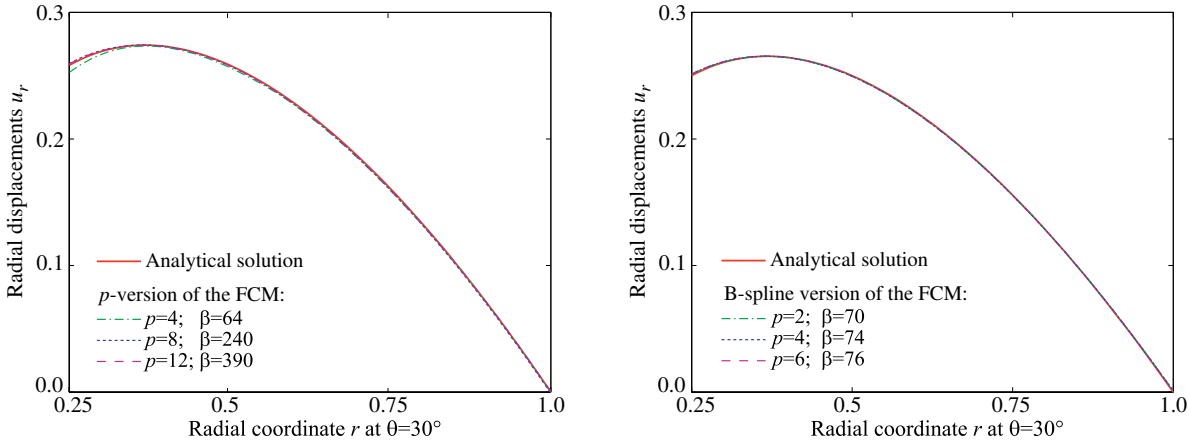
$$u_r = -\frac{r}{2}\ \frac{\ln r}{\ln 2} \tag{2.30a}$$

$$u_\theta = 0.0 \tag{2.30b}$$

the corresponding analytical stress fields can be derived from basic laws of elastostatics [170]

$$\sigma_r = \varepsilon_r = \frac{\partial u_r}{\partial r} = -\frac{1}{2}\ \frac{1}{\ln 2}\ [\ln r + 1] \tag{2.31a}$$

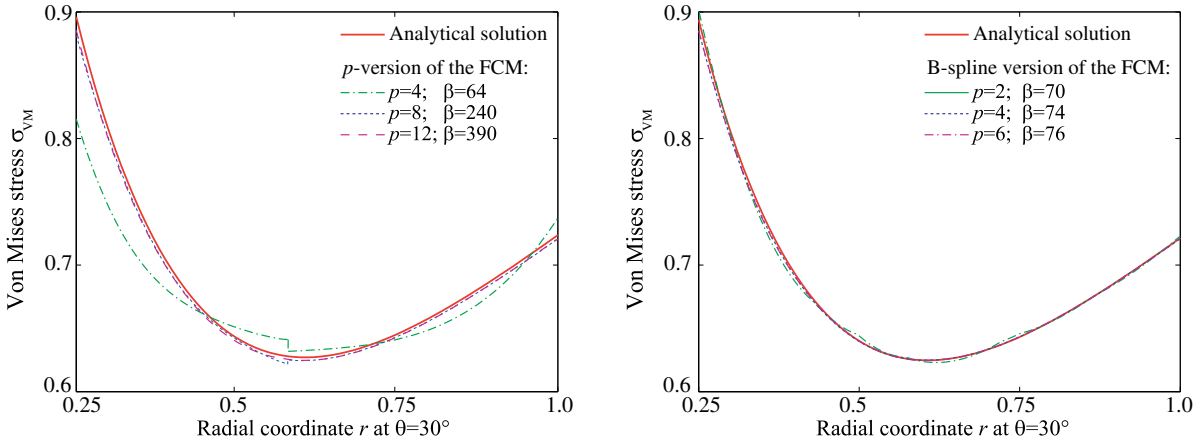
$$\sigma_\theta = \varepsilon_\theta = \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_r}{\partial r} = -\frac{1}{2}\ \frac{\ln r}{\ln 2} \tag{2.31b}$$



(a) Displacement results of the p-version mesh of Figure 2.20a.

(b) Displacement results of the knot span mesh of Figure 2.20b.

Figure 2.21: Radial displacements of the ring, plotted along the inclined cutline shown in Figure 2.19.



(a) Von Mises stress results of the p-version mesh of Figure 2.20a.

(b) Von Mises stress results of the knot span mesh of Figure 2.20b.

Figure 2.22: Von Mises stresses of the ring, plotted along the inclined cutline shown in in Figure 2.19.

$$\sigma_{r\theta} = 0 \tag{2.31c}$$

Note that the influence of material parameters does not appear explicitly due to the particular choice of Young's modulus E=1.0 and Poisson's ratio ν=0.0 (see Figure 2.19).

The discretizations with the p- and B-spline versions of the FCM used in this study consist of 4×4 p-version finite cells and 16×16 knot span cells, which are equipped with m=5 and m=3 levels of adaptive sub-cells as shown in Figures 2.20a and 2.20b, respectively. Boundary constraints are imposed in a weak sense according to Equations (2.7) and (2.8) via Nitsche's method. The penalty parameter β is chosen empirically for each discretization and polynomial degree in such a way that β is small, but still keeps the stiffness matrix positive definite. Solutions are obtained from the p-version mesh with polynomial degrees p=4 (322 dofs), p=8

(1,090 dofs) and p=12 (2,370 dofs) and from the B-spline mesh with p=2 (648 dofs), p=4 (800 dofs) and p=6 (968 dofs).

Figures 2.21 and 2.22 compare the radial displacement and the von Mises stress results, respectively, to the corresponding analytical reference obtained from Equation (2.29) and (2.30). The plots refer to a cutline that originates in the center of the ring and is inclined by an angle of 30° against the horizontal as shown in Figure 2.19 to avoid mesh induced symmetry effects. The displacement and stress results converge under p-refinement to the reference and confirm that the weak imposition of Dirichlet boundary conditions via Nitsche's method works well for both the p- and B-spline versions of the finite cell method. One can observe that for the examined discretizations, the B-spline version achieves a comparable quality in stresses with fewer degrees of freedom than the p-version due to its higher-order continuity.

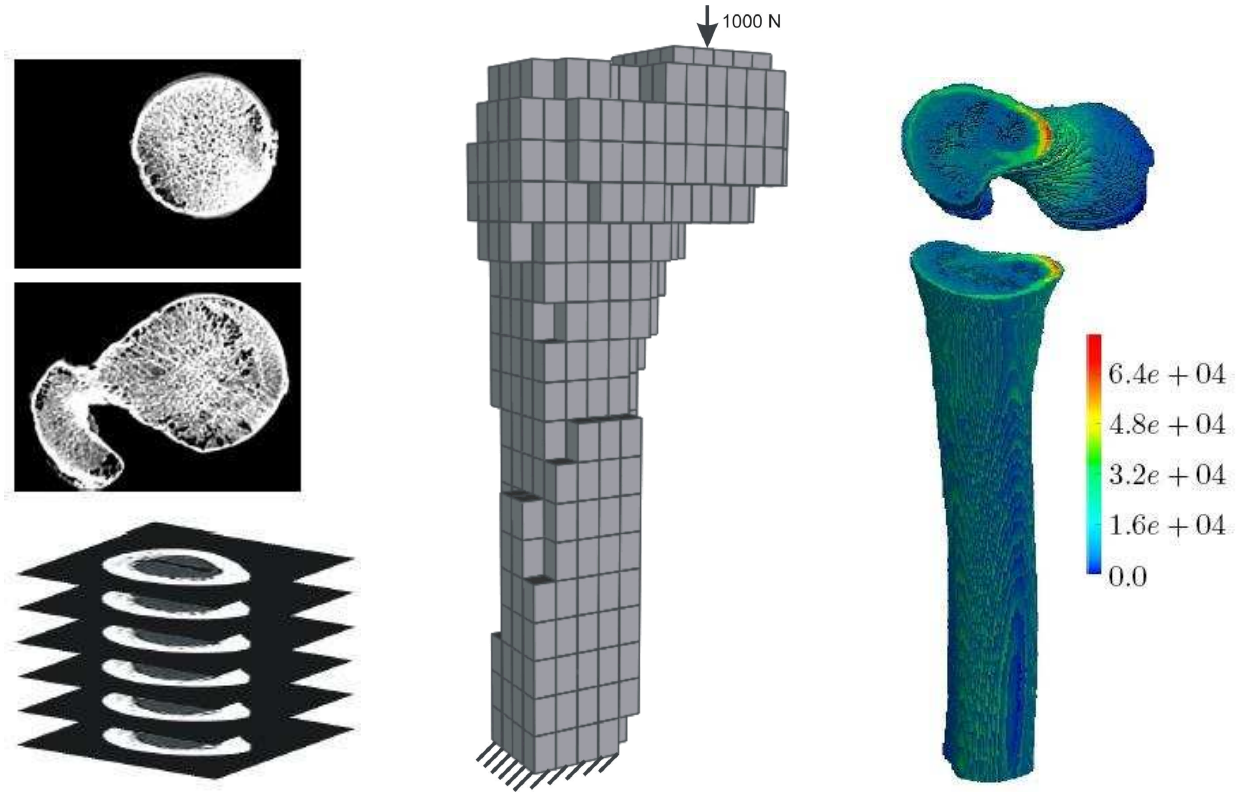
2.4.3 A proximal femur bone with inhomogeneous material

The human femur bone of Figure 2.23 is analyzed with the p- and B-spline versions of the finite cell method on the basis of voxel data derived from quantitative computed tomography (CT) scans. This kind of analysis is common in biomechanics to mimic patient-specific in-vivo behavior, which allows for a better understanding of bone stability and strength. The majority of research in this field is based on traditional finite elements (see e.g. [116, 174]), which are often limited in terms of accuracy and efficiency, the need for time-consuming segmentation of the CT data and an element-wise constant material assignment. Before the discussion of the FCM results, some details on the material model based on voxels are provided.

2.4.3.1 A voxel based model for the representation of inhomogeneous material

Voxels are volume elements, aligned in a structured spatial grid in Cartesian directions [2, 3, 38], each of which contains local material information. Voxel data sets have a limited resolution, as precise data is only available at the center of each volume element, and care has to be taken that the chosen voxel resolution represents changes in material parameters accurately enough for simulating the corresponding mechanical behavior. Voxel models of real structures can be directly obtained from computed tomography (CT), magnetic resonance tomography (MRT) or ultrasound scans. For a concise review of medical imaging technologies, see for example [146]. The CT scan of the femur shown in Figure 2.23a represents a discrete model [1] of the continuous material distribution with a resolution of $1024 \times 1024 \times 183$ voxels, each providing a measure for the radiodensity of the cortical and trabecular bone regions. First, a radiodensity limit needs to be defined, which separates material from void regions. Since a heterogeneous isotropic material behavior is sufficient to accurately represent the experimental observations [216], the radiodensity scale lying above that limit is translated into a corresponding scale of equivalent material parameters, i.e. Young's modulus E and Poisson's ratio ν, which are then stored independently for each voxel location according to the radiodensity identified there. With the relations given in [116], this results in a different Young's modulus for each voxel and a constant Poisson's ratio of 0.3.

[1] Courtesy of Prof. Zohar Yosibash, Dept. of Mechanical Engineering, Ben-Gurion University, Beer-Sheva, Israel; http://www.bgu.ac.il/∼zohary/



(a) Slices of the CT scan, indicating the thin cortical shell outside and the inner trabecular structure.

(b) Finite cell mesh of the bone, consisting of 678 p-version finite cells, and boundary conditions.

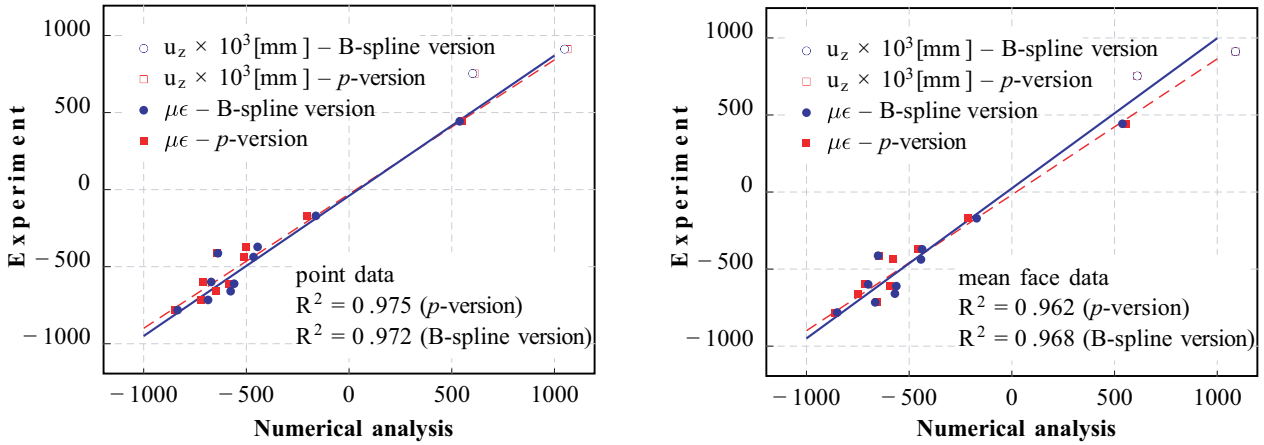
(c) Von Mises equivalent strain distribution $[\mu\varepsilon]$, evaluated for each voxel of the physical domain.

Figure 2.23: The finite cell method applied for analysis of a CT based proximal femur bone.

2.4.3.2 FCM analysis with the *p*- and B-spline versions

In the p-version of the FCM, the bone is embedded in 678 finite cells, each covering $40 \times 40 \times 10$ voxels (see Figure 2.23b). Cells that are completely outside the bone are neglected to minimize the computational effort. An equivalent voxel resolution per cell is chosen for the B-spline version of the FCM, but due to the support of the B-spline functions in adjacent cells, none of the cells of the fictitious domain Ω_{fict} can be removed, resulting in a full rectangular grid of 1,729 knot span cells. The inhomogeneity of the material is captured by $20 \times 20 \times 10$ sub-cells per knot span cell, applying 2^3 integration points per sub-cell. Thus the material properties of each voxel within the bone contributes to the analysis model. Integration points outside the physical domain Ω_{phys} are penalized with $\alpha = 10^{-4}$ in the sense of Figure 2.1. The proximal femur bone is loaded with 1,000 N on top of the femur head and Nitsche's method according to Equations (2.7) and (2.8) is applied to weakly satisfy the homogeneous Dirichlet boundary conditions at the distal face (see Figure 2.23b) in both FCM versions. Figure 2.23c illustrates the von Mises equivalent strains obtained from the discretization with the p-version of the FCM, visualized pointwise for each voxel.

For validation of the numerical results, the linear elasticity response in terms of strains is compared with results from a corresponding in-vitro experiment [168, 202]. Figure 2.24



(a) Pointwise values.

(b) Element face average.

Figure 2.24: Experimental results vs. simulation results obtained with the p- and B-spline versions of the FCM. The target parameter $\mu\varepsilon$ denotes von Mises equivalent strains [168, 202, 216].

provides a comparison of the numerical predictions and the experimental results for points located at the strain gauge position of the experiment and averaged results over small surface patches embedding the measuring points. A linear regression analysis shows a good correlation (> 0.96) between numerical predictions and measurements for both the p- and B-spline versions of the FCM. Sufficient convergence was found for both FCM versions at p=4. For this polynomial degree, the 678 finite cells of the p-version result in about 35,000 degrees of freedom, whereas the 1,729 knot span cells of the B-spline version only require about 13,000 degrees of freedom to yield equivalent results at the same accuracy level. The simulation results demonstrate that from an overall point of view, the p- and B-spline version of the finite cell method achieve comparable accuracy in the framework of a very complex example. Readers interested in a detailed description of experiments and FCM analysis are referred to the in-depth study by Ruess and co-workers [168].

Chapter 3

The finite cell method for geometrically nonlinear problems

We now turn from the small deformation range of linear elasticity to geometrically nonlinear elasticity, combining the finite cell method with the well-established framework of nonlinear finite element technology. Today, the large deformation analysis of structures in the framework of the finite element method can be considered a standard procedure, for which underlying principles of continuum mechanics are thoroughly understood and well documented [76, 107, 133, 136, 137, 204] and a multitude of reliable and efficient numerical algorithms exist [26, 32, 34, 64, 110, 214] and are available in commercial tools [128, 132, 190]. In the scope of the present work, no attempt is made to give an overview of the vast amount of pertinent aspects of nonlinear structural mechanics. Instead, we will confine ourselves to briefly introduce those components that we will actually make use of, and extensively refer to the wealth of existing literature for a more detailed description and for aspects that are not part of the present study.

In the following, we first introduce a geometrically nonlinear finite cell formulation in principal directions, based on the logarithmic strain measure. We then demonstrate that a standard FCM approach, applying the same kinematics over the complete domain, leads to the loss of uniqueness of the nonlinear deformation map within the fictitious domain. This motivates its modification by the idea of deformation resetting, which assumes the initial reference configuration within the fictitious domain after each Newton iteration. A first simple numerical experiment in one dimension indicates that this intervention allows for stable nonlinear finite cell analysis with both the p- and B-spline versions, preserving the full range of advantages of the linear elastic FCM, in particular high rates of convergence. Finally, we derive a geometrically nonlinear formulation of the penalty method for the imposition of unfitted Dirichlet constraints in elements cut by the geometric boundary.

3.1 A geometrically nonlinear formulation in principal directions

Turning to large deformation analysis, the linear elastic formulation of Section 2 is extended in a straightforward manner to a geometrically nonlinear FCM formulation based on logarithmic strains. The adaptation directly follows the standard finite element formulation, for which details can be found in [26, 34, 64, 110, 214].

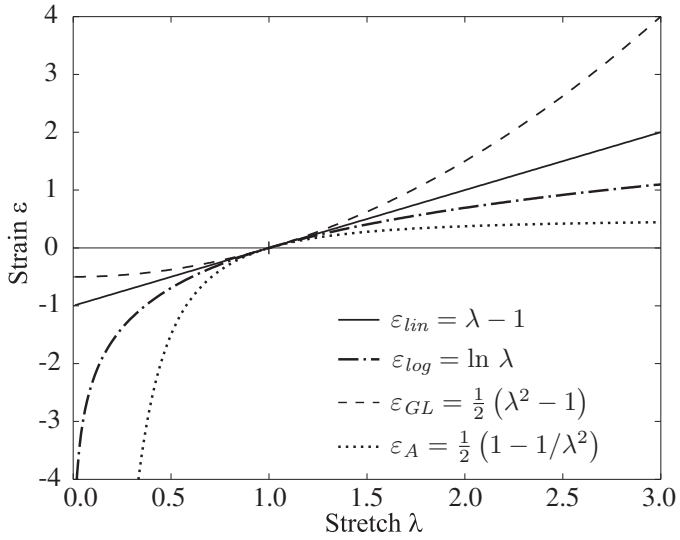


Figure 3.1: Engineering, logarithmic, Green-Lagrange and Almansi strain measures ε_{lin}, ε_{log}, ε_{GL}, ε_A, respectively, in one dimension (adapted from [32]).

3.1.1 Kinematics

In geometrically nonlinear statics, the deformation map $\boldsymbol{\varphi}$ describes the motion of each material point from its initial reference configuration $\boldsymbol{X}$ to its deformed configuration $\boldsymbol{x}$.

$$\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}) \tag{3.1}$$

The deformed configuration $\boldsymbol{x}$ of Equation (3.1) is also denoted as the spatial configuration. The deformation map Equation (3.1) is required to be one-to-one [34, 107]. The displacement field $\boldsymbol{u}$ and the deformation gradient $\boldsymbol{F}$ follow as

$$\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X} \tag{3.2}$$

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} \tag{3.3}$$

In view of its polar decomposition, $\boldsymbol{F}{=}\boldsymbol{V}\boldsymbol{R}$ can be represented by the rotation tensor $\boldsymbol{R}$ and the spatial stretch tensor $\boldsymbol{V}$. Therefore, $\boldsymbol{V}^2$ can be obtained from the left Cauchy-Green tensor

$$\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^T = (\boldsymbol{V}\boldsymbol{R})(\boldsymbol{R}^T\boldsymbol{V}) = \boldsymbol{V}^2 \tag{3.4}$$

Evaluation of the principal directions of $\boldsymbol{V}^2$ yields the orthogonal spatial Eigenvector triad $\{\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3\}$ and the associated Eigenvalues $\{\lambda_1^2, \lambda_2^2, \lambda_3^2\}$, which are identified as the squared principal stretches. The spatial logarithmic strain tensor $\boldsymbol{\varepsilon} = \ln \boldsymbol{V}$ in spectral form reads

$$\boldsymbol{\varepsilon} = \sum_{a=1}^{3} \ln \lambda_a \, \boldsymbol{n}_a \otimes \boldsymbol{n}_a \tag{3.5}$$

Logarithmic strains, also known as true or natural strains, represent a suitable strain measure for the entire deformation range [32, 34, 64], and are compared in one dimension to other common nonlinear strain measures in Figure 3.1.

In the framework of the finite cell concept as described in Section 2, the nonlinear kinematic relations Equations (3.1) to (3.5) can be applied throughout the complete embedding domain Ω (see Figure 2.1). This approach is referred to as the standard geometrically nonlinear finite cell formulation in the following. Due to the discontinuous penalization parameter α of Equation (2.3), the analytical deformation map and corresponding displacements Equations (3.1) and (3.2) exhibit a weak discontinuity (kink) along the geometric boundaries of the physical domain Ω_{phys}. Accordingly, the analytical deformation gradient and related stretches and strains Equations (3.3) to (3.5) exhibit a strong discontinuity (jump).

3.1.2 Constitutive equations

The Hencky hyperelastic model is the finite logarithmic strain based extension of the standard linear elastic material, whose strain energy function Ψ reads [34, 64]

$$\Psi = \alpha \cdot \left[\frac{E}{2(1+\nu)} \left((\ln \lambda_1)^2 + (\ln \lambda_2)^2 + (\ln \lambda_3)^2 \right) + \frac{\nu E}{2(1+\nu)(1-2\nu)} (\ln J)^2 \right] \tag{3.6}$$

with $J = \det \boldsymbol{F} = \lambda_1 \lambda_2 \lambda_3$ and material parameters Young's modulus E and Poisson's ratio ν. In the standard FCM formulation, geometric nonlinearity is applied throughout the complete domain Ω. Therefore, Equation (3.6) is factorized by parameter α of Equation (2.3), which penalizes the strain energy function Ψ within the fictitious domain Ω_{fict}. The principal Cauchy stresses along principal axes a=$\{1, 2, 3\}$ follow with Equation (3.6) as

$$\sigma_a = \frac{1}{J} \frac{\partial \Psi}{\partial \ln \lambda_a} = \frac{\alpha}{J} \left[\frac{E}{(1+\nu)} \ln \lambda_a + \frac{\nu E}{(1+\nu)(1-2\nu)} \ln J \right] \tag{3.7}$$

The fourth order spatial elasticity tensor in Cartesian coordinates can then be computed as

$$c_{ijkl} = \sum_{a,b=1}^{3} \frac{1}{J} \frac{\partial^2 \Psi}{\partial \ln \lambda_a \partial \ln \lambda_b} \boldsymbol{\eta}_{aabb} - \sum_{a=1}^{3} 2\sigma_a \boldsymbol{\eta}_{aaaa} + \sum_{\alpha,\beta=1}^{3} \frac{\sigma_a \lambda_b^2 - \sigma_b \lambda_a^2}{\lambda_a^2 - \lambda_b^2} \left(\boldsymbol{\eta}_{abab} + \boldsymbol{\eta}_{abba} \right) \tag{3.8}$$

implying the fourth order dyadic product $\boldsymbol{\eta}_{ijkl} = \boldsymbol{n}_i \otimes \boldsymbol{n}_j \otimes \boldsymbol{n}_k \otimes \boldsymbol{n}_l$. Due to the strong discontinuity in parameter α, all derived analytical quantities exhibit a loss of regularity along geometric boundaries.

3.1.3 Discretization and linearization

Taking into account Equations (3.1) to (3.8), the variational formulation of the standard FCM for large deformation elasticity can be derived from the principle of virtual work

$$\delta W(\boldsymbol{\varphi}, \delta \boldsymbol{u}) = \int_{\boldsymbol{\varphi}(\Omega)} \boldsymbol{\sigma} : \nabla_x \, \delta \boldsymbol{u} \, dv - \int_{\boldsymbol{\varphi}\left(\Omega_{phys}\right)} \boldsymbol{b} \cdot \delta \boldsymbol{u} \, dv - \int_{\boldsymbol{\varphi}(\Gamma_N)} \boldsymbol{t} \cdot \delta \boldsymbol{u} \, da \;=\; 0 \tag{3.9}$$

with body forces $\boldsymbol{b}$, traction vector $\boldsymbol{t}$ and test function $\delta \boldsymbol{u}$. Integrals are evaluated in the deformed configuration, where dv and da denote infinitesimal volume and area elements, respectively, since a spatial formulation is computationally more efficient in the case of p-version elements [72, 145, 188].

The basic kinematic quantities $\boldsymbol{u}$, $\delta\boldsymbol{u}$ and $\boldsymbol{F}$ can be discretized in the sense of Equations (2.4), (2.5) and by

$$\boldsymbol{F} = \boldsymbol{I} + \sum_{a=1}^{n} \nabla_{\boldsymbol{X}} N_a \boldsymbol{u}_a \tag{3.10}$$

where N_a denote n basis functions of a suitable high-order basis. In case of finite cells based on p-version basis functions, insertion of the displacement approximation Equation (2.4) and the reference configuration, interpolated subparametrically by Equation (2.17), into Equation (3.2) yields an interpolation of the deformed configuration

$$\boldsymbol{x} = \sum_{i=1}^{n_{vert}} N_i^{\mathrm{nod}}\, \boldsymbol{X}_i \;+\; \sum_{a=1}^{n} N_a\, \boldsymbol{u}_a \tag{3.11}$$

where n_{vert} denotes the number of element nodes and N^{nod} are the corresponding nodal basis functions. For finite cells based on high-order B-splines, the corresponding expression can be found with Equation (2.21) as

$$\boldsymbol{x} = \boldsymbol{X}_0 + h\,\boldsymbol{\xi} \;+\; \sum_{a=1}^{n} N_a\, \boldsymbol{u}_a \tag{3.12}$$

where h is the uniform width of the knot span cells and $\boldsymbol{X}_0$ denotes the physical origin in the parameter space $\boldsymbol{\xi}$.

Using these expressions in Equation (3.9), the discretized virtual work per high-order mode shape a can be formulated as the difference between the internal and external equivalent force vectors $\boldsymbol{f}^{int}$ and $\boldsymbol{f}^{ext}$, called residual $\boldsymbol{r}$

$$\delta W\left(\boldsymbol{\varphi}, N_a\delta\boldsymbol{u}_a\right) = \delta\boldsymbol{u}_a^T\left(\boldsymbol{f}_a^{int} - \boldsymbol{f}_a^{ext}\right) = \delta\boldsymbol{u}_a^T\boldsymbol{r}_a \tag{3.13}$$

$$\boldsymbol{f}_a^{int} = \int\limits_{\boldsymbol{\varphi}(\Omega)} \boldsymbol{B}_a^T\boldsymbol{\sigma}\; dv \tag{3.14}$$

$$\boldsymbol{f}_a^{ext} = \int\limits_{\boldsymbol{\varphi}\left(\Omega_{phys}\right)} N_a\,\boldsymbol{b}\; dv + \int\limits_{\boldsymbol{\varphi}(\Gamma_N)} N_a\,\boldsymbol{t}\; da \tag{3.15}$$

with the strain-displacement matrix $\boldsymbol{B}$ [17, 34, 108]. The linearization of the discretized weak form Equation (3.9) in the direction of an incremental displacement $\Delta\boldsymbol{u}$ can be expressed in terms of material and geometric parts $D\delta W_c$ and $D\delta W_\sigma$, respectively, as

$$D\delta W\left(\boldsymbol{\varphi}, N_a\delta\boldsymbol{u}_a\right)\left[N_b\Delta\boldsymbol{u}_b\right] = D\delta W_c + D\delta W_\sigma \tag{3.16}$$

$$D\delta W_c = \delta\boldsymbol{u}_a^T\left(\int\limits_{\boldsymbol{\varphi}(\Omega)} \boldsymbol{B}_a^T\boldsymbol{c}\,\boldsymbol{B}_b\; dv\right)\Delta\boldsymbol{u}_b \tag{3.17}$$

$$D\delta W_\sigma = \delta\boldsymbol{u}_a^T\left(\int\limits_{\boldsymbol{\varphi}(\Omega)} \left(\nabla_{\boldsymbol{X}} N_a\cdot\boldsymbol{\sigma}\,\nabla_{\boldsymbol{X}} N_b\right)\boldsymbol{I}\; dv\right)\Delta\boldsymbol{u}_b \tag{3.18}$$

with $\boldsymbol{c}$ being the matrix representation of the spatial elasticity tensor Equation (3.8) [34]. In Equations (3.17) and (3.18), the expressions in brackets can be identified as the entries $K_{c,ab}$ and $K_{\sigma,ab}$ of the material and geometric tangent stiffness matrices, respectively. Combining Equations (3.9) and (3.13) yields

$$(\boldsymbol{K}_c + \boldsymbol{K}_\sigma)\,\Delta\boldsymbol{u} = -\boldsymbol{r} \tag{3.19}$$

from which the classical Newton-Raphson procedure can be derived. In each Newton step, the linearized system Equation (3.19) is solved for $\Delta\boldsymbol{u}$, which updates the total displacements $\boldsymbol{u}$, until the norm of the residual vector $\boldsymbol{r}$ has converged below a tolerance close to zero.

In the scope of the present work, the stiffness contribution due to deformation dependent loads [34, 214] is not discussed. For a general overview and a treatment in the framework of standard finite elements, we refer to [103, 139, 184, 189, 215], for a treatment within the framework of the B-spline version and the p-version of the FCM, see [178] and [179], respectively.

3.2 The standard geometrically nonlinear finite cell formulation

With respect to the general three-dimensional formulation of the previous section, the geometrically nonlinear formulation in principal directions for the one-dimensional rod of Figure 2.8 simplifies to

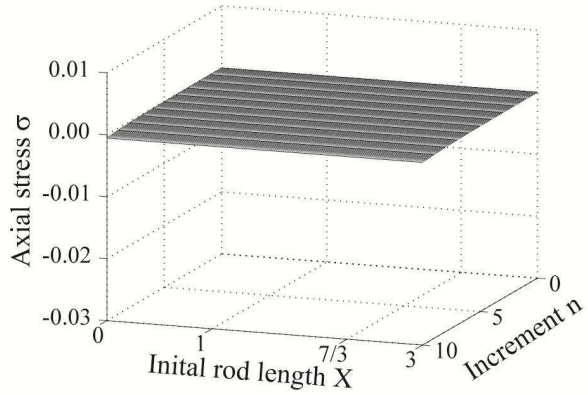
$$\Psi = \alpha \frac{E}{2} (\ln \lambda)^2 \tag{3.20}$$

$$\sigma = \alpha \frac{E}{J} \ln \lambda \tag{3.21}$$

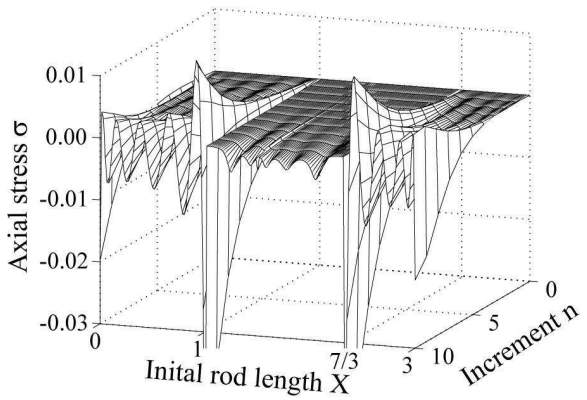
$$c = \alpha \frac{E}{J} - 2\sigma \tag{3.22}$$

with axial stretch λ and the determinant of the deformation gradient $J = \lambda^{1-2\nu}$ [34]. To illustrate the influence of large deformations within the fictitious domain, the sine load $f_{\sin}$ of Figure 2.8 is neglected for a moment and the prescribed displacement is set to a large value of Δu=1.0. Nonetheless, the corresponding physical stresses should be zero, since a rigid body movement of the right part of the rod is approximated. Thus, the resulting non-zero stresses directly reflect the modeling error due to finite α and the numerical error. In all computations of the present section, the total displacement load is divided into 10 increments, each of which needs two to four Newton-Raphson iterations to converge to a L^2-norm of the residual below 10^{-12}. Dirichlet boundary conditions are imposed strongly. The exact stress solution, which can be derived analytically according to [176], is plotted in Figure 3.2a for 10 displacement load increments between 0 and Δu and $\alpha = 10^{-5}$.

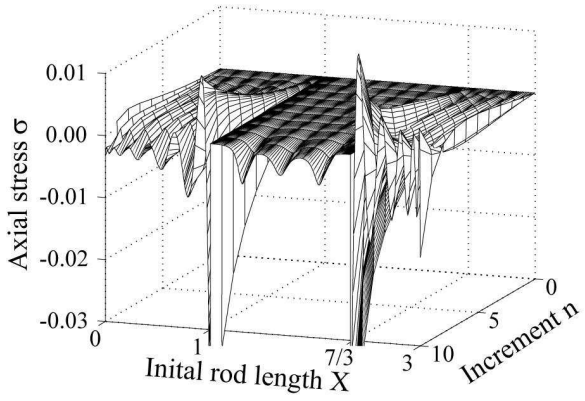
The standard finite cell method applies the same geometrically nonlinear formulation over the complete embedding domain Ω as described in Section 3.1, where the only difference between Ω_{phys} and Ω_{fict} constitutes the penalization of the strain energy function Ψ Equation (3.6) and its derivatives with α of Equation (2.3). However, numerical experiments with the one-dimensional rod of Figure 2.8 reveal that the smallest penalty parameter, which could be



(a) Analytical.



(b) p-version.



(c) B-spline version.

Figure 3.2: Stresses of the geometrically nonlinear rod example, obtained with the standard FCM formulation and p=15, are compared to the analytical solution with finite $\alpha = 10^{-5}$.

used successfully for each tested polynomial degree p, is α=10^{-4} and α=10^{-5} for the p- and B-spline versions, respectively. Corresponding solution fields obtained with two p-version finite cells and 16 knot span cells in the sense of Figure 2.8 are plotted in Figures 3.2b and 3.2c. For α smaller than these bounds, the determinant of the deformation gradient $\boldsymbol{F}$ falls below zero at some integration point within Ω_{fict}, which inevitably terminates the computation. From a mathematical point of view, this implies the loss of uniqueness of the deformation map, which is not one-to-one anymore. From a physical point of view, this can be interpreted as a penetration of material, which constitutes a severe violation of the principles of continuum mechanics [34, 107, 214]. With α as large as 10^{-4}, the penalization of Equation (2.3) is unable to sufficiently eliminate the influence of Ω_{fict}, so that a considerable modeling error is introduced. In addition, the strain energy contribution of Ω_{fict} is amplified by the nonlinear strain measure. Whereas engineering strains of linear elasticity are bound by definition to very small values $|\varepsilon_{lin}|\ll 1.0$, logarithmic strains of nonlinear elasticity are able to grow without bounds in order to yield physically meaningful measures for very large deformation states. However, in case of large deformations in the fictitious domain, nonlinear strains thus act as an additional counterbalance to α and increasingly outweigh the penalization.

As a consequence, the contribution of Ω_{fict} to the total strain energy Equation (2.22) grows, so that the nonlinear FCM scheme tries to accurately fit the solution in both the fictitious and physical domains due to the best approximation property discussed in Section 2.3. Thus, solution fields do not extend smoothly into the fictitious domain, but develop large oscillations throughout the discontinuous cells (see Figure 3.2). The corresponding convergence deteriorates to a low algebraic rate, which is a well-known issue for high-order elements with inter-element discontinuities [197]. Numerical experiments indicate this standard FCM formulations based on other nonlinear strain measures (see [32, 34, 64] and Figure 3.1) in combination with corresponding constitutive equations affect the stability of the deformation map in the same way. The standard FCM formulation of Section 3.1 thus suffers from a conflict of interest between stable analysis (increase of α) on the one hand and a reduction of the contribution of Ω_{fict} (decrease of α) on the other.

3.3 A modified formulation based on repeated deformation resetting

To avoid the stability problem of the deformation map in the fictitious domain Ω_{fict}, the physical consistency of the geometrically nonlinear FCM formulation is restricted to the physical domain Ω_{phys}. Within Ω_{fict}, the formulation is manipulated in such a way that stable geometrically nonlinear analysis with very small values $\alpha<10^{-10}$ is possible.

3.3.1 Motivation and implementation

Numerical experiments reveal that problems with the uniqueness of the deformation map occur at the location of maximum deformation within the fictitious domain Ω_{fict}. This motivates the following simple manipulation after each Newton iteration i [180, 181]

$$\boldsymbol{\varphi}^i(\mathbf{X}) = \begin{cases} \boldsymbol{x}^i & \text{deformed configuration } \forall \boldsymbol{X} \in \Omega_{phys} \\ \boldsymbol{X} & \text{reset to reference configuration } \forall \boldsymbol{X} \in \Omega_{fict} \end{cases} \tag{3.23}$$

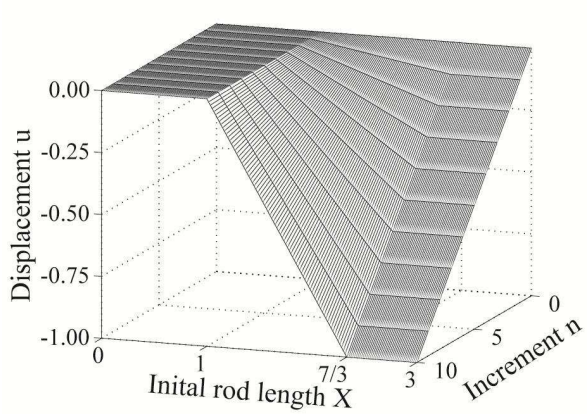
where $\boldsymbol{\varphi}^i$ and $\boldsymbol{x}^i$ denote the deformation map and the deformed configuration after the i^{th} Newton step. According to Equation (3.23), the deformation is repeatedly reset to the initial undeformed state to erase the complete deformation history within Ω_{fict}. Thus, at the beginning of the $(i+1)^{th}$ Newton iteration, the deformation gradient Equation (3.3) in Ω_{fict} is defined as

$$\boldsymbol{F} = \boldsymbol{I} \tag{3.24}$$

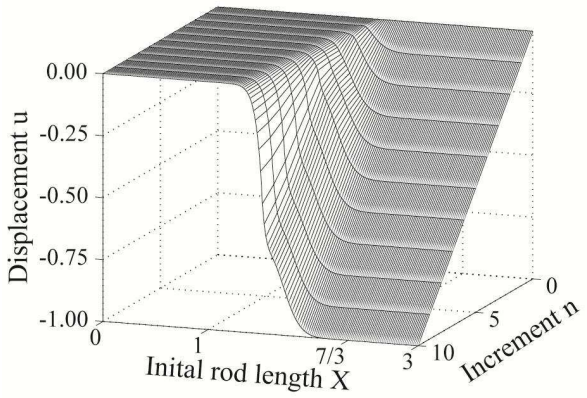
with $\boldsymbol{I}$ being the unit tensor. From Equation (3.24), the corresponding principal stretches Equation (3.5) and stresses Equation (3.7) directly follow as

$$\lambda_a = 1.0, \; a = \{1, 2, 3\} \tag{3.25}$$

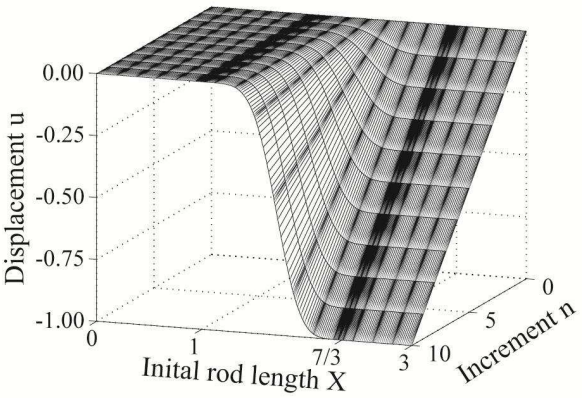
$$\boldsymbol{\sigma} = \mathbf{0} \tag{3.26}$$



(a) Reference.

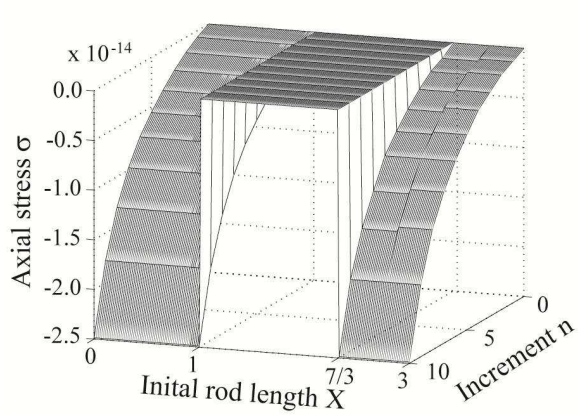


(b) p-version.

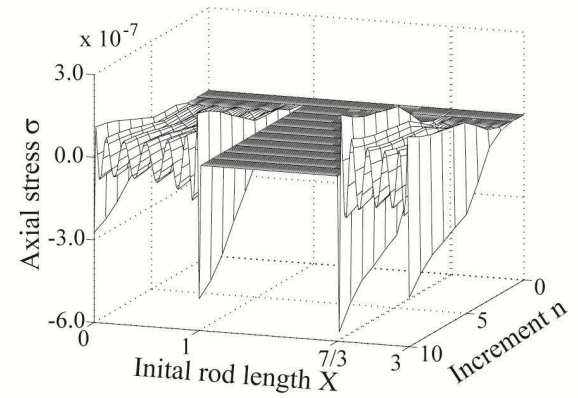


(c) B-spline version.

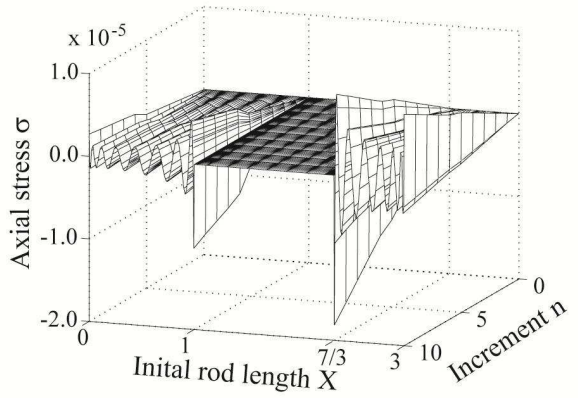
Figure 3.3: Displacements of the geometrically nonlinear rod example, obtained with the modified FCM formulation and p=15. The reference solution has been computed with three conforming standard linear elements.



(a) Reference.



(b) p-version.



(c) B-spline version.

Figure 3.4: Stresses of the geometrically nonlinear rod example, obtained with the modified FCM formulation and p=15. The reference solution has been computed with three conforming standard linear elements.

The resulting formulation is inconsistent, because it violates the principles of continuum mechanics and its analytical solution in Ω_{fict} turns unphysical. However, in the sense of the fictitious domain approach (see Figure 2.1), it does not affect the physical consistency and accuracy of the solution in the physical domain Ω_{phys}, provided that the influence of Ω_{fict} and its contribution to the total strain energy Equation (2.22) are extenuated by a sufficiently strong penalization.

The assumption of Equation (3.24) supersedes the calculation of the deformation gradient, so that any stability issues resulting from the numerical computation of the deformation gradient are automatically avoided and a very small penalty parameter of α=10^{-15} can be applied without any further problems. The corresponding stress and displacement fields obtained with the p- and B-spline versions are plotted in Figures 3.3 and 3.4, respectively. The repeated resetting of the deformation allows for a smooth extension of the FCM solution fields into the fictitious domain despite the presence of discontinuities in the corresponding analytical solutions (see Figures 3.3 and 3.4). The oscillatory behavior demonstrated in Figure 3.2 for the stress solutions obtained with the standard FCM formulation is still present, but is considerably reduced by several orders of magnitude.

For a computationally efficient implementation of the deformation resetting, the coincidence of linear and geometrically nonlinear elasticity at the deformation and stress free reference configuration can be exploited [26, 34, 64, 214]. Since the deformation resetting switches Ω_{fict} back to its reference configuration after each Newton iteration, it is fully equivalent to carrying out repeated linear elastic computations

$$\delta W = \begin{cases} \delta W\left(\boldsymbol{\varphi}, \delta \boldsymbol{u}\right) & \text{large deformation } \forall \boldsymbol{X} \in \Omega_{phys} \\ \delta W\left(\boldsymbol{u}, \delta \boldsymbol{u}\right) & \text{small displacements } \forall \boldsymbol{X} \in \Omega_{fict} \end{cases} \tag{3.27}$$

Numerical experiments show that provided a sufficiently small penalty parameter, the quadratic rate of convergence of the Newton algorithm can be fully maintained.

We also modified the standard geometrically nonlinear FCM formulation in an even simpler way, just setting a penalized unit matrix at Gauss points in the fictitious domain. According to our experience, this strategy also works for moderate deformations, but destroys the quadratic rate of convergence of the Newton algorithm, and does not converge at all for larger deformations (while the FCM with deformation resetting does). Even if considered as a "trick", deformation resetting will be shown to yield physically consistent results in the physical domain for a range of numerical tests. Therefore, we believe that from the viewpoints of intuitive derivation, easy implementation, accuracy and computational efficiency, the methodology described as deformation resetting is quite optimal.

3.3.2 High rates of convergence in energy measure

To test convergence in energy measure in terms of Equations (2.23) and (2.24) [120], the uniaxial rod of Figure 2.8 is considered with sine load $f_{\sin}$ and Δu=1.0. An overkill discretization with 1,000 cubic finite elements taking into account the left rod yields U_{ex}=$1.17182588 \cdot 10^{-5}$. The convergence for p-refinement in the p- and B-spline versions with a standard geometrically nonlinear formulation is plotted in Figure 3.5. It illustrates the convergence decay to a low algebraic rate of around q=1.0 on average, which can be attributed to the modeling error introduced by insufficient penalization in conjunction with oscillatory stresses (see Figure

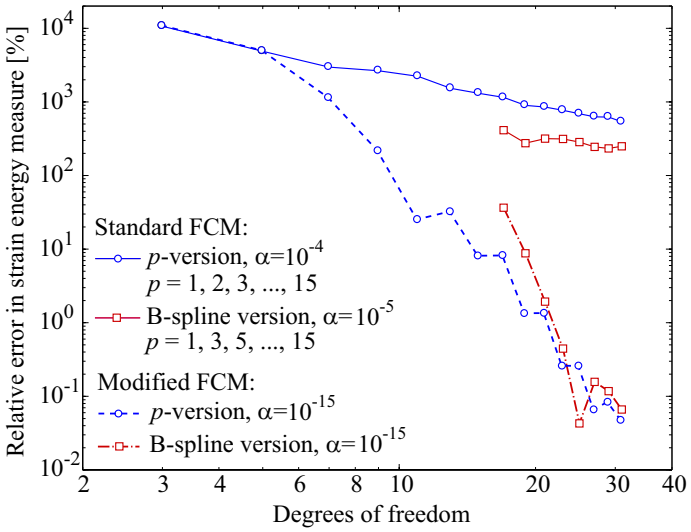


Figure 3.5: Convergence of the nonlinear example of the one-dimensional rod.

3.2). The modified geometrically nonlinear formulation allows for a considerable decrease of the penalty parameter to α=10^{-15}, which reduces the modeling error and the oscillatory behavior in stresses considerably (see Figures 3.4 and 3.5). Using the modified formulation, both p- and B-spline versions of the FCM are able to achieve exponential convergence with maximum rates of q=18.13 and q=28.09, respectively (see Figure 3.5). The flattening of the convergence curves indicate a take-over of the influence from Ω_{fict} at a value of around 0.04%. Under the assumption that the rate of the standard FCM formulation could be continued, approximately 30,000 times as many degrees of freedom (around 1 million) would be required to achieve a comparable level of accuracy.

3.4 The geometrically nonlinear penalty method for the weak imposition of Dirichlet constraints

In the one-dimensional example shown so far, Dirichlet boundary conditions could be imposed strongly by standard finite element techniques. More complex examples in two and three dimensions require their imposition along boundaries cutting through cells, which can be achieved in a weak sense with the help of Nitsche's method as presented in Sections 2.1.3 and 2.4.2 for linear elastic problems.

3.4.1 Penalty vs. Nitsche's method

The penalty method [11, 13, 220] is a subset of Nitsche's method and can be obtained from Equations (2.7) and (2.8) by omitting the so-called consistency terms that involve the computation of the normal vector $\boldsymbol{n}$ [78, 84]. The penalty method is based on the remaining terms that involve the penalty parameter β. It has similar advantages as Nitsche's method, for example it creates no additional unknowns and preserves the banded structure, symmetry and positive definiteness of the stiffness matrix. However, in contrast to Nitsche's method, it

is not consistent in a variational sense [78, 84] and can lead to strongly ill-conditioned system matrices. At least the latter does not affect the finite cell method in the present form, since system matrices are already ill-conditioned due to the penalization of the fictitious domain with parameter α of Equation (2.3) and the corresponding system of linearized equations requires a robust linear solver, which can handle the strong ill-conditioning.

In the framework of geometrically nonlinear problems, Nitsche's method requires the consistent linearization of all additional terms in Equations (2.7) and (2.8). Due to the missing consistency terms, the linearization of the penalty method is considerably simplified. Therefore, in the scope of the present work, the penalty method is applied in the following for the weak imposition of Dirichlet boundary conditions in both the p- and B-spline versions of the geometrically nonlinear finite cell method.

3.4.2 Discretization and linearization

The penalty method in variational form [13, 220] is based on the addition of a penalty term δW_p to the geometrically nonlinear variational equation of the principle of virtual work Equation (3.9) [17, 108]. The penalty term, obtained by integration over the Dirichlet boundary Γ_D can be formulated in the reference configuration as

$$\delta W_p\left(\boldsymbol{u}, \delta\boldsymbol{u}\right) = \beta \int_{\Gamma_D} \left(\boldsymbol{u} - \hat{\boldsymbol{u}}\right) \cdot \delta\boldsymbol{u} \, dA \tag{3.28}$$

where $\boldsymbol{u}$, $\hat{\boldsymbol{u}}$ and $\delta\boldsymbol{u}$ denote vectors of unknown displacements, prescribed displacements on Γ_D and test functions, and β is a scalar penalty parameter. The consistent derivation of the penalty method is based on a constrained minimization problem, formulated with the help of Lagrange multipliers, for which Equation (3.28) constitutes an approximation (see for example [17, 108] for an overview and [28] for details). Note that inhomogeneous Dirichlet boundary conditions, which constrain displacements in one special direction, can be easily implemented by applying the penalty parameter β only to the vector components of the constrained direction.

Inserting Equations (2.4) and (2.5) in Equation (3.28), the discretized virtual work of the penalty term δW_p at a known displacement state $\boldsymbol{u}$ results in the internal and external equivalent force vectors $\boldsymbol{f}_p^{int}$ and $\boldsymbol{f}_p^{ext}$, respectively, which can be expressed per high-order mode shape a as

$$\boldsymbol{f}_{a,p}^{int} = \beta \int_{\Gamma_D} \boldsymbol{u} \, N_a \, dA \tag{3.29}$$

$$\boldsymbol{f}_{a,p}^{ext} = \beta \int_{\Gamma_D} \hat{\boldsymbol{u}} \, N_a \, dA \tag{3.30}$$

The linearization of the discretized virtual work of the penalty term Equation (3.28) in the direction of an incremental displacement $\Delta\boldsymbol{u}$ can be found by taking the Gâteaux derivative [34, 107], which simply yields

$$D\delta W_p = \delta\boldsymbol{u}_a^T \left(\beta \int_{\Gamma_D} N_a N_b \, dA \right) \Delta\boldsymbol{u}_b \tag{3.31}$$

The expression in brackets can be identified as the entries $K_{p,ab}$ of the penalty contribution to the tangent stiffness matrix. Equation (3.31) is independent of the current state of deformation $\boldsymbol{u}$ and remains therefore constant throughout all iterations. Note that all quantities resulting from the penalty term are evaluated in the reference configuration [34, 107], since a push-forward to the deformed configuration leads to more involved expression due to the vector format of the area mapping (Nanson's formula) [107].

Combining the penalty tangent stiffness $\boldsymbol{K}_p$ with material and geometric tangent stiffnesses $\boldsymbol{K}_c$ and $\boldsymbol{K}_\sigma$, respectively, and adding the equivalent internal and external forces Equations (3.29) and (3.30) to the right hand side results in the following discrete system of equations

$$(\boldsymbol{K}_c + \boldsymbol{K}_\sigma + \boldsymbol{K}_p)\,\Delta\boldsymbol{u} = -\boldsymbol{r} \tag{3.32}$$

$$\boldsymbol{r} = \left(\boldsymbol{f}^{int} + \boldsymbol{f}_p^{int}\right) - \left(\boldsymbol{f}^{ext} + \boldsymbol{f}_p^{ext}\right) \tag{3.33}$$

from which the classical Newton-Raphson iterative procedure can be derived in the sense of Equation (3.19).

Chapter 4

Large deformation analysis with the finite cell method

In the following, we present a range of numerical benchmarks in two and three dimensions, which demonstrate stability, accuracy, physical consistency and exponential rates of convergence for the modified geometrically nonlinear FCM formulation, based on the deformation resetting idea. Benchmark results obtained with the finite cell method are compared to overkill solutions derived with standard linear quadrilateral and quadratic tetrahedral elements on conforming meshes, which are provided by the open-source nonlinear finite element code FlagShyp [86], developed by Bonet and Wood [34]. Corresponding mesh generation is accomplished with the meshers Netgen [142] and Visual Domesh [208].

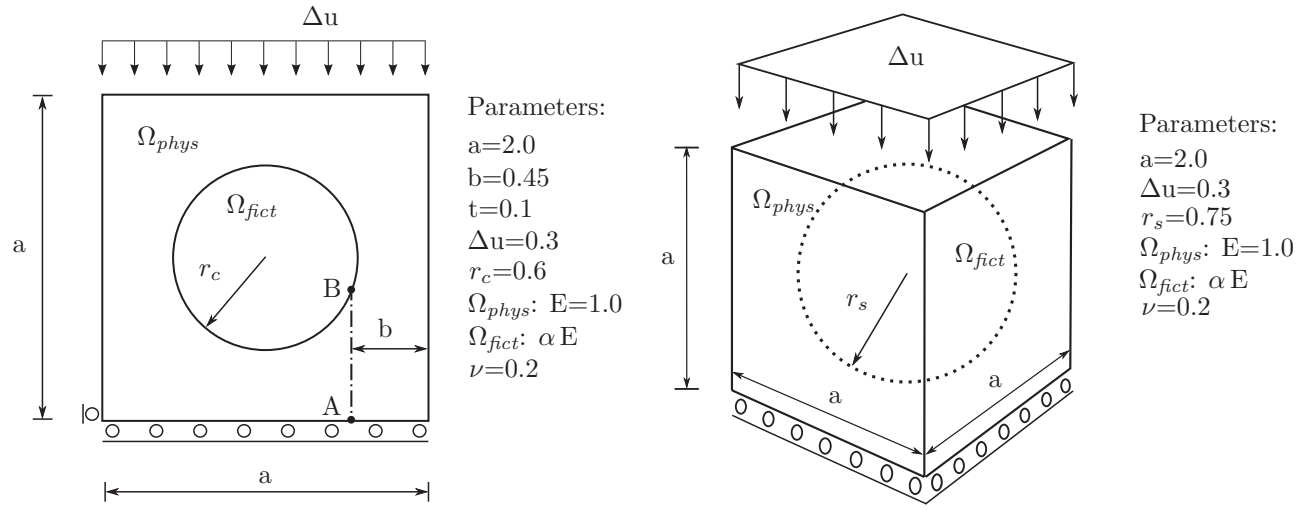
Furthermore, we examine the behavior of the FCM under severe mesh distortion, which is fundamental for the representation of very large deformation states, and present the application oriented large deformation analysis of a metal foam sample. The latter demonstrates the capability of the FCM to directly operate on the voxel based geometrical model of the complex foam structure, which illustrates the potential of the FCM as a seamless geometry-through-analysis procedure. We also discuss briefly the aspects of software implementation, hardware requirements and computational performance of the finite cell method. All examples are analyzed with both the p- and B-spline versions of the geometrically nonlinear finite cell method, which allows for a comparison of the two FCM versions in the last section, highlighting specific strengths of each approximation basis within the finite cell framework.

4.1 Standard benchmarks in two and three dimensions

The key advantage of the finite cell method are high rates of convergence at almost no meshing cost. The following numerical benchmarks demonstrate that the geometrically nonlinear finite cell method based on deformation resetting is able to achieve exponential rates of convergence, when a p-refinement of its high-order approximation bases is performed.

4.1.1 Plate with a circular hole

The first example problem consists of a two-dimensional square plate in plane stress, which is perforated by a circular hole. Material and geometric parameters as well as boundary

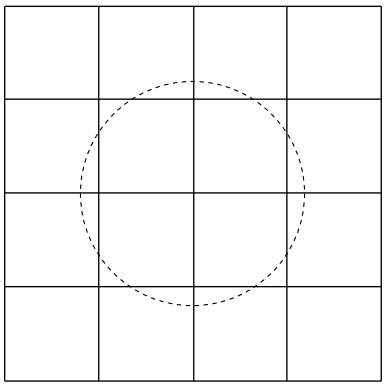


(a) 2D plate with circular hole. **(b)** 3D cube with spherical hole.

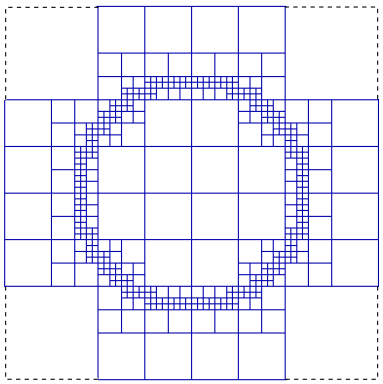
Figure 4.1: Benchmark examples: Parameters are width/height a; plate thickness t; radius r; Young's modulus E; domains Ω_{phys} and Ω_{fict}; Poisson's ratio ν; prescribed displacements Δu; cut line A-B

conditions are given in Figure 4.1a. For the FCM computations, the origin of the coordinate system is placed in the center of the circular hole. Thus, its geometry can be implicitly represented by the inequality $X^2 + Y^2 \leq r_c^2$, which allows for an efficient point location query at each Gauss point to determine the corresponding penalty parameter α in terms of Equation (2.3). The complete domain Ω is discretized by a structured mesh of 4×4 p-version finite cells and 25×25 knot span cells shown in Figures 4.2a and 4.3a, respectively, and complemented by integration sub-cells displayed in Figures 4.2a and 4.3b. The adaptive structure of the sub-cells, each of which is equipped with $(p+1)\times(p+1)$ Gauss points, leads to an aggregation of integration points around geometric boundaries, so that the discontinuity in α can be accurately resolved. To minimize the integration error, a sub-cell quadtree with overly large depths m=8 and m=4 is applied throughout all computations with the p- and B-spline versions, respectively. Dirichlet boundary conditions can be imposed strongly, since Dirichlet boundaries conform to cell boundaries. For the solution of the geometrically nonlinear system, the displacement load is cut into three equally sized increments, each of which requires not more than three to four Newton iterations in both FCM versions to converge to a value of the L^2-norm of the residual below 10^{-12}. The modified FCM formulation allows for a penalization of the fictitious domain with α=10^{-15}.

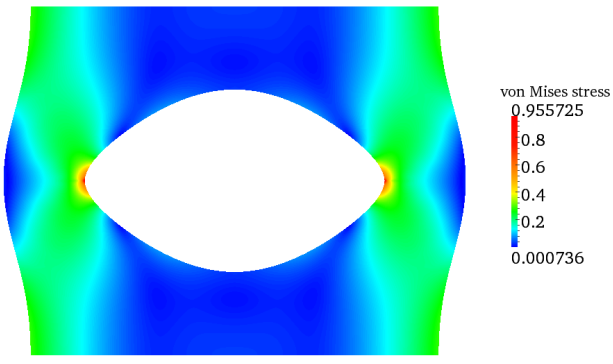
A first impression of the quality of the geometrically nonlinear FCM solutions can be obtained from the von Mises stress plots shown in Figures 4.2c and 4.3c obtained with polynomial degree p=15. Both FCM versions are able to accurately localize the typical stress concentration phenomenon at the vertical edges of the circular hole and are free of stress oscillations in the physical domain. For the computation of a reference solution in terms of strain energy, we use the symmetry in geometry and boundary conditions to reduce the plate to one quarter of the original system. The considered FlagShyp discretization with a mesh conforming to the geometric boundaries consists of 224,312 standard linear quadrilateral elements and 449,723 degrees of freedom, taking into account the physical domain Ω_{phys} only. Figures 4.4a and 4.4b give a more detailed view of the quality of the stress solution by plotting von Mises stresses along cut line A-B (see Figure 4.1), obtained from the p- and B-spline discretizations,



(a) FCM mesh consisting of 4×4 p-version cells.

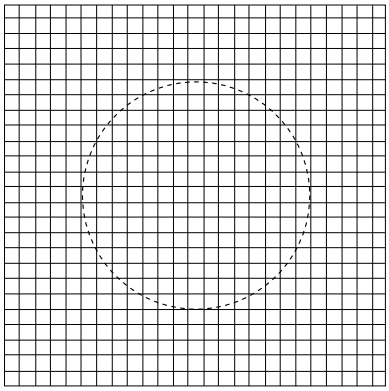


(b) Integration sub-cells displayed up to level k=4.

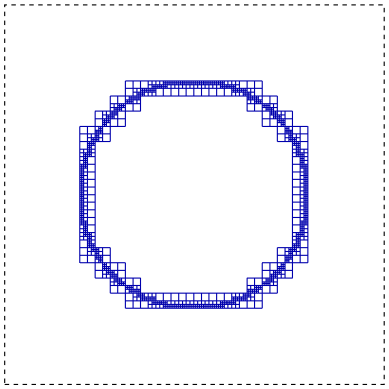


(c) Von Mises stresses on the deformed configuration, obtained with p=15.

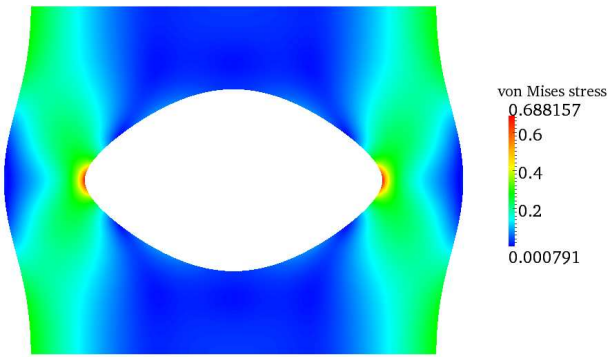
Figure 4.2: 2D plate with circular hole, computed with the p-version of the FCM.



(a) FCM mesh consisting of 25×25 knot span cells.

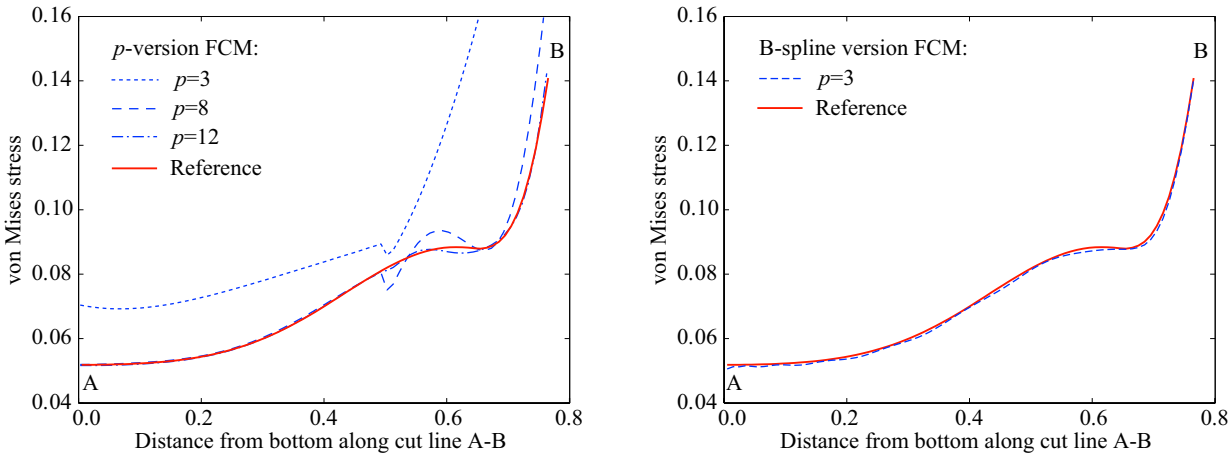


(b) Integration sub-cells displayed up to level k=3.



(c) Von Mises stresses on the deformed configuration, obtained with p=15.

Figure 4.3: 2D plate with circular hole, computed with the B-spline version of the FCM.



(a) p-version of the FCM.

(b) B-spline version of the FCM.

Figure 4.4: Von Mises stresses along cut line A-B of the 2D plate with circular hole, obtained with the p-version and B-spline discretizations shown in Figures 4.2 and 4.3, respectively.

respectively, and the FlagShyp reference solution. Both FCM versions converge towards the reference, while stresses are accurate and smooth up to the geometric boundary at point B. Comparing the p-version solution of p=12 (2,270 dofs) and the B-spline solution of p=3 (1,299 dofs), one can observe that the high-order continuity of B-splines achieves a more accurate stress solution at a lower polynomial degree with fewer degrees of freedom, while the reduced C^0 continuity along the p-version cell boundary distinctly influences the stress solution.

Multiplying the total strain energy, which has been obtained by the FlagShyp discretization of the quarter system, by four yields a reference $U_{ex} = 2.305691 \cdot 10^{-3}$ for the complete system. Convergence studies with different mesh sizes indicate that the given total strain energy U_{ex} is correct up to the 6^{th} decimal, so that relative errors in terms of Equation (2.23) being larger than 0.1% can be reliably determined. For the finite cell meshes of the p- and B-spline versions given in Figures 4.2a and 4.3a, respectively, a p-refinement study is performed. Using the overkill result from the Flagshyp discretization as a reference, Figure 4.5a reveals that both the p- and B-spline versions of the finite cell method are able to achieve exponential rates of convergence in strain energy measure. At an error level of 0.1%, the convergence curves of both FCM versions level off due to the impact of the finite value of the penalty parameter α. The jumps in the convergence curve of the B-spline version can be attributed to the influence of the integration error.

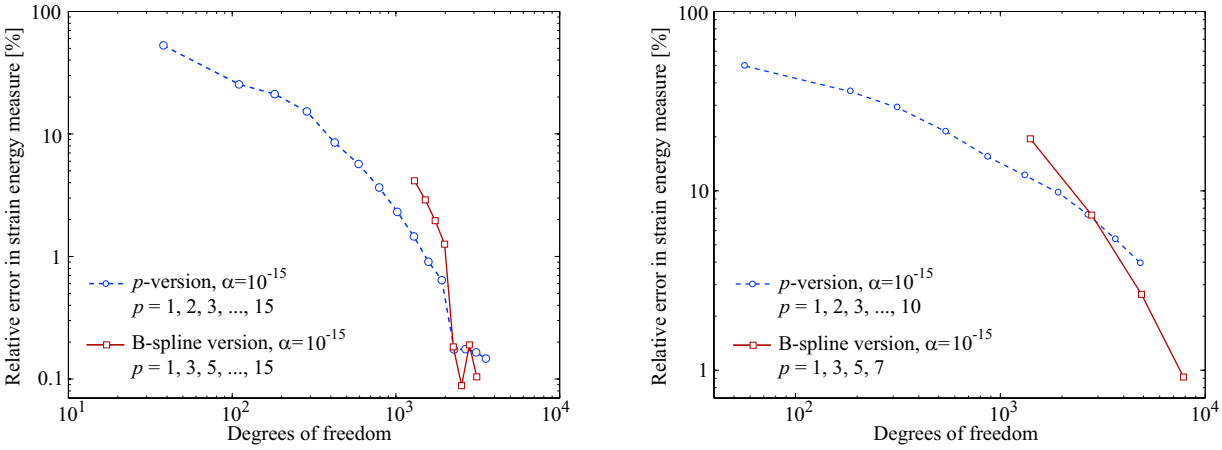
The reason for the different mesh size in the p- and B-spline versions lies in the different behavior of the support of the corresponding basis functions under p-refinement. To illustrate that statement, let us consider the 4×4 p-version mesh of Figure 4.2a. According to Section 2.2.1, the support of the nodal, edge and internal modes in a two-dimensional p-version discretization under p-refinement is invariant, being a maximum of 4×4, 2×2 and 1×1 finite cells, respectively. However, for B-spline basis functions, the maximum support spans $(p+1)\times(p+1)$ finite cells (see Section 2.2.2) and thus rapidly increases with the polynomial degree p considered. A corresponding 4×4 knot span mesh exhibits one basis function with global support at p=3, nine at p=5, and almost all of its basis functions are global at p=15. On the one

hand, such a discretization will lead to an almost fully populated stiffness matrix with full bandwidth. The B-spline version in this form can thus be characterized rather as a spectral method than a finite element scheme. At the same time, it is considerably less accurate than a p-version mesh, since it contains much fewer degrees of freedom. At polynomial degree p=15 for example, 4×4 knot span cells exhibit approx. 750 dofs, while approx. 3,500 dofs are created in a 4×4 p-version mesh. Therefore, a comparison of p-version and B-spline discretizations with the same number of cells seems not appropriate to us and is not presented in the scope of this work. Instead, the number of knot span cells is increased in such a way that p-version and B-spline meshes exhibit approximately the same amount of degrees of freedom and the support of the largest part of the B-spline basis functions is smaller than the complete domain Ω at issue.

4.1.2 Cube with a spherical hole

The 3D analogue of the perforated square plate is a cube with a spherical hole, whose geometry, material and boundary conditions are given in Figure 4.1b. The geometry of the sphere is again described implicitly by the inequality $X^2 + Y^2 + Z^2 \leq r_s^2$, which allows for efficient point location queries in 3D. The complete domain Ω is discretized by a structured FCM mesh of 2×2×2 p-version cells and 7×7×7 knot span cells as shown in Figures 4.6a and 4.7a, respectively. For integration purposes only, the finite cell meshes are complemented by adaptive sub-cells to take into account the discontinuity in α. In Figures 4.6b and 4.7b, only one half of the symmetric sub-cell structures are displayed to uncover their adaptive resolution of the spherical boundary in the center of the cube. Dirichlet boundary conditions can again be imposed strongly, since Dirichlet boundaries coincide with cell faces. For the solution of the geometrically nonlinear system, the displacement load is cut into 3 equally sized increments, each of which requires in both FCM versions only 3 to 4 Newton iterations to converge to a value of the L^2-norm of the residual below 10^{-10}. The modified FCM formulation allows for a penalization with α=10^{-15}.

A first impression of the quality of the geometrically nonlinear FCM solutions can be obtained from the von Mises stress plots given in Figures 4.6c and 4.7c, obtained with polynomial degree p=7. To allow an insight into the 3D stress state, only one eighth of the symmetric system is displayed. Both FCM versions are able to accurately localize the stress concentration around the spherical boundary. Moreover, the physical domain Ω_{phys} does not exhibit stress oscillations. For the computation of a strain energy reference, we again make use of the symmetry in geometry and boundary conditions to reduce the cube to one eighth of the original system. The considered FlagShyp discretization with a mesh conforming to the geometric boundaries consists of 15,300 standard 10-noded quadratic tetrahedrals with 69,862 degrees of freedom. Multiplying the resulting strain energy by 8 yields $U_{ex} = 6.6008376 \cdot 10^{-2}$. Convergence studies with different mesh sizes indicate an accuracy up to the 4^{th} decimal, so that relative errors being larger than 1.0% can be reliably determined. Performing a p-refinement study on the given finite cell discretizations, one can determine the convergence behavior in strain energy with respect to the given reference. The results shown in Figure 4.5b confirm also for the 3D case that exponential rates of convergence can be achieved with both the p-version and the B-spline version of the finite cell method.



(a) 2D plate with circular hole.

(b) 3D cube with spherical hole.

Figure 4.5: Convergence behavior of the 2D and 3D benchmark examples obtained with the modified FCM formulation based on deformation resetting.

4.2 Unfitted Dirichlet constraints and robustness under severe mesh distortion

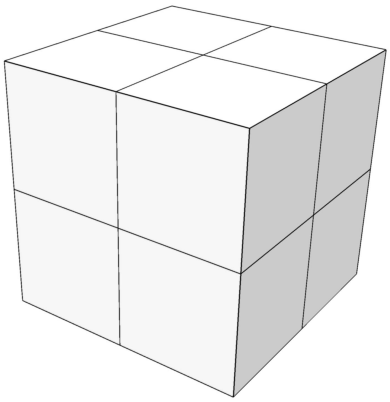
The discretization and linearization of the variational penalty method for the imposition of unfitted Dirichlet boundary conditions in the framework of geometrically nonlinear formulations has been derived in Section 3.4. In the following, its performance is examined in the framework of the geometrically nonlinear finite cell method by the ring example of Figure 2.19. Furthermore, the behavior of the FCM under severe mesh distortion, being fundamental for the representation of very large deformation states, is examined with the example of a twisted beam that also involves weakly imposed Dirichlet boundary conditions.

4.2.1 Ring with weakly enforced boundary conditions

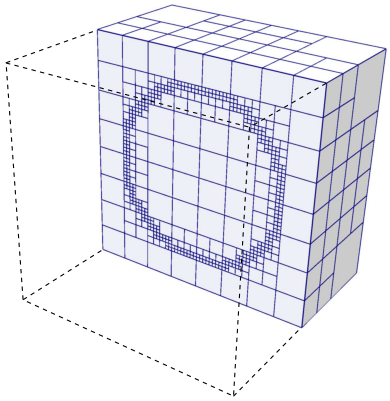
The problem set of Figure 2.19 is maintained with the only exceptions that geometrically nonlinear kinematics are assumed, body forces are omitted and traction over the inner boundary Γ_2 is replaced by a non-zero displacement boundary condition

$$\hat{u}_r = 0.25; \quad \hat{u}_\theta = 0.0 \tag{4.1}$$

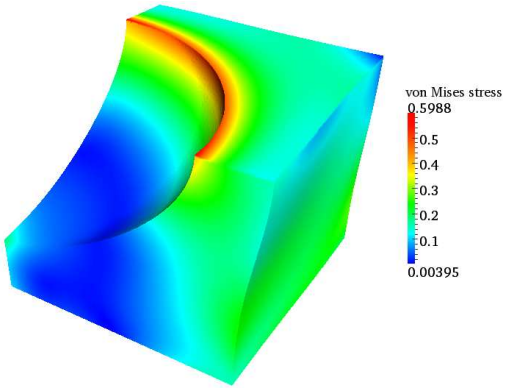
The ring is discretized with the p- and B-spline versions of the finite cell method as shown in Figures 2.20a and 2.20b, where the inner and outer circles are approximated in the sense of Equation (2.26) by an overly large number of 20,212 and 38,612 polygon segments in order to minimize the corresponding error influence. For the solution of the geometrically nonlinear system, the displacements on Γ_2 are cut into 3 equally sized increments, each of which requires 4 to 5 Newton-Raphson iterations to converge to a norm of the residual below 10^{-12}. Figure 4.8a and 4.8b show the regular and symmetric mesh deformation in the physical domain for the p-version and B-spline discretizations, respectively, which confirms that the physical domain is not influenced by the presence or the modified FCM formulation of the surrounding



(a) FCM mesh consisting of 2×2×2 p-version cells.

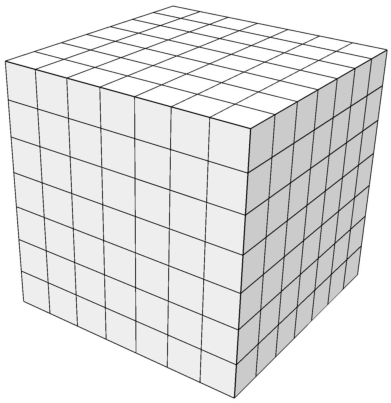


(b) Adaptive integration sub-cells of level m=5.

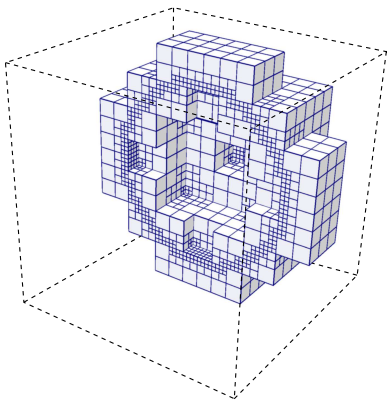


(c) Von Mises stresses on the deformed configuration (p=7).

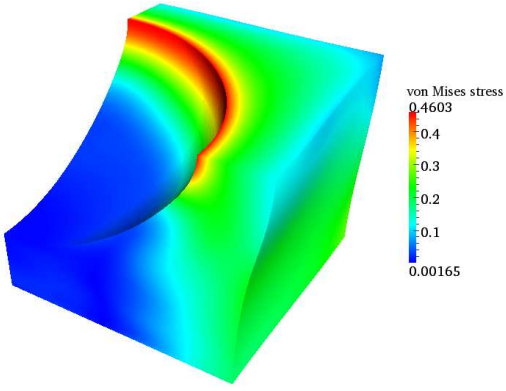
Figure 4.6: 3D cube with spherical hole, computed with the p-version of the FCM. Only one half of the symmetric sub-cell mesh is displayed to uncover its internal structure.



(a) FCM mesh consisting of 7×7×7 knot span cells.

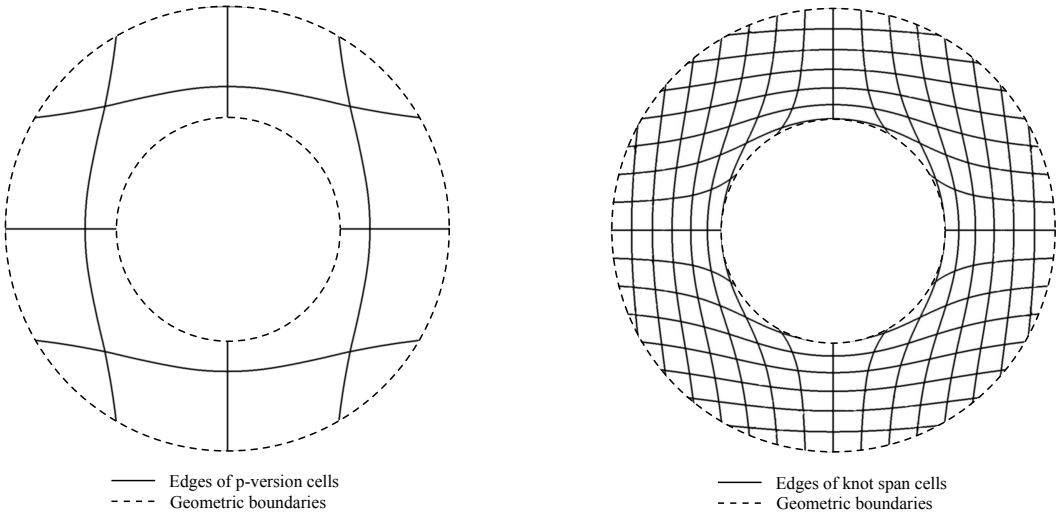


(b) Adaptive integration sub-cells of level m=3.



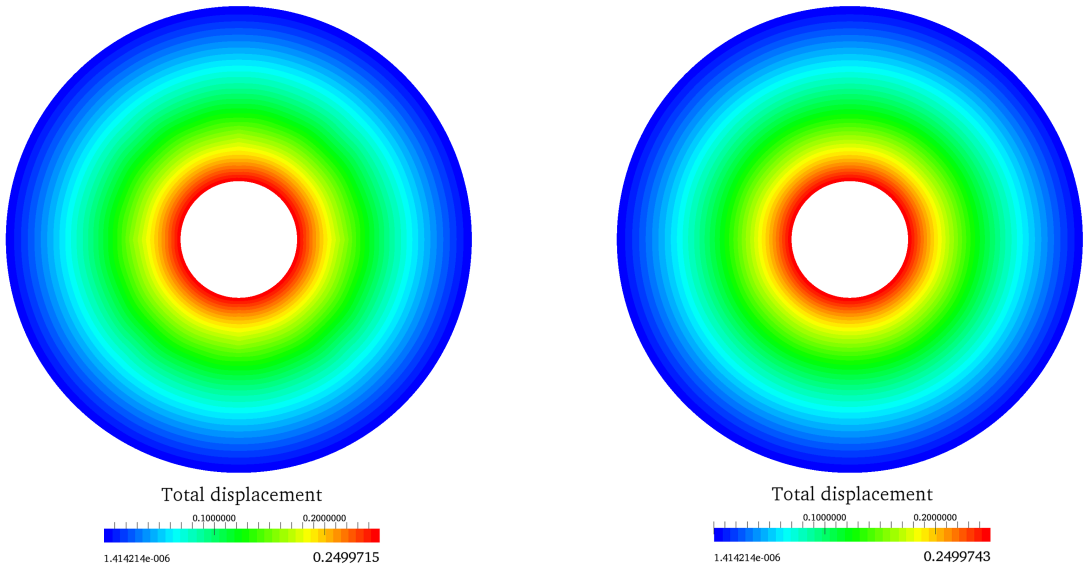
(c) Von Mises stresses on the deformed configuration (p=7).

Figure 4.7: 3D cube with spherical hole, computed with the B-spline version of the FCM. Only one half of the symmetric sub-cell mesh is displayed to uncover its internal structure.



(a) 4x4 p-version cells. **(b)** 16x16 knot span cells.

Figure 4.8: Deformed finite cell meshes of Figure 2.20, plotted over the physical domain, for the ring example at large strains.



(a) p-version. **(b)** B-spline version.

Figure 4.9: Total displacements plotted on the initial configuration, obtained with the above finite cell meshes of polynomial degree p=7.

fictitious domain. Figure 4.9a and 4.9b plot the corresponding total displacement solutions on the initial configuration, which show the expected circular pattern. The maximum values are located at the inner Dirichlet boundary and are accurate with respect to the boundary conditions specified by Equation (4.1) up to the 4^{th} decimal in both FCM versions.

For the computation of the corresponding relative errors in strain energy measure Equation (2.23), U_{ex} is computed by an overkill discretization in FlagShyp, which applies 41,400 linear quadrilaterals conforming to geometric boundaries for one quarter of the symmetric ring.

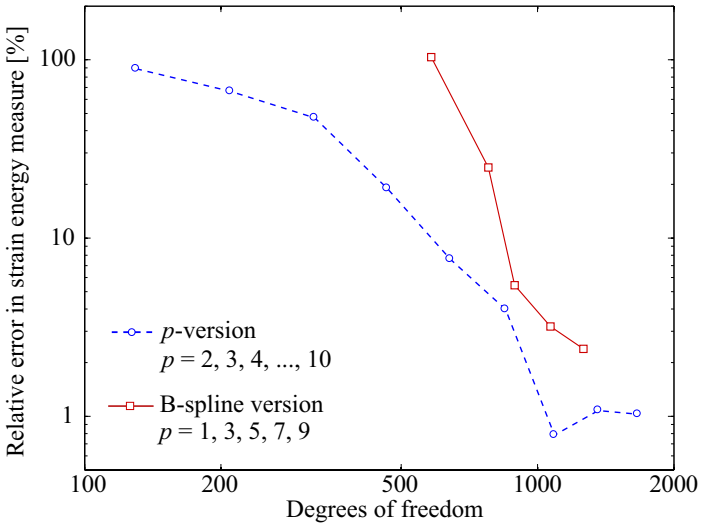


Figure 4.10: Convergence of the relative error in strain energy measure for the ring at large strains.

Multiplying the resulting total strain energy by four yields a reference for the complete system of U_{ex}=0.2855052. Figure 4.10 shows the convergence in strain energy measure, obtained by p-refinement of the p-version and B-spline discretizations, respectively. For each polynomial degree p, parameter β is empirically chosen in both FCM version within the interval $[10^5, 10^7]$, so that the corresponding solution yields a minimum error in strain energy measure. It can be observed that exponential rates of convergence can be achieved for the p- and B-spline versions. However, the convergence curve flattens at an error level of around 1%, due to the influence of the finite penalization parameters α and β. A detailed examination and discussion of possible effects of the penalty parameter β on the condition number of the stiffness matrix and the convergence of the p-version of the finite cell method is provided in [207]. The present example confirms the potential of the penalty method for incorporating unfitted Dirichlet boundary conditions, while maintaining the key advantages of the FCM in terms of simple mesh generation and exponential rates of convergence.

4.2.2 Beam under axial torsion

Previous studies on large deformation analysis with p-version [72, 103, 68] and isogeometric finite elements [56, 77, 127] show that higher-order and higher-continuity shape functions permit increased levels of mesh distortion. In the present context, severe distortion of the structured finite cell mesh may occur as a result of the mapping of cells from the reference to the deformed configuration. In the scope of this work, the ability of the p- and B-spline versions of the FCM to use the robustness of their high-order bases for the representation of very large deformation states is illustrated by the geometrically nonlinear benchmark of a three-dimensional twisted beam, for which a system sketch and material properties are given in Figure 4.11. The dimensions of the beam are chosen according to Lipton et al. [127], where the example is examined for incompressible large deformation elasticity. Torsion is imposed by a Dirichlet boundary constraint, rotating the cross-sectional plane at one end of the beam by angle θ, while the cross-section at the opposite end of the beam is completely fixed.

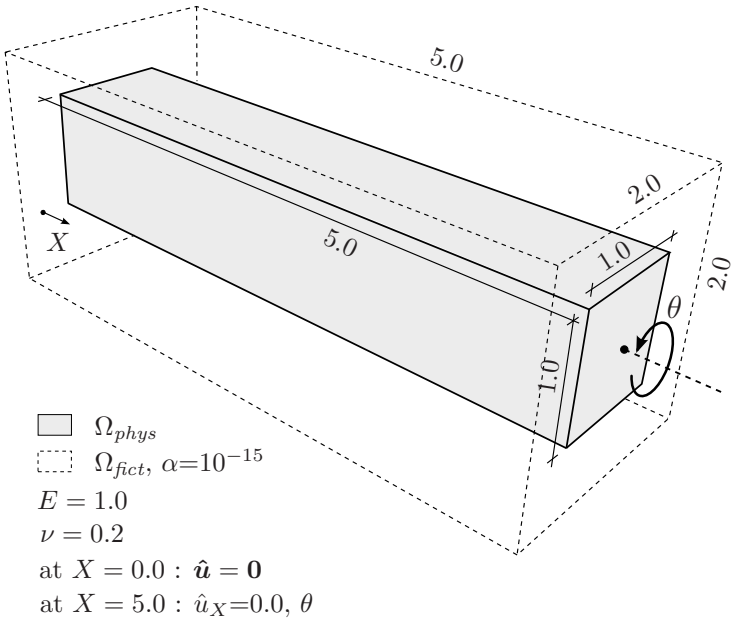
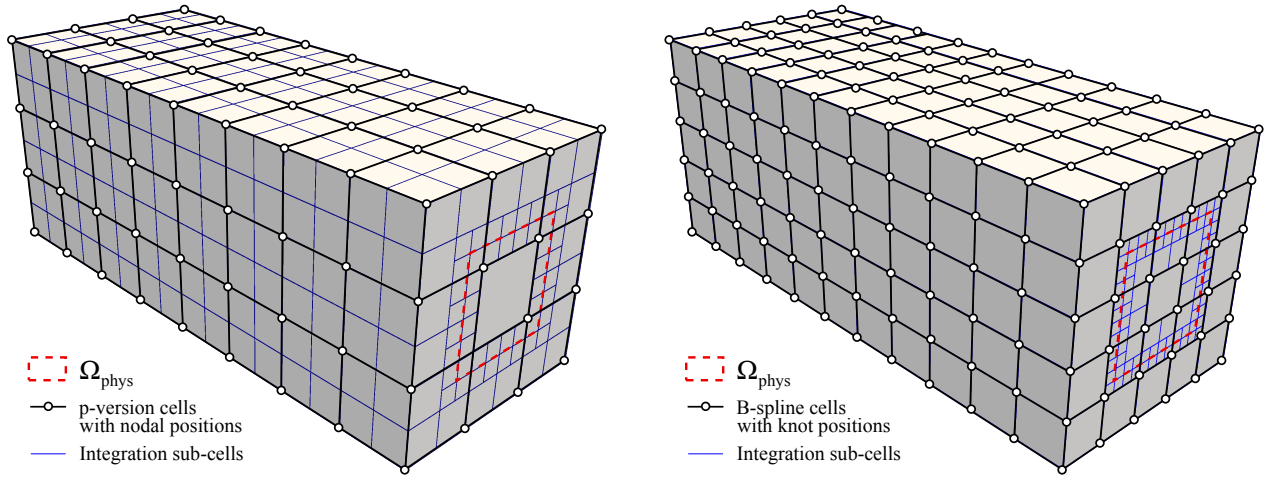


Figure 4.11: Twisted beam in three dimensions.

For testing the solution behavior of the FCM, the beam is embedded in a fictitious domain Ω_{fict} of dimensions 5×2×2, which contains the physical domain Ω_{phys} in its center as illustrated in Figure 4.11 and is penalized by parameter α=10^{-15} in the sense of Figure 2.1. The modified geometrically nonlinear formulation introduced in Section 3.3 is applied. Corresponding discretizations with the p- and B-spline versions of the FCM, consisting of 8×3×3 p-version cells of p=3 with 2,508 degrees of freedom and 11×5×5 knot span cells of p=3 with 2,688 degrees of freedom, are displayed in Figures 4.12a and 4.12b, respectively, and are comparable in terms of polynomial degree and number of degrees of freedom. Due to the cuboidal geometry of the beam, integration of the FCM stiffness matrix contributions can be accomplished exactly with m=2 levels of integration sub-cells, which conform to the discontinuity in α. Unfitted Dirichlet boundary conditions at both ends are imposed weakly via the penalty method with β=10^6 as discussed in Section 3.4, leaving all boundary parts of Ω_{fict} unconstrained.

First, the behavior of the FCM versions for moderately large deformations is examined by imposing a rotation angle of θ=180° (half a rotation around the beam axis). To obtain a qualitative reference, a regular finite element analysis with the p-version of the finite element method is performed. The conforming finite element mesh consists of 20×4×4 high-order elements of polynomial degree p=6 with 43,110 degrees of freedom. In analogy to the FCM, Dirichlet constraints are enforced via the geometrically nonlinear penalty formulation presented in Section 3.4. The resulting von Mises stress and the corresponding deformation pattern are shown in Figure 4.13. Corresponding solutions obtained with the p- and B-spline version of the finite cell method are displayed in Figures 4.14 and 4.15, respectively. The deformed FCM meshes illustrating the behavior of the parts of the finite cells in the physical domain show a regular deformation pattern, in particular the cells cut by the geometric boundary. The corresponding von Mises stresses match well to the reference of Figure 4.13 despite the presence of the fictitious domain. One can identify a slight advantage of the B-spline version, which comes closer in terms of the maximum and minimum stress values and provides a



(a) 72 cubic p-version finite cells with 2,508 degrees of freedom.

(b) 265 cubic knot span finite cells with 2,688 degrees of freedom.

Figure 4.12: Discretization of the complete domain with the p- and B-spline versions of the FCM.

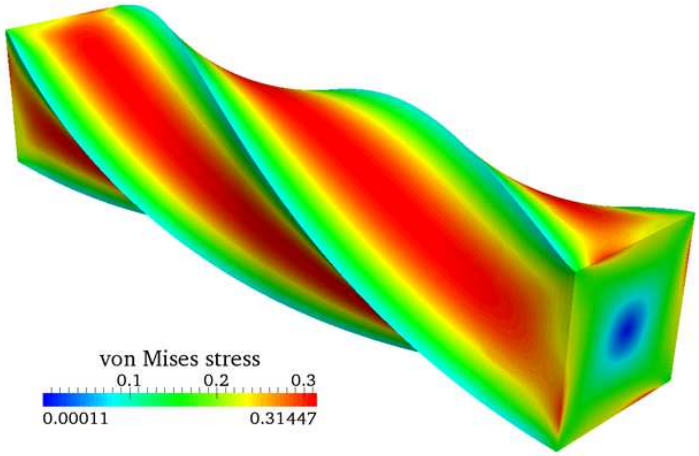
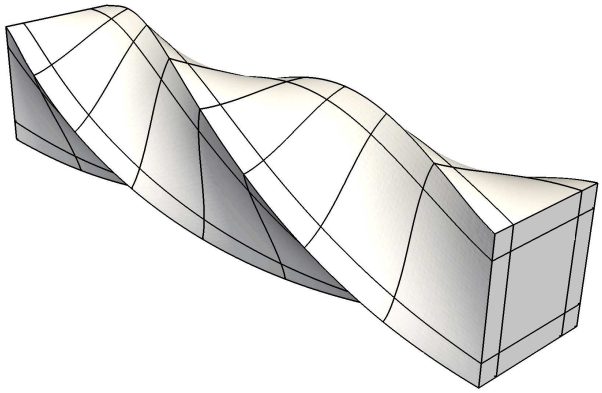


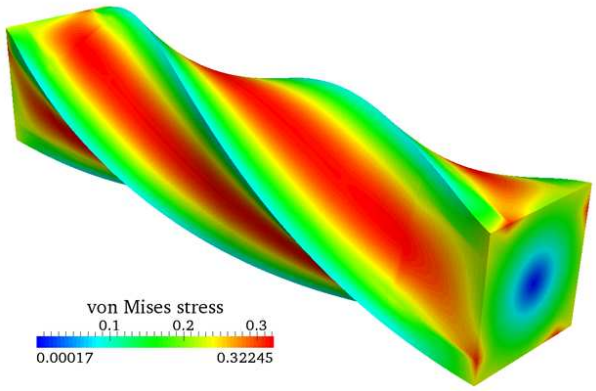
Figure 4.13: Qualitative reference for the beam twisted by θ=180°, obtained with standard FEM.

C^2-continuous stress pattern, while the p-version exhibits jumps across the C^0-continuous cell boundaries. The displacement and stress results show no observable effect from the distortion of the FCM meshes on the displacement and stress accuracy.

Second, the behavior of the finite cell versions under extreme deformations is examined by imposing a rotation angle of θ=1080° (three complete rotations around the beam axis). Corresponding solutions obtained with the p- and B-spline versions of the finite cell method are displayed in Figures 4.16 and 4.17, respectively. Deformed FCM meshes in the physical domain, in particular the finite cells cut by the geometric boundary, still show a regular deformation pattern with the expected helix-like shape despite a considerable distortion of the cells. The corresponding von Mises stresses show a qualitative agreement at first sight, but a closer look reveals that there are some differences in terms of maximum and minimum

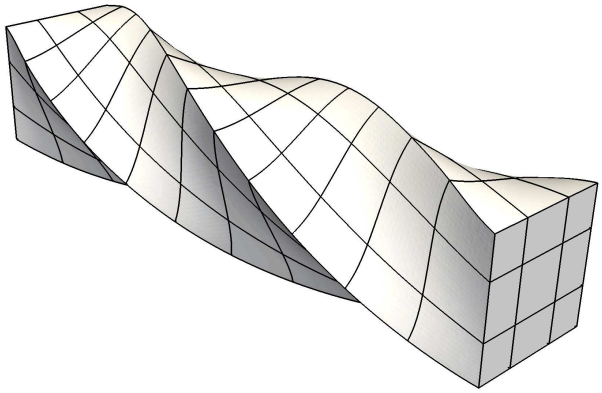


(a) Deformed p-version finite cell mesh in the physical domain.

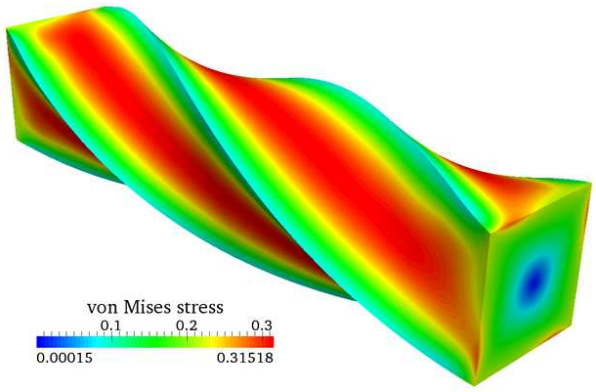


(b) Von Mises stress plotted on the deformed configuration.

Figure 4.14: The beam under a moderately large rotation of θ=180°, computed with the p-version of the FCM. The fictitious domain parts of the mesh and the stress field are not displayed.



(a) Deformed knot span finite cell mesh in the physical domain.

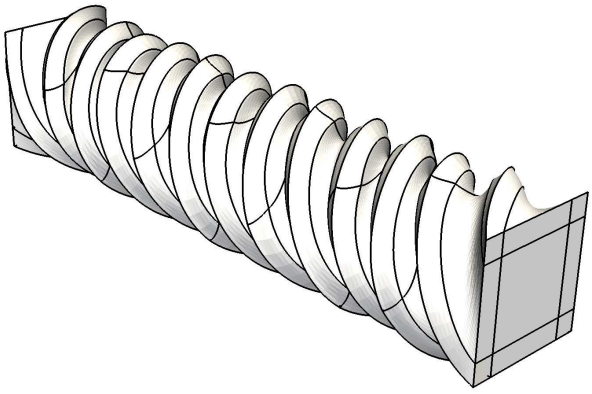


(b) Von Mises stress plotted on the deformed configuration.

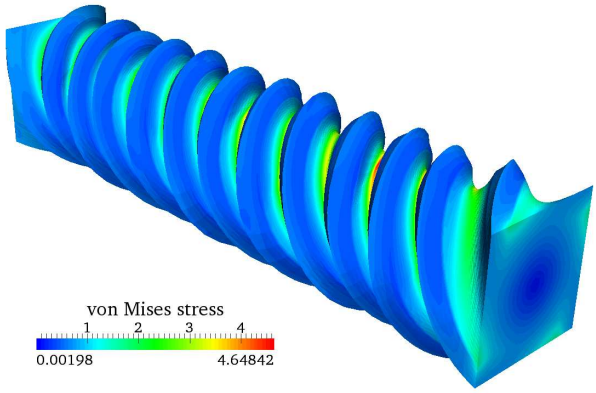
Figure 4.15: The beam under a moderately large rotation of θ=180°, computed with the B-spline version of the FCM. The fictitious domain parts of the mesh and the stress field are not displayed.

values and the location of stress peaks. Due to its higher-order continuity, the B-spline results lead again to smoother deformation and stress patterns than the p-version results. To find the limit deformation states for the present example, representable with the p-version and B-spline discretizations of Figures 4.12a and 4.12b, the rotation angle θ is continuously increased in increments of $\Delta\theta$=9°, until the nonlinear mapping in the physical domain fails due to numerical problems. The p-version of the FCM is able to represent a maximum rotation of the beam of θ=1107°, whereas the knot span finite cell mesh of the B-spline version achieves a slightly larger maximum of θ=1143°.

Figures 4.18 and 4.19 illustrate the distortion of the fictitious domain for moderately and extremely large deformations by plotting the corresponding deformed meshes in Ω_{fict}. From the point of view of the typical FCM solution behavior discussed in Section 2.3, it is beneficial for the quality of the FCM results, if the solution fields in the fictitious domain can

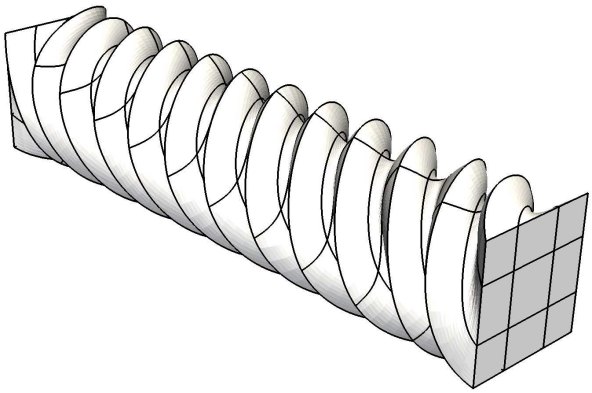


(a) Deformed p-version finite cell mesh in the physical domain.

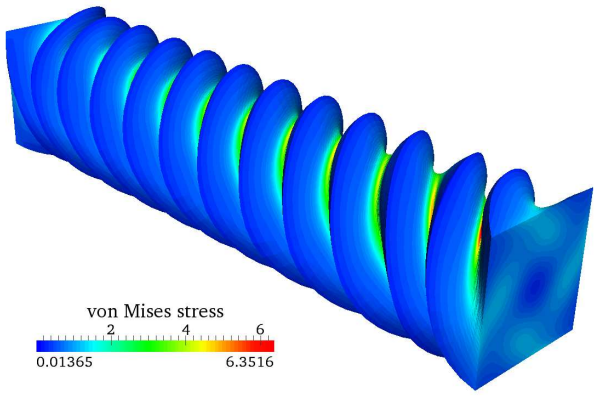


(b) Von Mises stress plotted on the deformed configuration.

Figure 4.16: The beam under an extreme twist of θ=1080°, computed with the p-version of the FCM. The fictitious domain parts of the mesh and the stress field are not displayed.



(a) Deformed knot span finite cell mesh in the physical domain.



(b) Von Mises stress plotted on the deformed configuration.

Figure 4.17: The beam under an extreme twist of θ=1080°, computed with the B-spline version. The fictitious domain parts of the mesh and the stress field are not displayed.

"move" as freely as possible to permit a smooth extension of the physical solution into the fictitious domain. For the moderately large rotation of the beam, this freedom is provided to a comparable extent by both FCM versions (see Figure 4.18). The resulting deformation in the fictitious domain completely lacks physical meaning and an arbitrary penetration of the deformed meshes may occur.

For the very large twist of the beam, the deformation in the fictitious domain of the p-version discretization is considerably larger than the corresponding deformation in the B-spline discretization. Note that Figures 4.19a and 4.19b exhibit different scales, and can be related to each other only by comparison with the deformed physical meshes, which are also displayed. The difference in the size of the deformation indicates that in the presence of extremely large deformations, the C^0-continuous p-version can better accomplish a smooth extension of the solution fields than the B-spline version, where the higher-order continuity of

the basis functions seems to constrain the freedom of solution fields in the fictitious domain. Due to the modified formulation with deformation resetting in Ω_{fict}, the extreme distortion of the mesh in the fictitious domain does not affect the numerical stability of the overall FCM scheme, since the corresponding geometrically linear computations always assume the reference configuration and a nonlinear mapping does not need to be computed.

The results of the beam under extremely large rotation clearly indicate that the finite cell method fully inherits the robustness of the high-order bases. Both FCM versions are able to represent extremely large deformation states despite severe distortion of the physical part of the corresponding mesh. However, the influence of the fictitious domain on the solution fields of the physical part might be increased, in particular in higher-continuity B-spline discretizations.

4.3 FCM as a seamless geometry-through-analysis procedure for voxel models

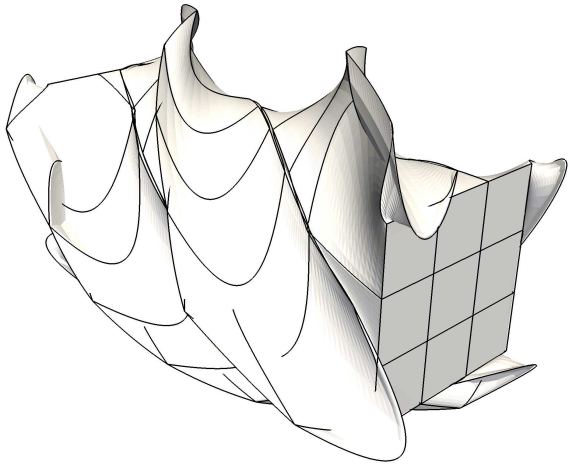
The fundamental advantage of the finite cell method is the very simple and fast grid generation irrespective of the geometric complexity involved. It is based on the disconnection of the high-order grid from the geometry, which is instead represented by the change of parameter α at integration point level. The standard way of representing very complex three-dimensional geometries are volumetric models based on voxel partitioning [2, 3, 38], with large relevance for example in biomedical and material science related applications [85, 168, 216]. Voxel models are usually derived from computed tomography (CT) scans [146], but can also be generated from a B-rep geometry of a CAD model [213]. In the following, the straightforward integration of voxel data for the geometrically nonlinear simulation of a compression test for a complex metal foam structure illustrates the potential of the FCM as a seamless geometry-through-analysis procedure for voxel based geometrical models of very complex geometry.

4.3.1 Efficient generation of adaptive sub-cells from voxels

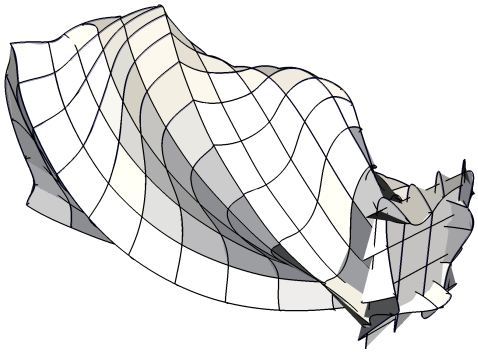
In contrast to the example of a voxel based geometrical model of a bone structure, accommodating inhomogeneous material properties, which has been presented in Section 2.4.3, the focus here lies on solids of very complex geometry with homogeneous material. In the case of one single material, the voxel information required for FCM analysis consists solely of the penalty parameter α. The information necessary at each voxel location can then be reduced to a bit code $b_{vox} \in \{0, 1\}$, which determines whether there is material ($b_{vox} = 1$) or a void ($b_{vox} = 0$). Thus, geometric boundaries of the physical domain Ω_{fict} are represented by the change of b_{vox} from one voxel to the next. The bit encoding significantly speeds up input/output operations and minimizes memory requirements with respect to a direct storage of the floating point number α. The bit code can be simply generated from a CT scan by defining a radiodensity limit, below which b_{vox} is set to zero, while everything above is set to one. The CT based open cell aluminium foam sample[2] shown in Figure 4.20 illustrates this concept. The voxel model has also been transferred to a conventional triangulated surface parameterization in STL format [3] given in Figure 4.21, which conveys a clear image of the geometric boundaries.

[2]Courtesy of IZFP Fraunhofer Institute for Non-Destructive Testing, Saarbrücken, Germany; http://www.izfp.fraunhofer.de

[3]Surface Tesselation Language, also Standard Triangulation Language

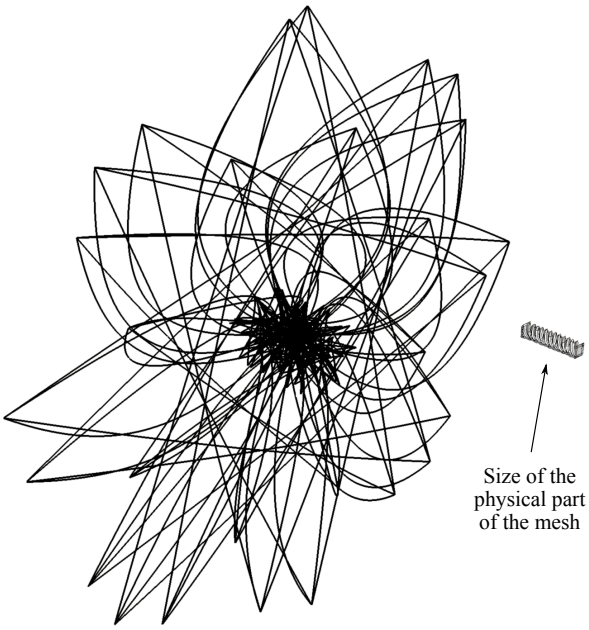


(a) Deformed p-version cells of Figure 4.12a.

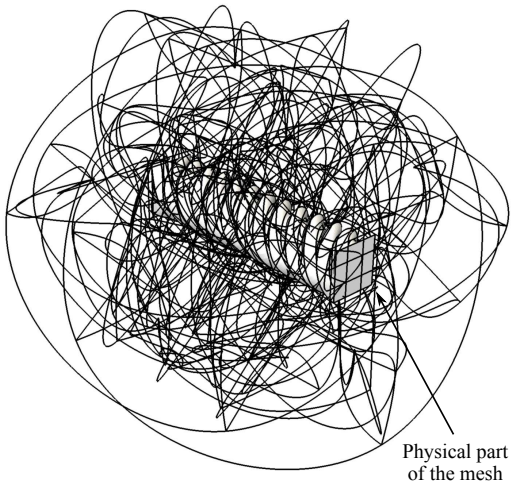


(b) Deformed knot span cells of Figure 4.12b.

Figure 4.18: Deformed finite cell mesh including the fictitious domain for a beam rotation of θ=180°



(a) Deformed p-version cells of Figure 4.12a.



(b) Deformed knot spans of Figure 4.12b.

Figure 4.19: Deformed finite cell mesh including the fictitious domain for a beam rotation of θ=1080°, given by the contours of the cell faces covering the fictitious domain. The deformed physical part of the mesh as shown in Figures 4.16a and 4.17a is also plotted to give an idea of the size of the deformation of the fictitious domain.

With the help of the bit code, an adaptive integration mesh that decomposes finite cells cut by geometric boundaries into sub-cells to accurately take into account the discontinuity in α (see Section 2.2.1), can then be established by the following simple algorithm

1. Traverse all sub-cells of the currently finest level k (start with the finite cells at k=0) and query each Gauss point if it is inside Ω_{phys} or Ω_{fict}.

2. If Gauss points of the same sub-cell are located in different domains (hence, a geometric boundary must be present), split the sub-cell into sub-cells of the next level $k{=}k+1$.

3. Provide all new sub-cells with $(p{+}1)^n$ Gauss points, where n denotes the number of Cartesian directions.

4. Repeat this process, until a sufficient sub-cell depth $k{=}m$ is reached.

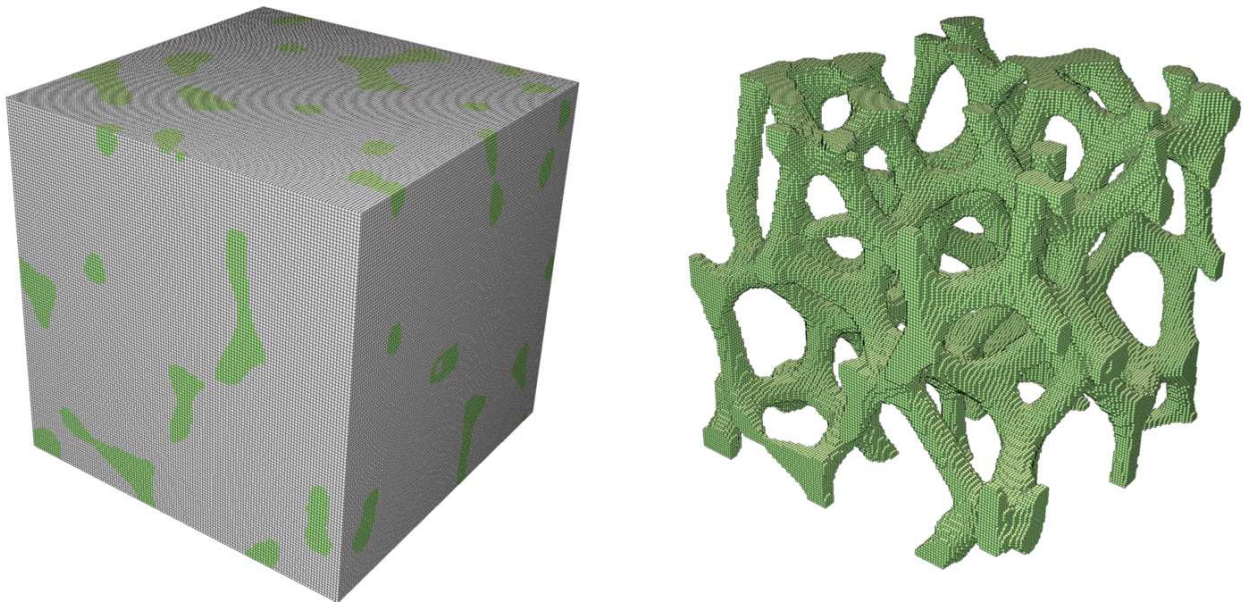
The voxel based bit code provides an ideal geometrical basis for the corresponding location query, which determines for an arbitrary point in space, if it is located in the physical or fictitious domain. Assuming a lexicographical ordering of the voxel data, the integration point position $\{X, Y, Z\}^T$ in the reference configuration can be related to the voxel index k_{vox} with corresponding b_{vox} by

$$k_{vox} = \left\lfloor \frac{(X - X_0)\, n_x}{L_x} \right\rfloor n_y n_z + \left\lfloor \frac{(Y - Y_0)\, n_y}{L_y} \right\rfloor n_z + \left\lfloor \frac{(Z - Z_0)\, n_z}{L_z} \right\rfloor + 1 \tag{4.2}$$

where $\{X_0, Y_0, Z_0\}$, $\{L_x, L_y, L_z\}$ and $\{n_x, n_y, n_z\}$ denote the origin, the length and the number of voxels in each Cartesian direction, and $\lfloor\,\rfloor$ is the floor function.

To illustrate the efficiency of the algorithm, the voxel based geometry of the metal foam shown in Figure 4.20 is discretized with $5{\times}5{\times}5$ p-version finite cells of polynomial degree $p{=}7$ and $m{=}3$ levels of adaptive sub-cells. The fully automated generation of the corresponding discretization shown in Figure 4.22 can be accomplished in only 42 seconds[4]. The main costs result from the loading of voxel information encoded by approximately 10^9 (1 billion) bits, the

[4]On a Intel(R) Core(TM)2 P8800 @ 2.66 GHz



(a) Voxelized sample cube. **(b)** Voxels with $b_{vox} = 1$.

Figure 4.20: Voxel based geometrical model of an aluminium foam sample, derived from a CT scan. For better visibility, the original resolution of 1024^3 voxels is reduced to 128^3.

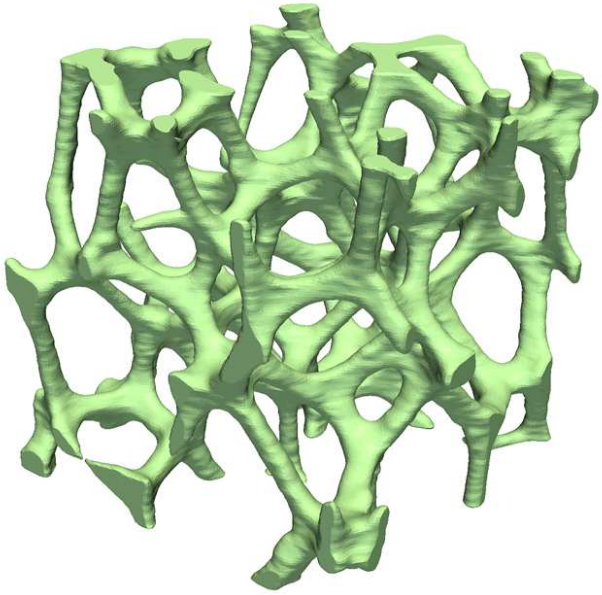


Figure 4.21: STL surface parameterization of the foam structure.

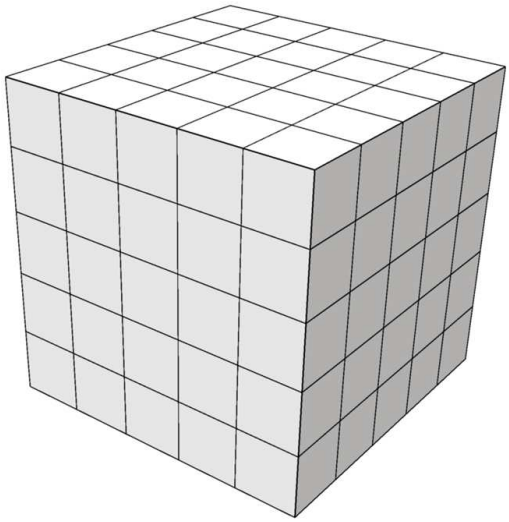
generation of 25,247 adaptive sub-cells and about 680,000 integration point queries according to Equation (4.2). It can be easily observed that the adaptive aggregation of sub-cells around geometric boundaries increases quickly with k.

4.3.2 Nonlinear analysis of a metal foam sample

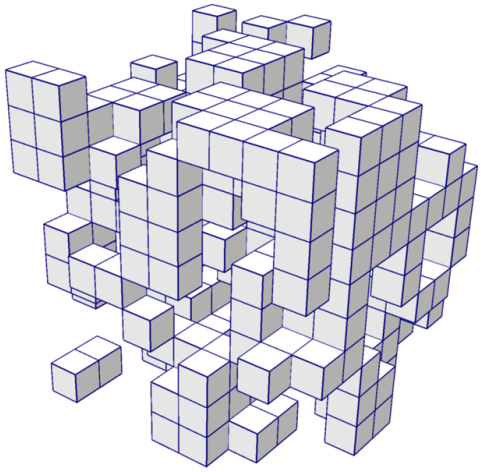
Metal foams provide high stiffness at reduced weights, and are therefore frequently used for lightweight structures in automotive and aerospace applications [15]. The p- and B-spline versions of the FCM are applied to simulate a compression test for an aluminium foam sample of size 20×20×20 mm. Its internal geometry is provided by voxels with a resolution of 1024 in each Cartesian direction, each of which encodes α. Figure 4.20b shows all voxels of material index 1 associated with aluminium in a coarsened resolution of 128^3.

The foam sample is assumed as part of a larger specimen, which is uniformly compressed along the vertical axis. A corresponding RVE[5] model [188, 225] specifies boundary conditions as follows: Displacements normal to the top surface are gradually increased to 1.6mm (8% compressive deformation), modelling the influence of a testing machine, whereas the displacements normal to all other surfaces are fixed due to the bottom support and the influence of the surrounding material of the specimen. Since Dirichlet boundaries coincide with cell faces, Dirichlet constraints are imposed strongly. The aluminium foam is characterized by a Young's modulus of E=70.000 N/mm^2, penalized by α=10^{-12} at all integration points in Ω_{fict}, and a Poisson's ratio of ν=0.35. For the p-version of the FCM, the discretization shown in Figure 4.22 is used, while the B-spline version discretizes the sample with 12×12×12 knot span cells and m=2 levels of adaptive sub-cells. Thus, the smallest sub-cells of the p-version and B-spline discretizations, respectively, contain approx. 25 and 21 voxels in each direction.

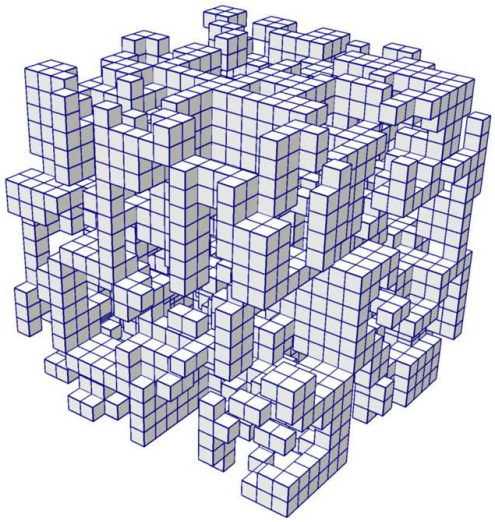
[5]Representative volume element



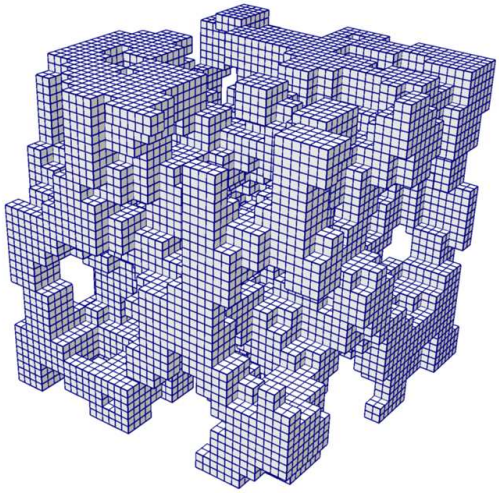
(a) Structured high-order mesh (k=0), consisting of 125 p-version finite cells.



(b) First level k=1 of the adaptive integration structure, consisting of 306 sub-cells.



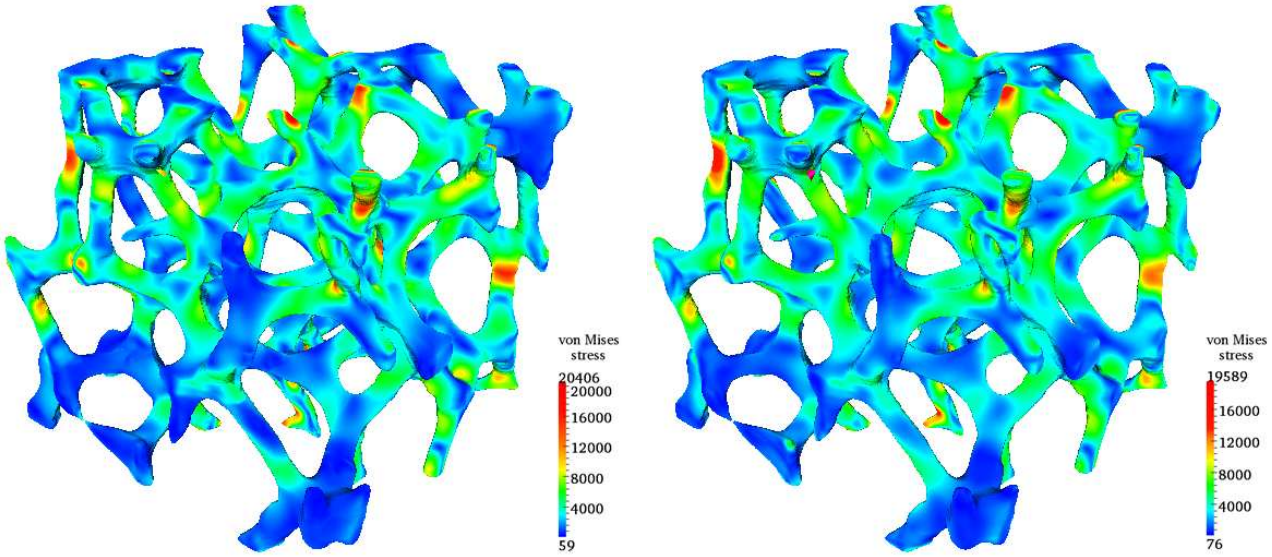
(c) Second level k=2 of the adaptive integration structure, consisting of 2,197 sub-cells.



(d) Third level k=3 of the adaptive integration structure, consisting of 22,744 sub-cells.

Figure 4.22: Discretization of the metal foam sample with the p-version of the FCM with m=3 levels of sub-cells, each with $(p+1)^3$ Gauss points, leading to an adaptive aggregation of integration points around geometric boundaries.

The resulting von Mises stresses computed with the p- and B-spline versions of the FCM at p=7 are shown in Figures 4.23a and 4.23b, respectively. They exhibit accurate localization of stress concentrations at the convex sides of the foam members, which agrees well with engineering experience. Both plots show good accordance in terms of stress patterns, absolute values and locations of stress peaks. Figure 4.24a plots the equivalent force obtained from integration of the normal stresses over the top surface vs. the prescribed displacement of the top surface for different polynomial degrees p, computed with the p-version mesh of Figure 4.22. It can be observed that the increase of p leads to a better reproduction of the nonlinear



(a) p-version of the FCM. **(b)** B-spline version of the FCM.

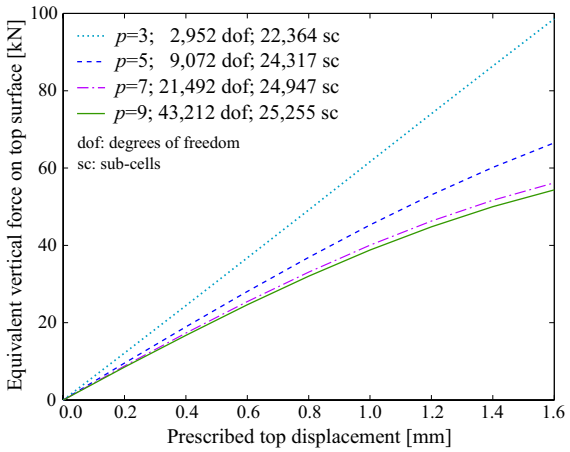
Figure 4.23: Von Mises stress of the metal foam sample plotted on the deformed configuration. The results are obtained from the FCM discretizations of polynomial degree p=7 as described above.

behavior of the foam. Figure 4.24b illustrates the convergence of the equivalent top force under p-refinement, obtained with the p- and B-spline versions. Both schemes converge towards a final load of around 54 kN, where the B-spline discretization tentatively yields a comparable accuracy with fewer degrees of freedom.

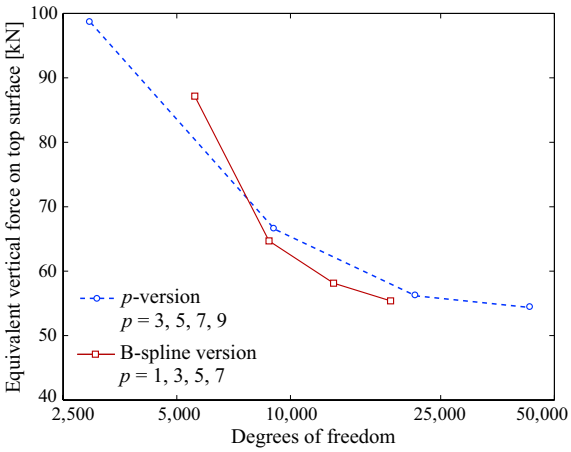
4.4 Software implementation and computational performance

A significant assessment of the computational performance of a numerical scheme cannot be done by considering its accuracy alone, but also has to account for the aspects of hardware and software resources involved as well as the corresponding computational time required. In the following, we discuss theses aspects for the example of the foam analysis.

Our in-house finite cell code, which has been set up from the scratch by the author, implements both FCM versions in the same object-oriented software framework [177]. It has been programmed in C++ and makes extensive use of Sandia's library package Trilinos [203], Intel's math kernel library [111], the optimization options of Intel's C/C++ compiler [112] and shared-memory parallelization with OpenMP [45]. It applies the same routines for linear algebra operations, adaptive integration, the elastic material at finite strains, assembly of the stiffness matrix and the Newton-Raphson iterative procedure. The only difference between the implementation of the p-version and B-spline version consists of the numbering and evaluation of the shape functions. In the context of the finite cell method, the solution of the linearized system of equations, which is strongly ill-conditioned due to the penalization of the fictitious domain, cannot be achieved with iterative solvers. Therefore, the code interfaces the direct solvers Lapack [123] and Pardiso [113]. Since the major cost of FCM results from integration



(a) Force-displacement behavior with increasing p, obtained from the p-version FCM discretization of Figure 4.22.



(b) Convergence of the equivalent vertical force for the p- and B-spline versions of the FCM.

Figure 4.24: Convergence behavior under p-refinement with p- and B-spline versions of the FCM.

over the large number of sub-cells with $(p+1)$ Gauss points in each local direction, a major performance gain is achieved by the shared-memory parallelization of the loop that computes local stiffness matrices for cells and sub-cells with subsequent assembly into the global system matrix. A *parallel for* construct creates a team of n threads to execute the main loop over sub-cells in parallel, where n is the number of threads available. With n=8, we achieved a strong speed up of the loop of around five.

In addition to the comparison of their accuracy, the common software basis also allows for a fair comparison between the p- and B-spline versions of the FCM in terms of required main memory resources and computational time. Using the linear solver Lapack [123], our in-house finite cell code computes the given p-version and B-spline discretizations of the foam example at polynomial degree p=7, which are similar in terms of number of degrees of freedom and accuracy of the top force (see Figure 4.24b). The p-version discretization is characterized by 21,492 dofs defined over 24,947 integration sub-cells with approx. 12.75 million Gauss points, while the B-spline discretization involves 18,411 dofs defined over 37,414 sub-cells with approx. 19.16 million Gauss points. The direct solver requires large main memory resources, in particular due to increased levels of bandwidth and population of the high-order stiffness matrices. From the total amount of 70 gigabytes of available main memory, the solution of the p-version system required a maximum memory usage of about 55%, while the solution of the B-spline system required a maximum of about 90%. Both discretizations converge in three load increments with four to five Newton iterations to a L^2-norm of the residual below 10^{-6}. For the p-version mesh, the total analysis of the foam could be accomplished in about four hours[6], whereas the B-spline mesh took about nine and a half hours.

While the required hardware resources and the computational times of both FCM versions seem to be quite high at first sight, one should keep in mind that the FCM analysis of the foam achieves a seamless integration of the voxel based geometrical model. In comparison to

[6]Using eight threads on two interconnected Intel(R) Xeon(R) W5590 @ 3.33GHz with 70 GB RAM

standard low-order approaches, no preprocessing steps such as segmentation of the voxel data or generation of a conforming mesh are required, and the high-order finite cell method yields high accuracy with a moderate amount of degrees of freedom.

4.5 Comparison of the *p*- and B-spline versions of the finite cell method

The benchmark examples presented in Section 4.1 clearly demonstrate that basis functions of the p-version of the finite element method and high-order B-splines are both well-suited for the application in the framework of the geometrically nonlinear finite cell method. Apart from the general validity of both approaches, a more detailed assessment reveals that each of the two methods has its specific strengths, which are briefly highlighted in the following.

A comparison of the p- and B-spline discretizations of the same polynomial degree p with a comparable amount of degrees of freedom shows that the *p-version of the FCM* has advantages in terms of computational efficiency. Hierarchic shape functions of the p-version are defined over only a few adjacent cells as opposed to B-splines, which are supported by a much larger number of knot spans due to their piecewise definition. This results in a smaller number of cells necessary to achieve the same amount of degrees of freedom and a higher degree of adaptivity of the integration sub-cells, which can be clearly observed for example in the p-version and B-spline discretizations of the ring discussed in Section 4.2.2 (see Figure 2.20). A further consequence is a smaller degree of population and bandwidth of the stiffness matrix in the p-version of the FCM, in particular for high-order computations in 3D. The combination of fewer sub-cells at a comparable resolution of geometric boundaries and smaller degree of population at the same polynomial degree makes the p-version of the FCM computationally more efficient than the B-spline version, as illustrated by the computational times given in Section 4.4 for the foam analysis.

The *B-spline version of the FCM* has advantages in terms of accuracy. Due to the smooth extension of solution fields into the fictitious domain, the finite cell method can make use of the higher-order continuity of the B-spline basis. In contrast, the p-version basis is only C^0-continuous along cell boundaries, and therefore leads to discontinuities in stresses. Consequently, the p-version basis can be considered richer than actually required by the target solution, for which a best approximation in terms of the strain energy is to be found. Therefore, the smooth approximation space of the B-spline version can achieve the same level of accuracy with fewer degrees of freedom than the p-version of the FCM, which "wastes" part of its approximation power to unnecessary solution components. This characteristic feature can be observed in the strain energy results of the benchmark examples presented in Section 4.1, in particular for the cube with a spherical hole, as well as for the foam analysis shown in Section 4.3. It partly compensates the efficiency drawback in comparison to the p-version described above. However, the beam under very large torsion presented in Section 4.2.2 indicated that C^0-continuous p-version cells show some advantages over higher-order continuity B-spline cells in the presence of extremely large deformations, since they permit a greater freedom for solution fields in the fictitious domain.

Chapter 5

Hierarchical *hp-d* adaptivity in the finite cell method

In the previous Sections 2, 3 and 4, we demonstrated that the finite cell concept is an accurate and efficient numerical scheme for linear and geometrically nonlinear problems of solid mechanics, provided that they have a smooth solution. For problems with non-smooth solutions, for example in the presence of material interfaces or singularities, the finite cell method exhibits undesirable properties, such as unphysical stress oscillations around inter-element discontinuities or a decay of convergence to a low algebraic rate, which is typical for high-order methods. In the framework of high-order finite elements [197, 198], different strategies have been developed to improve the quality of stresses and to restore high rates of convergence. The *hp*-adaptive method generates a geometrically graded mesh, where both element size and polynomial degree p become smaller towards the location of a singularity or discontinuity [12, 61, 62, 157, 192, 197]. The *rp*-adaptive method tries to identify the locations of inter-element discontinuities and subsequently changes the high-order discretization in such a way that all discontinuities in the solution coincide with element edges or faces, where they can be represented exactly due to the C^0-continuity of the p-version basis [90, 147].

A key advantage of the finite cell method is its simplicity in terms of mesh generation. Since classical *hp*- and *rp*-methods involve complex meshing procedures, their application in the framework of the finite cell method would neutralize the advantage of simple meshing, thus considerably reducing the attractiveness of the FCM. An alternative approach is the *hp-d*-adaptive method originally introduced by Rank [158], which is based on a local overlay mesh of additional basis functions. Since the only constraint for the overlay mesh is linear independence with respect to the original basis, it can be chosen in such a way that it preserves the simple meshing property of the finite cell method. Combining the concepts of *hp-d*-adaptivity and the FCM, we derive an *hp-d*-adaptive p-version, based on a local overlay of linear hierarchical basis functions, and an *hp-d*-adaptive B-spline version, based on a local overlay of hierarchical subdivision B-splines. The behavior of the finite cell method for non-smooth solutions can be considerably improved, which is demonstrated for interface and singular problems. While the *hp-d*-adaptive p-version of the FCM as presented herein can be characterized as a proof-of-concept, which is intended to illustrate the potential of hierarchical *hp-d* adaptivity beyond this work, the *hp-d*-adaptive B-spline version is an important prerequisite, which will be further developed for efficient and easy-to-handle local refinement of NURBS for adaptive isogeometric and embedded domain analysis in Sections 6 and 7.

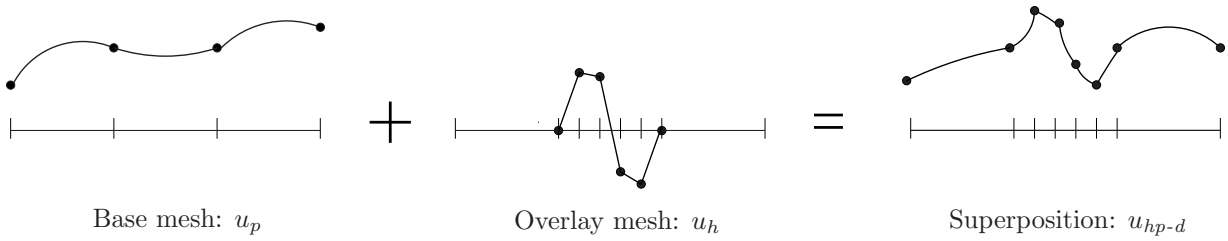


Figure 5.1: Basic idea of the *hp-d* method in one dimension (adapted from [70]).

5.1 The hierarchical *hp-d* method

Figure 5.1 outlines the principle of the *hp-d* method for a simple one-dimensional example. The starting point is a standard high-order finite element approximation u_p on a so-called base mesh, discretizing the whole computational domain. In order to locally improve this solution - for example, in a region where a significant error occurs - a h-version approximation u_h defined on a fine overlay mesh is superimposed to the base mesh. The overall finite element approximation $u_{hp\text{-}d} = u_p + u_h$ results from the superposition of base and overlay approximations. C^0-continuity of $u_{hp\text{-}d}$ is guaranteed by imposing homogeneous Dirichlet boundary conditions on the overlay approximation u_h. Originally proposed by Rank [158, 159] for adaptive domain decomposition (hence the d), the *hp-d* method has been successfully adapted for a range of applications, such as the resolution of local features in non-smooth solutions [159], multi-scale structural analysis [121, 160] and model adaptivity [70, 73]. In the following, we show how the p-version of the FCM can be complemented by an overlay of local linear hierarchical basis functions, which synergetically use the h-adaptivity of the sub-cell based integration scheme. A similar procedure is derived for the B-spline version, which is complemented by an overlay of local high-order hierarchical subdivision B-splines.

5.1.1 Linear overlays for the *p*-version basis

Let us assume that we want to apply the finite cell method to an elastic body, which consists of n different homogeneous material phases as shown in Figure 5.2. Note that in the multi-material case, interfaces between the phases play the same role as geometric boundaries in the fictitious domain case, and therefore have to be resolved as well by adaptive integration sub-cells as described in Section 2.2.2. Following Hansbo [99] and Mergheim and Steinmann

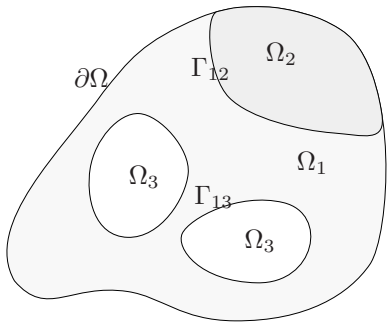
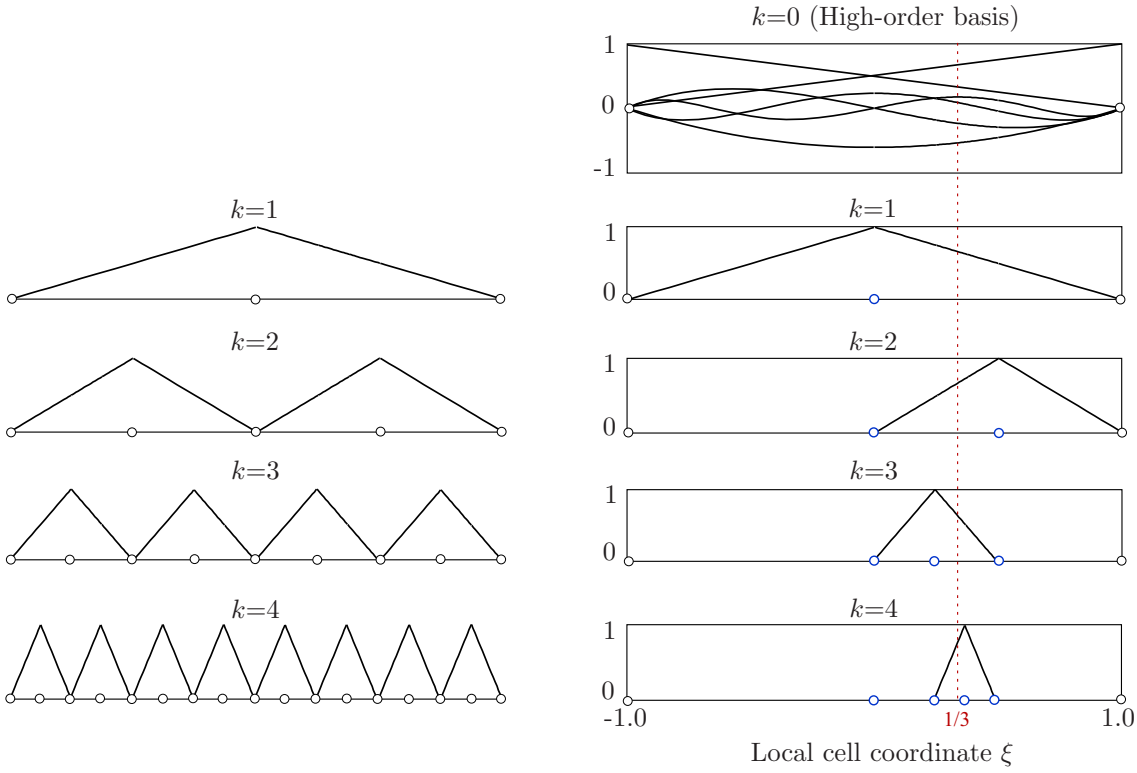


Figure 5.2: Multiphase body of three materials: Domain Ω, bounded by $\partial\Omega$, open sets Ω_1,Ω_2, Ω_3, and internal boundaries Γ_{12}, Γ_{13}



(a) Hierarchical representation of the linear basis according to levels k.

(b) Adaptive *hp-d* overlay to resolve a discontinuity at ξ=1/3 (dotted line).

Figure 5.3: Principle of the *hp-d*-adaptive *p*-version of the finite cell method in one dimension for the example of an inter-element discontinuity.

[135], a material interface problem can be described as follows: A domain Ω with boundary $\partial\Omega$ is divided into n not necessarily compact open sets Ω_k by internal boundaries Γ_{ij}, representing the material interfaces between Ω_i and Ω_j. For any sufficiently regular function f in $\Omega_1 \cup \Omega_2 \cup \ldots \cup \Omega_n$, the jump of f on Γ_{ij} can be defined by $[f] := f_i \,|_{\Gamma_{ij}} - f_j \,|_{\Gamma_{ij}}$, where $f_k = f \,|_{\Omega_k}$ is the restriction of f to Ω_k. Under the assumption of perfect bonding, the following continuity conditions hold on each material interface Γ_{ij} for displacements $\boldsymbol{u}$ and strains $\boldsymbol{\varepsilon}$

$$[\boldsymbol{u}] = \boldsymbol{u}_i - \boldsymbol{u}_j = 0 \tag{5.1}$$

$$[\boldsymbol{\varepsilon}] = \frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}} - \frac{\partial \boldsymbol{u}_j}{\partial \boldsymbol{x}} \neq 0 \tag{5.2}$$

Thus, the displacement solution exhibits a weak discontinuity (kink) along material interfaces, whereas the strains and related stresses exhibit a strong discontinuity (jump). If a high-order cell is cut by a material interface Γ_{ij}, this leads to unphysical oscillations in the solution fields and the convergence rate of the FCM decays to a low algebraic rate [147, 178, 197].

5.1.1.1 Construction of univariate linear hierarchical overlays

To counteract these undesired phenomena, the *hp-d*-adaptive *p*-version of the FCM complements the high-order *p*-version discretization with an adaptive overlay of *h*-version basis

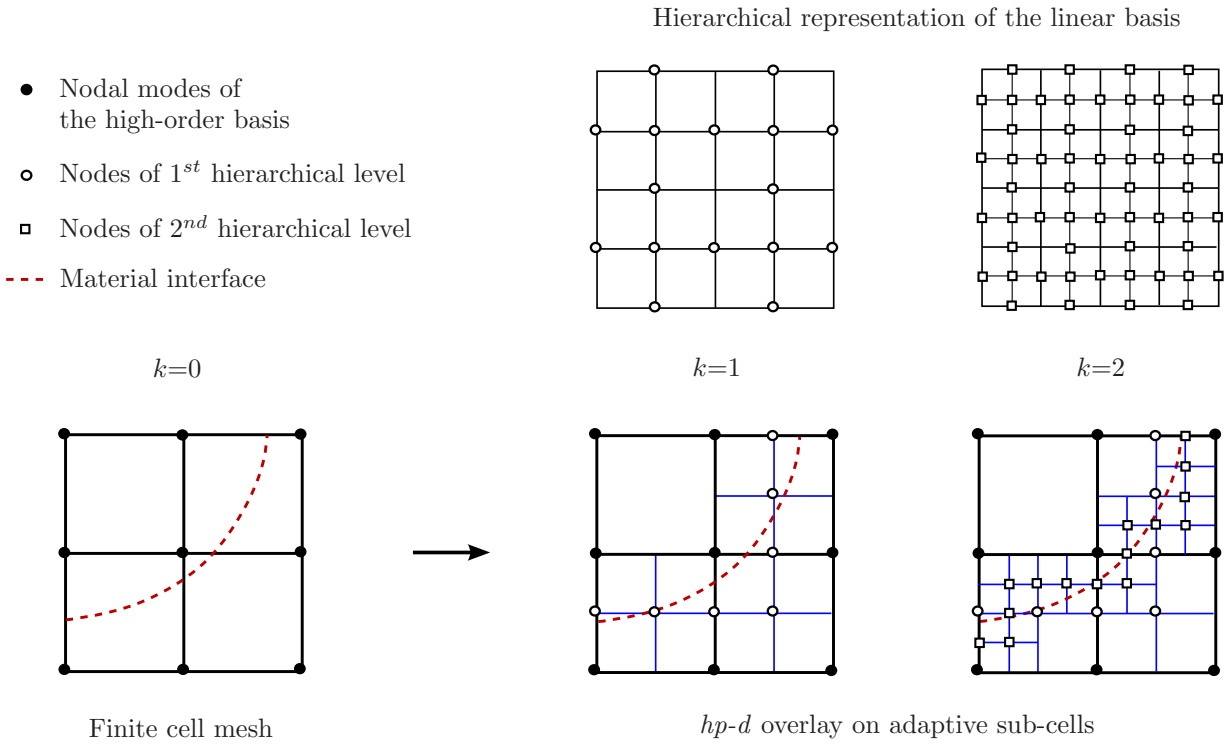


Figure 5.4: Principle of the *hp-d*-adaptive *p*-version of the finite cell method in two dimensions for the example of an inter-element discontinuity.

functions to locally improve the approximation. A fundamental component of the finite cell method is the adaptive integration approach introduced in Section 2.2.2, which is based on a decomposition of cells cut by material interfaces. It already realizes *h*-adaptivity around geometric boundaries in the form of sub-cells, however at this point only for integration purposes. Analogous to the two-dimensional case shown in Figure 2.2, the sub-cell structure in one dimension is organized in the form of a binary tree, which partitions sub-cells of level $k{=}i$ cut by the geometric boundary into two equally spaced sub-cells of level $k{=}i{+}1$. The resulting sub-cell structure shows strong parallels to a hierarchical representation of a linear basis [37, 122, 217] in the sense that both are organized in levels k and carry out a bisection in each level. A linear hierarchical basis is shown in Figure 5.3a.

The key concept for a computationally efficient implementation of the *hp-d*-adaptive *p*-version of the FCM is to synergetically use the sub-cell structure to construct an adaptive overlay from linear hierarchical basis functions [179]. In each level $k > 0$ of the binary tree, those hat functions from level k of the hierarchical basis are added, which have their nodal degrees of freedom between two newly generated sub-cells of the adaptive integration structure. This procedure is illustrated in Figure 5.3b for the one high-order finite cell, where we assume a discontinuity at a local coordinate position of $\xi{=}1/3$. In each level k, the sub-cell containing the geometric boundary is bisected and the corresponding hat function is added to the overlay. Figure 5.3b demonstrates that the resulting basis functions of the overlay achieve a rapid concentration of nodes around the discontinuity.

The *hp-d*-adaptive FCM in this form can be easily implemented into existing finite cell codes by the addition of the hierarchical overlay. Since it operates directly on the existing

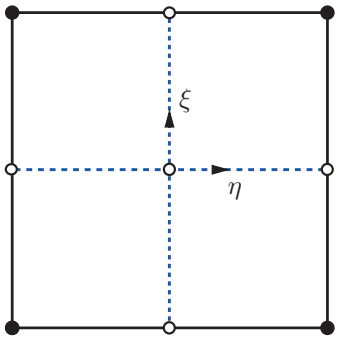


Figure 5.5: Sub-cell partitioning of two-dimensional quadrilateral cell and corresponding potential nodal candidates for the hierarchical *hp-d* overlay (open circles).

sub-cell structure of the adaptive integration scheme, it requires no additional mesh generation and completely retains the easy meshing property of the original FCM [179]. Its hierarchical adaptivity considerably reduces the number of additional degrees of freedom in comparison to global h-refinement of cells or an overlay with equally spaced h-version basis functions.

5.1.1.2 Construction of multivariate linear hierarchical overlays

In two dimensions, the overlay is constructed analogously [179]. As illustrated in Figure 5.4, we start from a number of p-version finite cells cut by a geometric boundary, which are equipped with high-order basis functions up to polynomial degree p and several levels k of adaptive integration sub-cells. Looking at the hierarchical representation of a two-dimensional linear basis in Figure 5.4, we note that its arrangement into levels k corresponds to the organization of the quadtree based integration sub-cells (see also Figure 2.2). In the next step, we successively iterate over each level k of hierarchical sub-cells and add a hat function of the corresponding hierarchical level k to the finite cell basis, if all four sub-cells around a nodal position are contained in the adaptive integration structure. Hence, there are five potential candidates for each of the two-dimensional sub-cells as illustrated in Figure 5.5, from which the outer nodal positions on the four edges are only selected, if the corresponding neighboring sub-cells are also part of the sub-cell structure. We note that the finest level $k{=}m$ of basis functions of the overlay requires a depth of the hierarchical sub-cell structure equal or larger than m. The *hp-d* overlay in this form fully re-uses the h-adaptivity provided by the sub-cell based integration structure and does not require the introduction of additional Gauss points or sub-cells. It is therefore easy to implement, maintains fast mesh generation irrespective of the geometric complexity, and achieves a high degree of local adaptivity around geometric boundaries at a limited amount of additional degrees of freedom.

5.1.2 Subdivision overlays for high-order B-splines

In the B-spline version of the finite cell method, a hierarchical overlay for *hp-d* refinement can be motivated in a natural way by using the subdivision property of B-splines

$$N_p^h(\xi) = 2^{-p} \sum_{j=0}^{p+1} \binom{p+1}{j} N_p^{h/2}(2\xi - j) \tag{5.3}$$

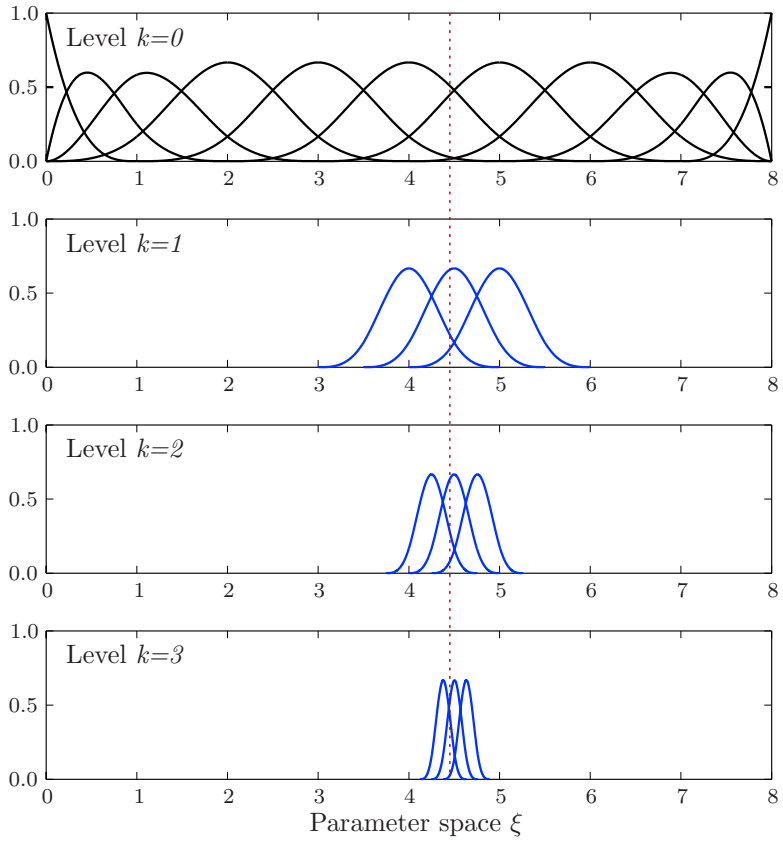


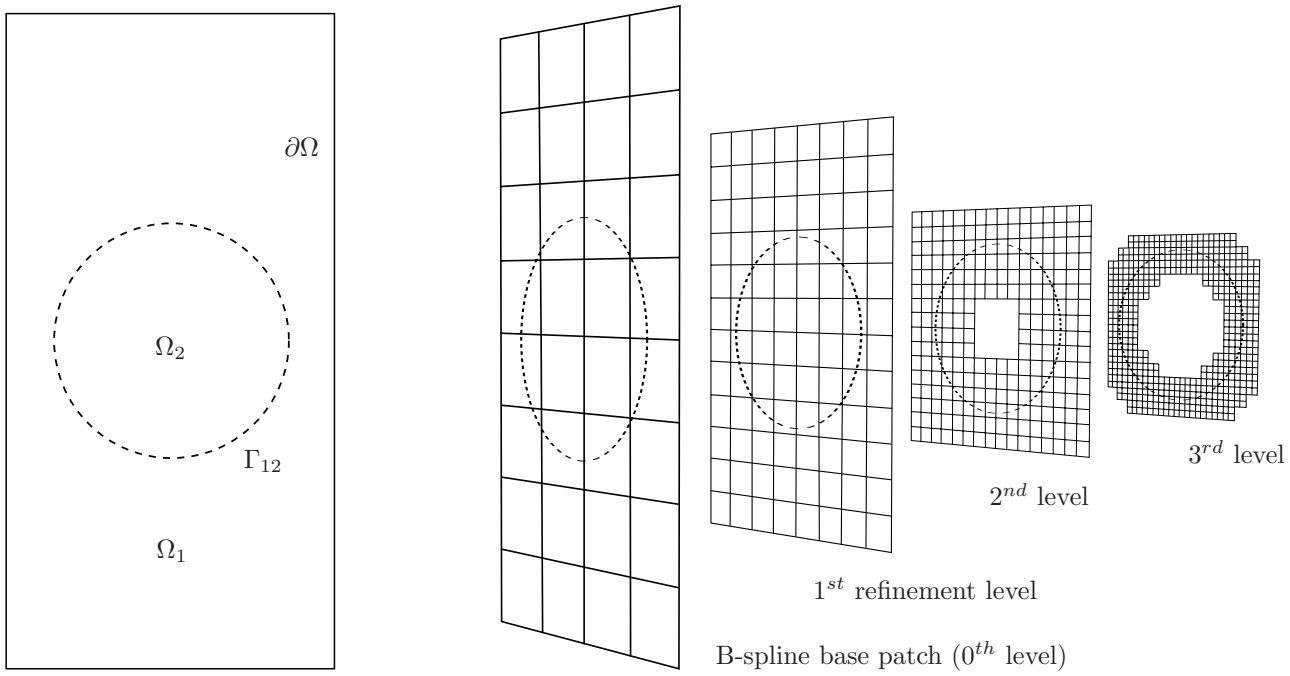
Figure 5.6: Refinement towards the interface: In each level k, three basis functions of contracted support are added central to the knot span of level *k-1*, in which the discontinuity occurs. The location of the discontinuity is assumed at ξ=4.48.

Equation (5.3) also known as the two-scale relation states that a B-spline of degree p and grid width h can be expressed as a linear combination of several B-splines of the same degree, but half the grid width $h/2$ [106, 118, 226]. The two-scale relation is the fundamental principle for the development of efficient and easy-to-handle hierarchical refinement schemes for non-uniform rational B-splines (NURBS) in Section 6. In this context, it is discussed in greater detail in Section 6.2.1 and illustrated in Figures 6.4 and 6.5.

5.1.2.1 Construction of univariate hierarchical subdivision overlays

Based on the two-scale relation Equation (5.3), a hierarchical overlay can be constructed from successively contracted groups of B-splines [178]. The corresponding construction procedure for a material interface problem in one dimension is illustrated in Figure 5.6. We start by selecting the knot span cell from the B-spline base discretization that is cut by the interface. We then consider groups of *three* B-splines of grid width $h/2$. We select the group, which consists of the B-spline centered to the knot span element cut by the interface and its left and right neighbors, and add it to the hierarchical basis as the first hierarchical overlay level k=1. We restart this process by considering the knot spans of the current hierarchical level k, and repeat this process until the maximum level k=m has been reached. Depending on the overlay level k and the initial grid width h, the grid width of each hierarchical level h_k is

$$h_k = 2^{-k}\, h, \quad 1 \leq k \leq m \tag{5.4}$$



(a) System definition **(b)** Levels of hierarchically contracted knot spans

Figure 5.7: Plate with circular inclusion: Example of hierarchical subdivision overlays in 2D.

The resulting grid of the *hp-d* adaptive basis consists of a nested sequence of bisected knot span cells, which keep the regular grid property of the finite cell method. In the sense of Figure 5.6, levels $k<i$ and $k>i$ will be referred to in the following as higher and lower levels, respectively, with respect to the current level $k=i$. The choice of the central B-spline and its two neighbors guarantees that the refined basis functions symmetrically embed the knot span element of the higher level, which contains the interface. It is shown in [178] that a group size of three B-splines per overlay level offers the best compromise between accuracy and increase in degrees of freedom in comparison to a smaller or larger group size. Since the construction of a B-spline of level k centered to a knot span of level $k+1$ requires that it is defined over an even number of contracted knot spans $p+1$, the *hp-d* adaptive B-spline version of the FCM uses B-splines of odd polynomial degrees $p=1,3,5,\ldots$ in the following.

Equation (5.3) restricts the amount of B-splines for each interface and overlay level to a maximum of $p+1$ basis functions. In case that a group of contracted basis functions contains $p+2$ or more functions, the approximation basis is not linearly independent, since one B-spline of higher level can be represented as the sum of B-splines on one of the overlay levels in the sense of the two-scale relation Equation (5.3). On the other hand, B-splines of higher levels cannot be taken out of the approximation basis, if they are not representable by some combination of lower level B-splines by Equation (5.3), since this leads to the loss of completeness of the approximation basis with respect to a first order polynomial.

It is important to note that there are special cases, where the refinement strategy described here still leads to a finite element basis with linear dependencies. For example, consider a knot span discretization of polynomial degree $p=3$, where two interfaces are located in two neighboring knot span elements. With the refinement procedure of Figure 5.6, a number of $p+2=5$ consecutive B-splines of the next refinement level $k+1$ are generated, which can

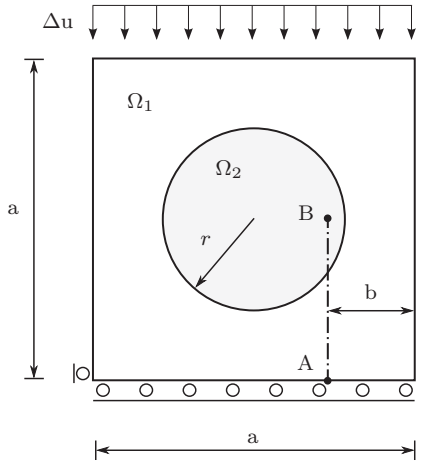


Figure 5.8: Plate with circular inclusion.

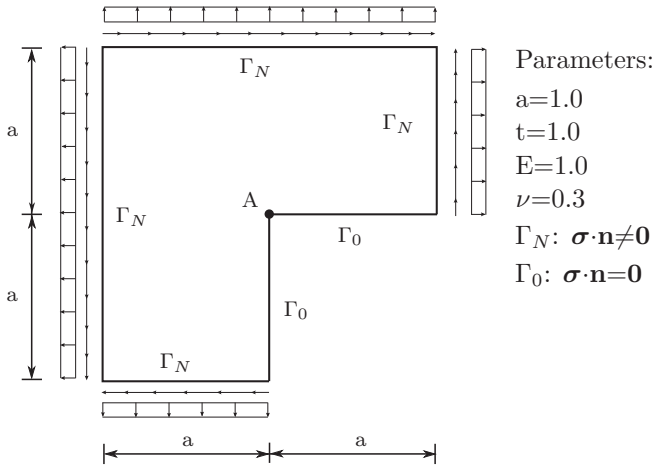


Figure 5.9: L-shaped domain.

represent one of the B-spline basis functions of level k according to the two-scale relation Equation (5.3). Increasing the number of neighboring knot spans with interfaces, this situation can be easily constructed for any polynomial degree p. In this case, one has to go forward to the more advanced refinement strategy described in Section 6, which identifies all linear dependent B-spline basis functions of level k and erases them from the basis.

5.1.2.2 Construction of multivariate hierarchical subdivision overlays

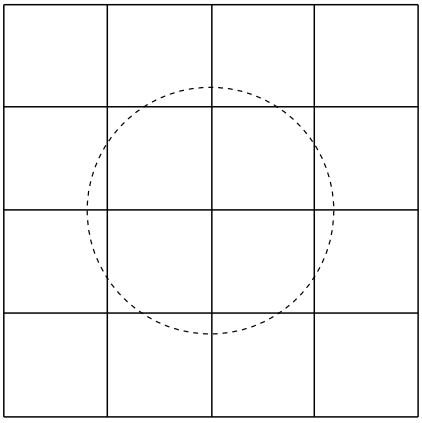
The subdivision based hierarchical *hp-d* strategy for material interface problems in the framework of the B-spline version of the FCM can be generalized straightforward for higher dimensions [178]. Analogous to the one-dimensional definition, interfaces in two or three dimensions, which occur in the form of curves or surfaces, respectively, are refined as follows:

- Traverse all knot span cells of the currently finest level. Check material properties at each integration point.
- If integration points of the same knot span element are associated with different material properties (hence, a material interface must be present), find in each parameter direction the *three* univariate B-splines of the next refinement level, consisting of the contracted B-spline centered to the knot span element as well as its two direct neighbors.
- Use the univariate contracted B-splines to *locally* construct the corresponding tensor product B-splines of half the grid width and add them as additional overlay to the approximation basis.
- Provide all new knot span elements of the currently finest level with integration points and corresponding material parameters.
- Repeat this process, until a sufficient refinement level is reached.

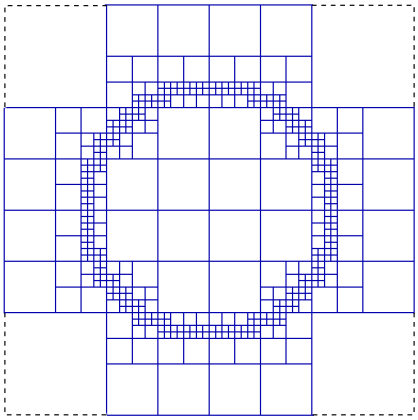
The resulting *hp-d* refinement around a two-dimensional circular interface is illustrated in Figure 5.7. In this form, hierarchical subdivision overlays are easy to implement with a high degree of automation and therefore maintain the simple meshing property of the B-spline version of the finite cell method [178].

5.2 Analysis with the *hp-d* adaptive *p*-version of the finite cell method

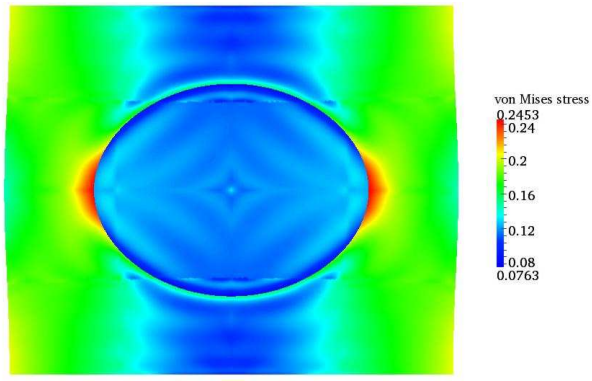
The *hp-d* adaptive *p*-version of the finite cell method is implemented in the framework of our in-house finite cell code, which has been described in more detail in Section 4.4. In the following, we first illustrate the potential of the *hp-d* adaptive *p*-version of the finite cell method to locally improve the approximation of non-smooth solutions. We therefore examine two well-known benchmarks, i.e. the material interface problem of a two-dimensional plate with a circular inclusion [66, 135, 150, 178, 179, 195], which has a discontinuous solution along interfaces, and the problem of the L-shaped domain with a singularity at the re-entrant corner [157, 197, 198]. The results of this section are intended as a proof-of-concept that should motivate further research in this direction beyond the scope of this work.



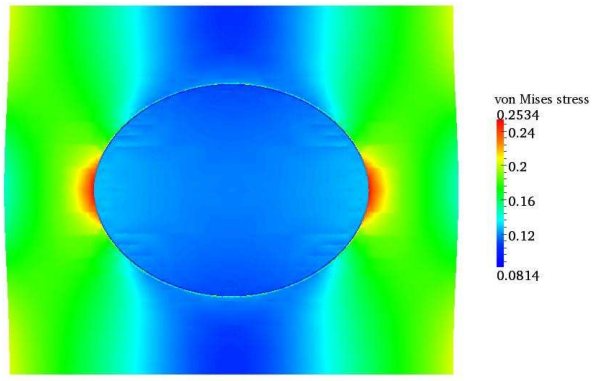
(a) Base mesh of 4×4 high-order *p*-version finite cells.



(b) *h*-adaptive integration grid / hierarchical overlay mesh, depth m=6.



(c) Von Mises stress plotted on the deformed configuration, obtained with the base mesh of p=10 only.



(d) Von Mises stress plotted on the deformed configuration, obtained with the base mesh of p=10 and m=9 overlay levels.

Figure 5.10: Standard *p*-version and *hp-d* adaptive *p*-version of the FCM for the inclusion problem.

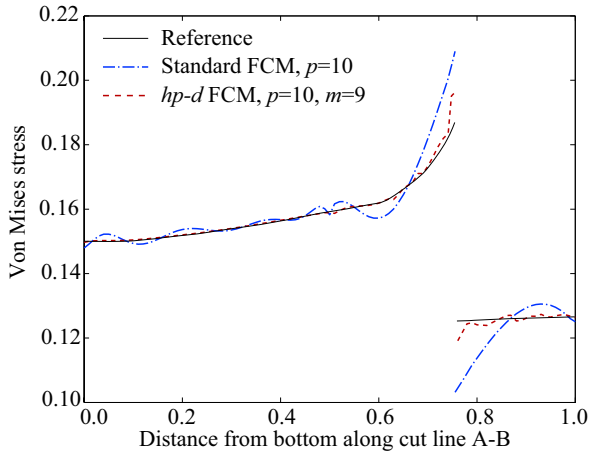


Figure 5.11: Von Mises stress plotted along undeformed cutline A-B (see Figure 5.8).

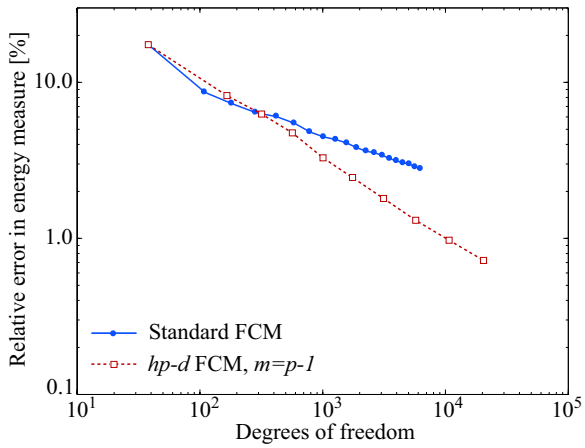


Figure 5.12: Convergence behavior for the plate with circular inclusion.

5.2.1 Plate with a circular inclusion

In Figure 5.8, material and geometric parameters as well as boundary conditions and the location of cutline A-B for a plate in plane stress with a circular inclusion are given. According to Equations (5.1) and (5.2), the solution fields exhibit discontinuities along the material interface, i.e. kinks in displacements and jumps in stresses and strains. In line with Sections 3 and 4, we assume a linear elastic material based on geometrically nonlinear kinematics. A reference is obtained with the open-source nonlinear finite element code FlagShyp [34]. The discretization, which makes use of the symmetry in geometry and boundary conditions to reduce the plate to one quarter of the original system, conforms to the interface and consists of 106,674 standard linear quadrilaterals with 213,681 degrees of freedom. Multiplication by four leads to a reference strain energy of $U_{ex} = 4.330559 \cdot 10^{-3}$ for the complete system. Convergence studies with different mesh sizes indicate that the given U_{ex} is correct up to the 4^{th} decimal digit.

Assuming a geometrically nonlinear formulation as discussed in Section 3.1, the finite cell method can be analogously applied to material interface problems, where the fictitious domain Ω_{phys} is replaced by a second fully valid physical domain Ω_2 with substantial stiffness. The geometry of the interface is not reflected in the high-order mesh, but taken into account in a FCM sense at integration point level via the difference in Young's modulus E. Whether an integration point is located inside or outside of the inclusion is determined implicitly by substituting its global coordinates in the inequality $x^2 + y^2 \leq r^2$. For this class of problems, the α parameter of Equation (2.3) denotes the ratio of the two Young's moduli E_1 and E_2 involved. Figure 5.10a shows the discretization of the complete domain by a base mesh of 4×4 p-version finite cells. Figure 5.10b illustrates the adaptive sub-cell structure, which adaptively resolves the location of the interface for accurate integration of the jump in Young's modulus. The standard p-version of the FCM leads to excessive stress oscillations in cells cut by material interfaces, since the high-order approximation basis alone is not able to accurately represent the discontinuous solution, which is illustrated for polynomial degree p=10 in Figure 5.10c.

The *hp-d* adaptive p-version of the FCM introduced in Section 5.1.1 offers a better approximation of discontinuities along material interfaces. It uses the sub-cell structure of Figure

5.10b to additionally construct overlays of linear hierarchical basis functions. The corresponding stress plot of Figure 5.10d demonstrates that an addition of m=9 overlay levels to the base mesh of polynomial degree p=10 leads to an effective reduction of oscillations and an improved localization of stress concentrations along vertical inclusion boundaries. This observation is further corroborated by the von Mises stresses plotted along cut line A-B in Figure 5.11. Whereas the standard FCM approach yields strong oscillations over the whole domain, the *hp-d* adaptive FCM solution is free of large-scale oscillations, reducing the oscillatory stress behavior to the close vicinity of the interface location. Figure 5.12 compares the convergence in strain energy measure in terms of Equations (2.23) and (2.24) for the standard and the *hp-d* adaptive *p*-version. While *p*-refinement on the base mesh achieves only a low algebraic rate of convergence of around q=0.25, *hp-d* adaptivity complementing the base mesh by m=p-1 overlays of linear hierarchical functions is able to improve the convergence rate to q=0.55.

5.2.2 The L-shaped domain

The solution fields on the L-shaped domain shown in Figure 5.9 exhibit a corner singularity at the intersection point A of the two re-entrant edges. For a detailed discussion of its mathematical characteristics and *hp*-refinement strategies for its efficient numerical solution, see for example [61, 62, 157, 192, 197, 198]. For the specific problem considered in the following, plane strain linear elasticity is assumed (see for example [83, 108, 197]), for which an analytical solution of the strain energy exists in the form

$$U_{ex} = 4.15454423 \, \frac{A_1 a^{2\lambda_1}}{E} \tag{5.5}$$

if the following traction boundary conditions are prescribed along the outer boundaries Γ_N

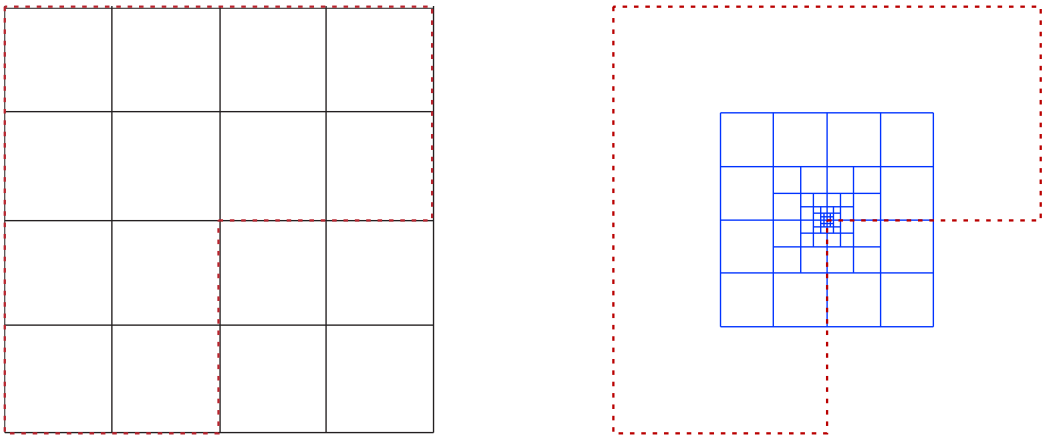
$$\sigma_x = A_1 \lambda_1 r^{\lambda_1 - 1} \left[(2 - Q_1(\lambda_1 + 1)) \cos\left((\lambda_1 - 1)\theta\right) - (\lambda_1 - 1) \cos\left((\lambda_1 - 3)\theta\right) \right] \tag{5.6}$$

$$\sigma_y = A_1 \lambda_1 r^{\lambda_1 - 1} \left[(2 + Q_1(\lambda_1 + 1)) \cos\left((\lambda_1 - 1)\theta\right) + (\lambda_1 - 1) \cos\left((\lambda_1 - 3)\theta\right) \right] \tag{5.7}$$

$$\tau_{xy} = A_1 \lambda_1 r^{\lambda_1 - 1} \left[(\lambda_1 - 1) \sin\left((\lambda_1 - 3)\theta\right) + Q_1(\lambda_1 + 1) \sin\left((\lambda_1 - 1)\theta\right) \right] \tag{5.8}$$

where r and θ denote polar coordinates with origin at the corner singularity point A. The values of the constants involved are the smallest eigenvalue λ_1=0.544483737, the generalized stress intensity factor A_1=1.0 and Q_1=0.543075579. The re-entrant edges Γ_0 are stress-free and constants a and E are geometric and material parameters given for the present case in Figure 5.8. A detailed derivation of Equations (5.5) to (5.8) can be found in [197].

We consider the L-shaped domain problem to illustrate the applicability of *hp-d* adaptive finite cell method in the presence of singular points within the physical domain. The L-shaped domain is discretized by a base mesh of 16 *p*-version finite cells as shown in Figure 5.13a. In a FCM sense, we do not account the L-shape of the domain, but discretize an embedding domain of simpler geometry, where the lower left quadrant of the mesh is penalized by parameter α=10^{-12}. Since the geometric boundary of the physical domain conforms to cell edges, no adaptive integration is required. Therefore, we additionally construct sub-cells around the re-entrant corner, where we construct overlays of linear hierarchical basis functions as described in Section 5.1.1. Figure 5.13b illustrates the resulting sub-cell structure, which realizes an adaptive *h*-refinement towards the singular point A. In each overlay level k, all sub-cells



(a) Base mesh of 16 high-order p-version cells.

(b) Overlay mesh of linear hierarchical functions.

Figure 5.13: Discretization of the L-shaped domain with the p-version and the hp-d adaptive p-version of the FCM. Since the physical domain conforms to the finite cell mesh, no sub-cells are necessary for adaptive integration. The dashed line indicates the geometric boundary.

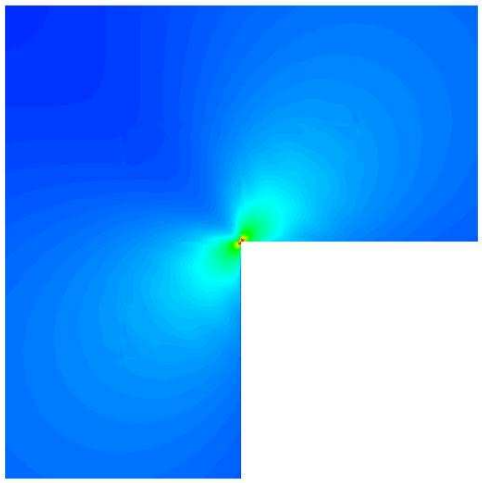


Figure 5.14: Von Mises stress obtained with p=12 and m=11 overlay levels.

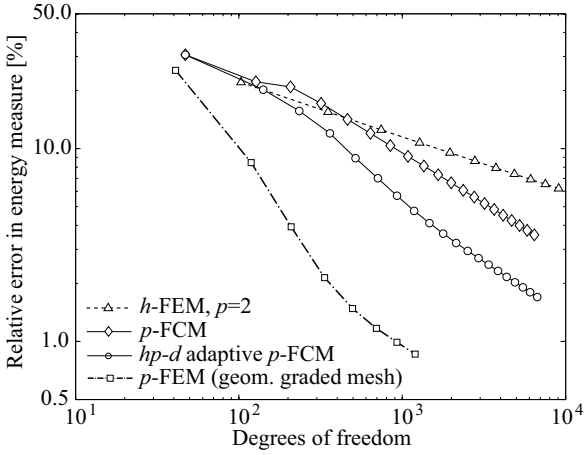


Figure 5.15: Convergence in energy norm for the L-shaped domain (graded mesh results are extracted from [197, 198]).

touching point A are refined by partitioning them into four new sub-cells of the next level k+1. Following Figures 5.4 and 5.5, all those newly created sub-cell nodes are equipped with linear hat functions of level k+1 that are either in the center of the partitioned cell or lie on a boundary that touches other sub-cells of level k+1.

The resulting von Mises stress obtained with polynomial degree p=12 of the base mesh and m=11 adaptive overlay levels is plotted in Figure 5.14, illustrating the considerable localization of stresses in point A. The stress solution further away from the location of the singularity is smooth and can be accurately fitted by the high-order base mesh. We apply a hp-d refinement strategy, which increases the polynomial degree p on the given base mesh and the number of hierarchical overlay levels m=p-1 simultaneously. In Figure 5.14, the resulting

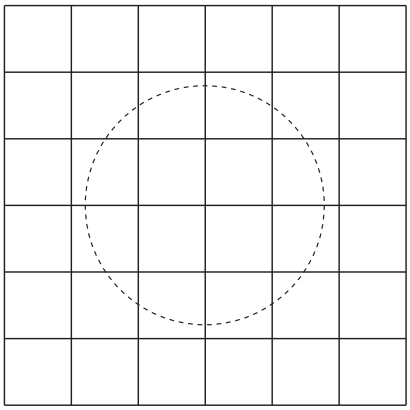
convergence behavior is compared with that obtained by the standard p-version of the finite cell method that uses only p-refinement on the base mesh. In addition, we plot the convergence curves for classical h-refinement with p=2 on a conforming finite element mesh and for classical p-refinement on a conforming geometric mesh, which is adaptively graded towards the location of the singularity, which are reported by Szabó and Babuška in [197]. Whereas h-refined finite elements, bisecting the characteristic element width h in each refinement step, achieve a rate of convergence of around q=0.2775, all other refinement strategies involving the elevation of the polynomial degree p exhibit a very similar final rate between q=0.5413 and q=0.5518. These results are in good accordance with corresponding theoretical estimates [197], predicting q_{est}=0.272 and q_{est}=0.544 for h- and p-refinement, respectively. The performance of the different p-refinement strategies is largely influenced by their pre-asymptotic rates, which can considerably reduce the error level before the final rate tunes in. While the efficiency of the geometrically graded conforming finite element mesh with maximum rates of q=1.36 cannot be reached, the *hp-d* adaptive p-version of the finite cell method still achieves a distinct pre-asymptotic exponential convergence with a maximum rate of q=0.84, which is considerably better than the standard p-version of the FCM with p-refinement only.

5.3 Analysis with the *hp-d* adaptive B-spline version of the finite cell method

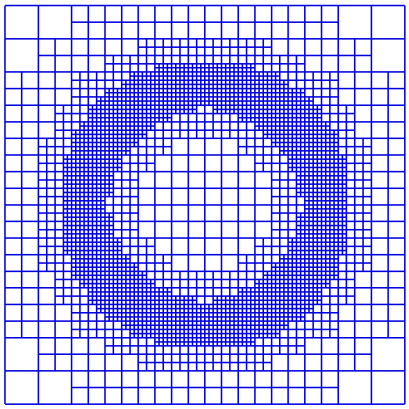
The *hp-d* adaptive B-spline version of the finite cell method is implemented in the framework of nonlinear elasticity in principal directions, provided by our in-house finite cell code (see Section 4.4). In the following, we will illustrate its potential to locally improve the approximation of non-smooth material interface problems, whose solution is discontinuous along interfaces. We therefore examine two benchmark problems, i.e. the two-dimensional plate with a circular inclusion introduced in the previous section and a three-dimensional cube with a spherical inclusion. We also apply the *hp-d* adaptive B-spline version of the finite cell method to simulate a compression test of a foam composite sample. The *hp-d*-adaptive B-spline version in this form constitute the basis for the derivation of efficient and easy-to-handle schemes for local refinement of NURBS in Sections 6 and 7.

5.3.1 Plate with a circular inclusion

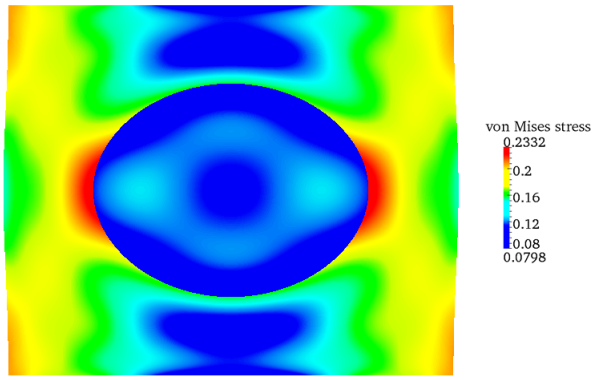
Material parameters, geometry and boundary conditions correspond to the ones shown in Figure 5.8. In analogy with the p-version, the B-spline version of the FCM can be easily adapted for the computation of material interface problems as described in Section 5.2.1. An axis-aligned grid of knot span cells discretizes the complete domain irrespective of the interface geometry, which is taken into account in a FCM sense at integration point level via the difference in Young's modulus E. Figure 5.16 illustrates the beneficial effect of *hp-d* adaptivity, which are placed adaptively around the location of interfaces as described in Section 5.1.2. It compares von Mises stresses for an example discretization of 6×6 knot span elements of polynomial degree p=5 with no subdivision overlays (see Figure 16a) to a corresponding discretization with m=4 hierarchical overlay levels, which resolve the circular interface (see Figure 16b). The solution of the standard B-spline version shown in Figure 5.16c exhibits large oscillations throughout the domain, provoked by the strong discontinuity in stresses



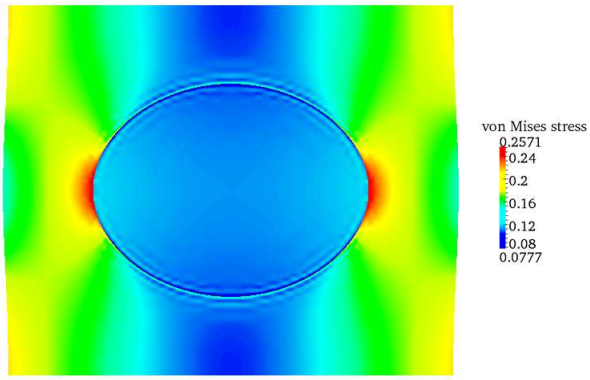
(a) Base mesh of 6×6 knot span finite cells of polynomial degree p=5.



(b) Knot spans of m=4 hierarchical overlays with subdivision B-splines of p=5.



(c) Von Mises stress plotted on the deformed configuration, obtained from the high-order base mesh only.



(d) Von Mises stress plotted on the deformed configuration, obtained from the base mesh with m=4 overlay levels.

Figure 5.16: Knot span discretizations without and with *hp-d* adaptivity for the inclusion problem.

that cannot be approximated by C^{p-1}-continuous basis functions. The *hp-d* adaptive B-spline version leads to a solution that is able to localize the typical stress concentrations along vertical inclusion boundaries, while stresses away from the interface are free of oscillations (see Figure 5.16d). We observe that already a moderate number m=4 of hierarchical refinement levels considerably improves the stress solution with respect to the standard FCM discretization. This behavior is further examined in Figure 5.17, where von Mises stresses obtained along cut line A-B with a distance of b=0.5 from the right boundary of the plate (see Figure 5.8) are compared to the reference solution obtained with standard finite elements in Section 5.2.1. It can be observed that *hp-d* adaptivity around the interface confines the stress oscillations to the close vicinity of the interface, whereas stresses further away are free of oscillations and fit the reference solution well.

A convergence study considering the total strain energy examines the effects of *h*-, *p*- and *hp-d* adaptive strategies. Their effectiveness is again assessed on the basis of the rate of convergence in strain energy measure according to Equations (2.23) and (2.24), where the

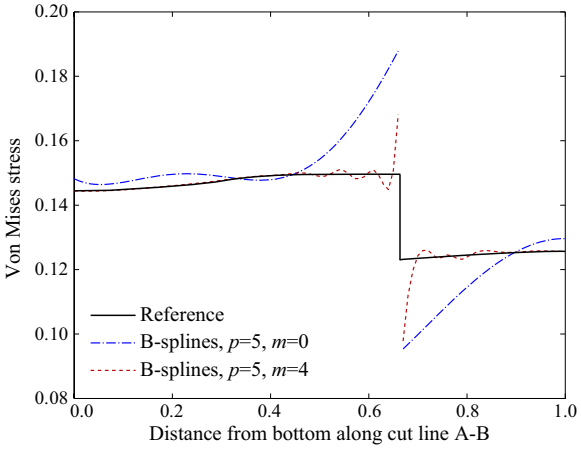


Figure 5.17: Von Mises stress along the undeformed cut line A-B in a distance of b=0.5 from the left plate boundary (see Figure 5.8).

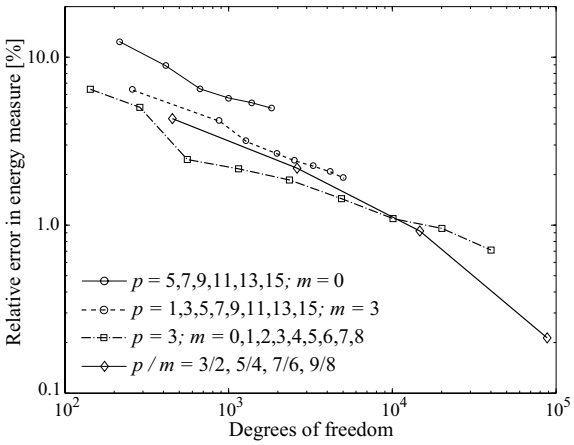


Figure 5.18: Convergence behavior for the circular inclusion problem, obtained with different h-, p- and hp-d adaptive strategies.

FlagShyp reference given in Section 5.2.1 serves as U_{ex}. Figure 5.18 displays the resulting convergence behavior. First, global p-refinement without hp-d adaptivity is examined by increasing the polynomial degree from p=5 to 15 on an mesh of $(p+1)\times(p+1)$ knot span cells. Due to the excessive oscillations in stresses, provoked by the discontinuities at the material interface, the convergence in strain energy measure only achieves a low algebraic rate of $q \approx 0.33$ at a comparably high error level. Second, p-refinement on the same grid is supplemented by a constant number m=3 overlay levels of hierarchical subdivision B-splines around the interface. While the rate of convergence only slightly improves to $q \approx 0.41$, the overall error level is considerably reduced. Third, the effect of local h-refinement with hierarchical subdivision B-splines is examined, which starts from a cubic B-spline discretization of 6×6 knot span elements. The polynomial degree is held constant at p=3 and one overlay level is successively added around the interface up to a maximum number of m=8 levels, which achieves a rate of convergence of $q \approx 0.34$. Finally, a combination of p-refinement and successive increase of the number hierarchical overlay levels is tested. The base mesh consists again of $(p+1)\times(p+1)$ knot span elements, for which p-refinement is performed. In addition, m=*p-1* hierarchical refinement levels are successively added around the interface. This refinement strategy results in a maximum rate of convergence of $q \approx 0.82$. We observe that the standard B-spline version of the FCM leads to low rates of convergence in strain energy measure, which can be considerably improved by adaptive overlays of subdivision B-splines around material interfaces.

5.3.2 Cube with a spherical inclusion

The generalization of the two-dimensional plate with circular inclusion to three dimensions leads to the problem of a cube with a spherical inclusion. A system sketch together with geometry, material and boundary conditions are given in Figure 5.19. A corresponding reference is obtained with the standard finite element code Flagshyp [34]. The corresponding finite element mesh, which conforms to the material interface, discretizes only one eighth of the symmetric system with 24,960 standard quadratic tetrahedral elements and 107,181 degrees

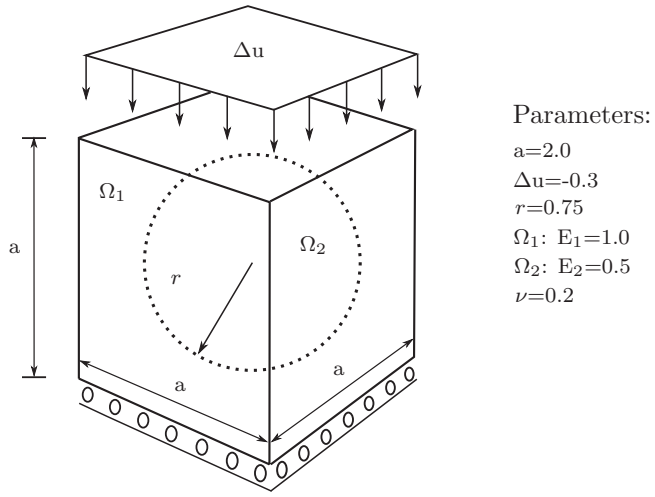
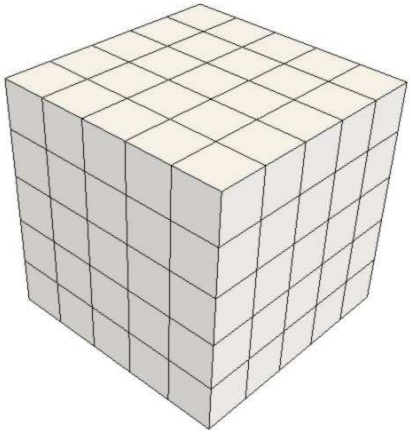


Figure 5.19: The spherical inclusion problem: Parameters are width/height a; radius r; Young's moduli E_1 and E_2; Poisson's ratio ν; prescribed displacements Δu

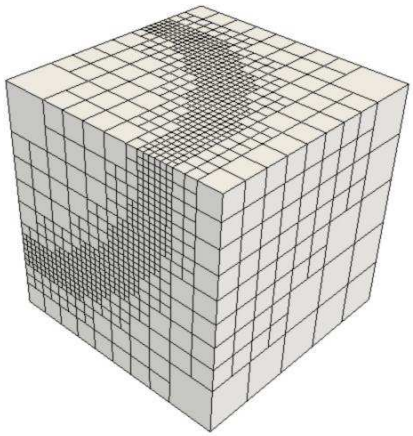
of freedom. It yields a reference strain energy for the reduced system of $U_{ex} = 1.14299 \cdot 10^{-2}$. Convergence studies with different mesh sizes indicate that the given U_{ex} is correct up to the 4^{th} decimal, so that relative errors in strain energy measure, which are larger than 1%, can be reliably determined.

In order to be able to compute higher resolutions with the computational power provided by a standard workstation, computations with the *hp-d* adaptive B-spline version of the FCM are also performed for the symmetrically reduced one eighth system. Figure 5.20 illustrates the effect of hierarchical overlays of subdivision B-splines in three dimensions. It compares von Mises stresses for a base mesh of 5×5×5 cubic knot span cells of polynomial degree p=3 (see Figure 5.20a) to a corresponding discretization with m=3 adaptive hierarchical overlay levels around the interface (see Figure 5.20b). In analogy to the two-dimensional cases, the solution of the base mesh in Figure 5.10c exhibits large oscillations throughout the domain, whereas the solution obtained with hierarchical overlays in Figure 5.10d confines oscillatory stresses to the close vicinity of the interface, while areas further away are free of oscillations. The solution with the *hp-d* adaptive B-spline version is furthermore able to accurately localize the typical stress concentration phenomenon along the vertical edge of the interface.

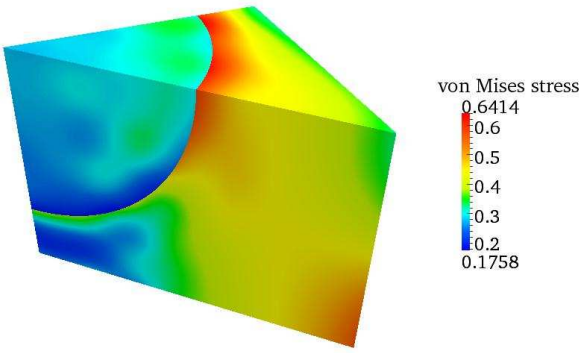
Energy convergence obtained with different adaptive strategies is examined in Figure 5.21. First, global p-refinement without local refinement is examined by increasing the polynomial degree from p=3 to 9 on a base mesh of $p \times p \times p$ knot span cells. Due to the impact of oscillations, convergence in strain energy measure only achieves a rate of $q \approx 0.22$. Second, global p-refinement on the same grid is supplemented by a constant number of m=2 hierarchical overlay levels around the interface, which results in a reduced level of error at a slightly improved rate of convergence $q \approx 0.30$. Third, the effect of local h-refinement with hierarchical subdivision B-splines is examined. On a base mesh of $5 \times 5 \times 5$ knot span elements of constant polynomial degree p=3, we add one hierarchical level of subdivision B-splines per refinement step up to a maximum of m=3, which results in a rate of convergence of $q \approx 0.31$. The *hp-d* strategy proposed in the previous section is not examined here, since several hierarchical overlay levels at a higher polynomial degree p in three dimensions imply an excessive increase in the bandwidth of the stiffness matrix, so that its solution requirements quickly exceed the



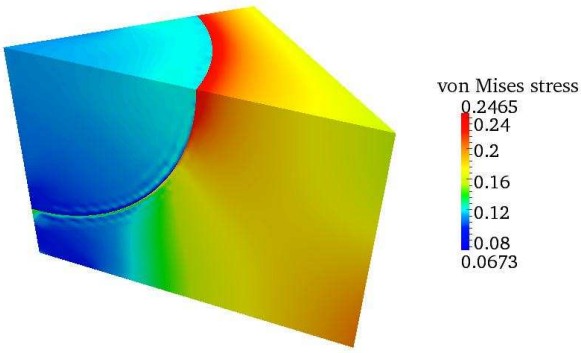
(a) Cubic base mesh, discretizing one eighth of the system.



(b) Adaptive mesh; three hierarchical overlays with cubic subdivision B-splines.



(c) Von Mises stress on the deformed configuration, obtained with the base mesh only.



(d) Von Mises stress on the deformed configuration, obtained with the *hp-d* adaptive mesh.

Figure 5.20: Knot span finite cell discretizations without and with *hp-d* adaptivity.

hardware resources of standard workstations. Nonetheless, the convergence results illustrate that the *hp-d* adaptive B-spline version of the FCM is able to reduce the error level and to improve the rate of convergence also in the three-dimensional case. Hierarchical B-spline discretizations of moderate polynomial degree p with local hierarchical overlays around interfaces offer a good trade-off between accuracy and accessibility with standard workstations.

5.3.3 An inter-penetrating phase composite

Multi-material composites provide high stiffness at reduced weights, and are therefore frequently used for lightweight structures in automotive and aerospace applications. A recent example are inter-penetrating phase composites (IPC), which combine open-cell aluminium foams with a filling material, such as rubber silicates [46], thermoplastic polymers [51] or epoxy-based syntactic foams [114]. The *hp-d* adaptive B-spline version of the FCM is applied to simulate a nonlinear compression test for an IPC sample. The specimen consists of an aluminium foam characterized by Young's modulus E_{foam}=70.000 N/mm^2 and Poisson's

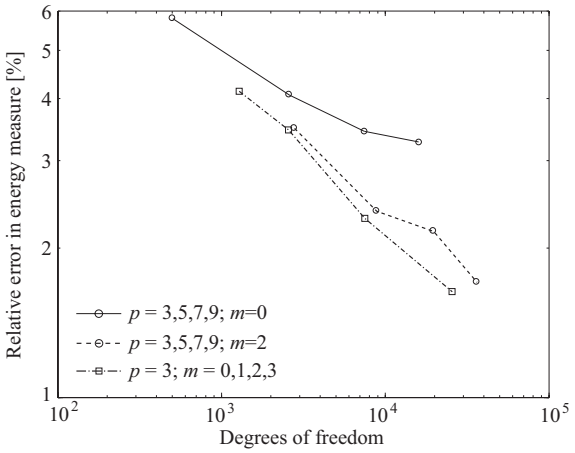


Figure 5.21: Convergence behavior obtained with different adaptive strategies for the spherical inclusion problem.

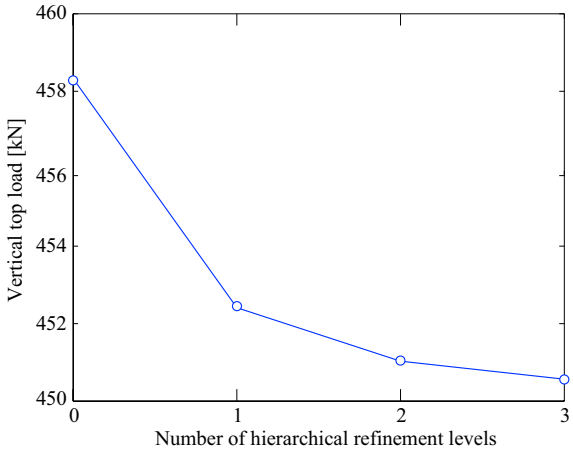


Figure 5.22: Convergence of the top load with increasing number of *hp-d* adaptive subdivision overlays for the IPC composite.

ratio $\nu = 0.35$, which is embedded in an alkyd resin matrix with E_{matrix}=21.000 N/mm^2 and $\nu = 0.35$. Its geometry is given by a voxel model similar to Figure 4.20. The STL triangulation of the foam surface shown in Figure 5.23 provides a good idea of the material interfaces.

As shown for the example of the 3D cube with spherical inclusion, local overlays of hierarchical subdivision B-splines on a grid of moderate polynomial degree p yield good accuracy at a manageable computational cost. Therefore, the foam sample of dimensions $10 \times 10 \times 10$mm is discretized by a cubic B-spline grid of 8×8×8 knot span finite cells, which are adaptively refined around material interfaces by three levels of hierarchical B-splines, amounting to a system with 215,799 degrees of freedom. All geometry related operations, in particular the point location query that determines whether an integration point is located in the foam or the matrix, is carried out according to the voxel based procedures described in Section 4.3.1. The hierarchical grid structure is shown in Figure 5.24. The IPC sample is assumed as part of a larger specimen, which is uniformly compressed along the vertical axis. A corresponding RVE model [188, 224] specifies Dirichlet boundary conditions as follows: Displacements normal to the top surface are gradually increased to -1 mm (10% compressive deformation), modelling the influence of a testing machine, whereas the displacements normal to all other surfaces are fixed due to the bottom support and the influence of the surrounding material of the specimen. The geometrically nonlinear system is solved by applying the displacement load on top in four increments, each of which requires five to six Newton-Raphson iterations to converge below a norm of the residual of 10^{-8}.

Convergence with increasing adaptive hierarchical refinement is illustrated in Figure 5.22 by the vertical top force, obtained by integrating the normal stress component over the top surface of the *hp-d* adaptive finite cell discretizations with m=0,1,2 and 3 hierarchical overlay levels. It can be observed that the vertical top force converges to a value of around 451 kN. Figure 5.25 plots von Mises stresses obtained with three overlay levels on the deformed structure. The *hp-d* adaptive B-spline version yields steady stress patterns free of oscillations away from the interfaces. Characteristic phenomena of multiphase materials typically occurring close to interfaces are stress concentrations in the stronger phase and stress minima in the weaker phase, which are clearly reproduced and observable in the stress plot. The present example

demonstrates the potential of *hp-d* adaptivity for the simulation of multiphase structures with very complex interface geometries, where avoiding the expensive generation of explicit interface parameterizations or conforming meshes can considerably simplify and speed up the simulation process.

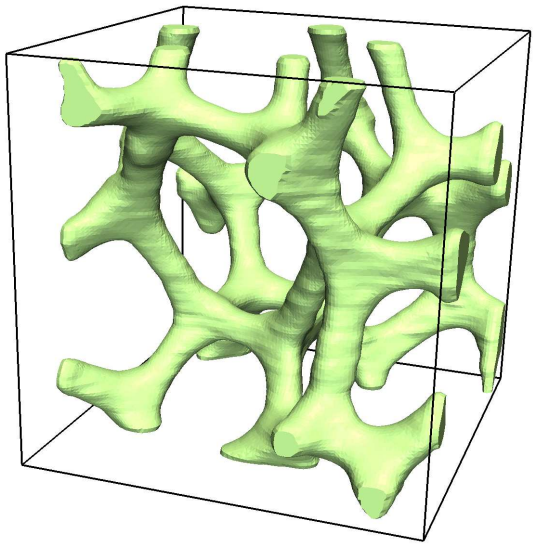


Figure 5.23: STL mesh of the material interfaces between the foam and the resin matrix in the sample cube.

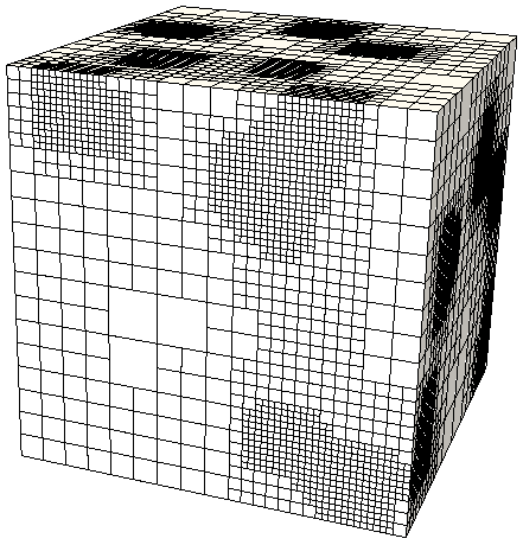


Figure 5.24: Hierarchical overlay mesh of the *hp-d* adaptive B-spline discretization, resolving the location of interfaces.

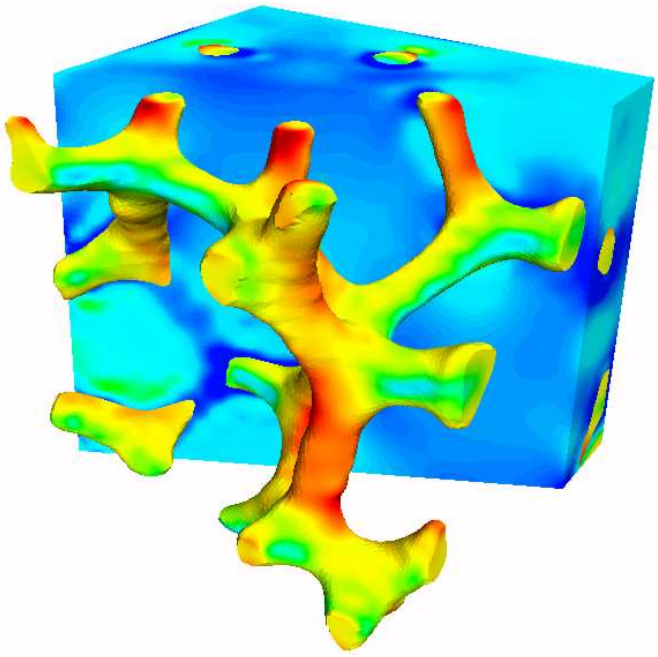


Figure 5.25: Von Mises stress plot, obtained from the nonlinear simulation of a compression test for an IPC composite sample. Only part of the matrix is visualized to reveal the stresses in the foam.

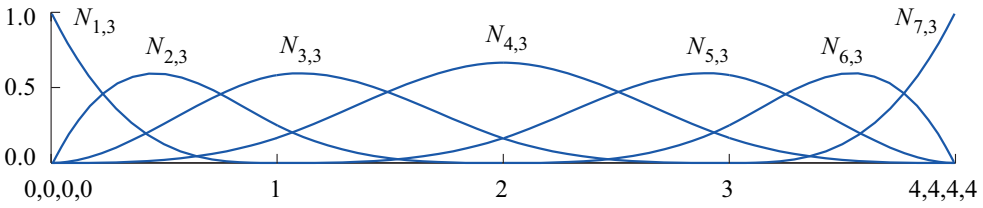
Chapter 6

Hierarchical refinement of non-uniform rational B-splines (NURBS)

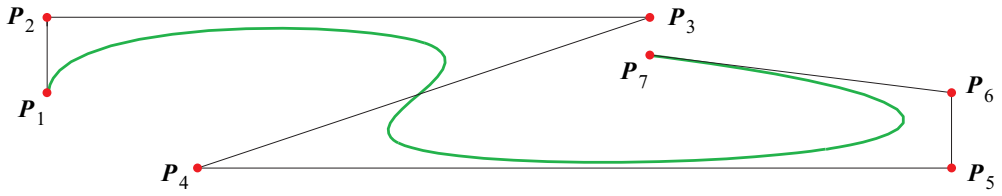
Isogeometric analysis (IGA) was introduced by Hughes and co-workers [56, 109] to bridge the gap between computer aided design (CAD) and finite element analysis (FEA). Its core idea is to use the same basis functions for the representation of geometry in CAD and the approximation of solutions fields in FEA. An important requirement for efficient FEA is local mesh refinement, which permits adaptive resolution of local features in the solution [4, 193]. Due to their relative simplicity and ubiquity in today's CAD tools, isogeometric analysis has been largely based on non-uniform rational B-splines (NURBS) [56, 109]. However, in contrast to the standard Lagrangian nodal basis, a multivariate NURBS basis does not provide a natural possibility for truly local mesh refinement [22, 56]. The standard knot insertion based refinement of NURBS is a global process that propagates throughout the domain due to the tensor product structure of the basis. The demand for adaptivity in isogeometric analysis has recently expedited a shift from NURBS to other geometry representations such as T-splines [22, 69, 185] or PHT-splines [143, 144], which naturally accommodate local mesh refinement, but are considerably more difficult in terms of theory and implementation.

We believe that hierarchical refinement offers a suitable way to overcome the local refinement restriction in NURBS based isogeometric analysis. The idea of hierarchical B-spline refinement, which was introduced by Forsey and Bartels [88, 89] for local surface refinement in CAD, involves no propagation of refinement and automatically satisfies linear independence of the refined basis. It has been successfully adopted on the analysis side by Höllig and co-workers [105, 106, 119] for local mesh refinement in B-spline finite elements. In the following, we will explore the combination of hierarchical refinement based on the principle of B-spline subdivision with ideas of the *hp-d* approach [158, 179] presented in the previous chapter and established concepts for hierarchical refinement of standard nodal based finite elements [65, 98, 122, 217]. The result offers truly local refinement of tensor-product B-splines and NURBS, full analysis suitability of the refined basis, straightforward implementation in tree data structures and simple generalization to higher dimensions at the same time.

We begin with a brief introduction to B-spline and NURBS basis functions, which expand the short introduction on structured B-spline discretizations presented in Section 2.2.2 as part of the B-spline version of the finite cell method. We then illustrate our basic refinement concepts for a simple 1D example, before we extend them to B-splines in multiple dimensions and outline the main aspects of an efficient and straightforward implementation. Finally, we



(a) Cubic B-spline patch with interpolatory ends.



(b) B-spline curve generated from the above basis using control points $\boldsymbol{P}_i$.

Figure 6.1: Example of cubic B-spline basis functions and a corresponding B-spline curve in 2D.

generalize our hierarchical refinement procedure to non-uniform rational B-splines for adaptive NURBS based isogeometric analysis.

6.1 B-spline and NURBS basis functions

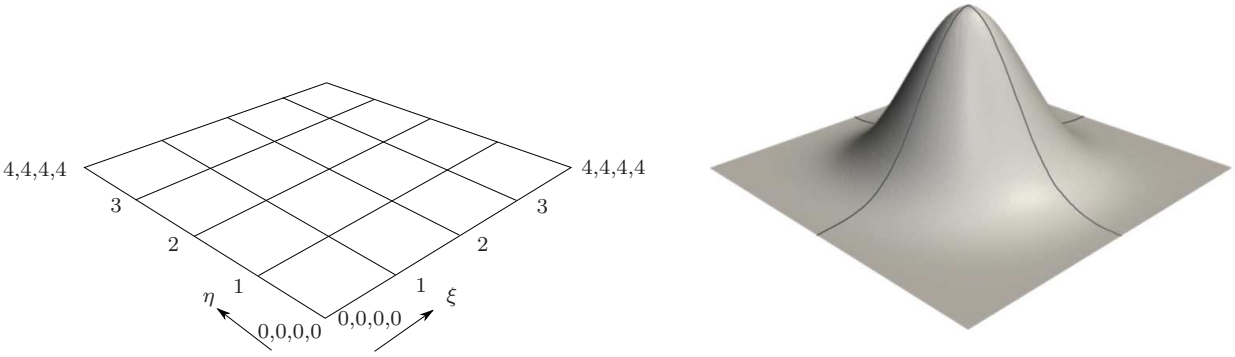
We start with a brief review of some technical aspects of B-spline and NURBS bases for isogeometric analysis, placing particular emphasis on the features important for understanding the concepts and algorithms of their hierarchical refinement. Readers interested in a broader and more detailed introduction are referred to the books of Piegl and Tiller [154], Rogers [165] and Farin [82], which offer a comprehensive review of the underlying geometric concepts and algorithms.

6.1.1 Univariate B-splines

A univariate B-spline basis of polynomial degree p consists of n basis functions $N_{i,p}(\xi)$, where $i = 1, \ldots, n$ and ξ is an independent variable defined in a suitable parameter space. It is generated from a knot vector Ξ, which is a non-decreasing sequence of coordinates in the parameter space, written as

$$\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}, \quad \xi_1 \leq \xi_2 \leq ... \leq \xi_{n+p+1} \tag{6.1}$$

The knots of each individual basis function $N_{i,p}(\xi)$ can be identified as the consecutive entries $\{i, i+1, ..., i+p+1\}$ in Ξ. The resulting $p+1$ knot spans contain piecewise polynomials, which join smoothly up to a continuous differentiability of C^{p-1}. If constructed from equally spaced knots, a B-spline basis function is called uniform. B-spline basis functions $N_{i,p}(\xi)$ of arbitrary polynomial degree p can be generated recursively with the Cox-de Boor formula, starting from



(a) Tensor product structure of open knot vectors in the parameter space.

(b) Cubic basis function generated from Ξ=$(0, 1, 2, 3, 4)$ in ξ- and η-directions.

Figure 6.2: Bivariate cubic knot spans and a corresponding uniform B-spline basis function.

piecewise constants $N_{i,0}$

$$N_{i,0}\left(\xi\right) = \begin{cases} 1, & \text{if } \xi_i \leq \xi \leq \xi_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{6.2}$$

$$N_{i,p}\left(\xi\right) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}\left(\xi\right) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}\left(\xi\right) \tag{6.3}$$

A knot can occur at the same coordinate k times, which reduces the smoothness of the B-spline curve at the knot position to C^{p-k}. Figure 6.1a illustrates a B-spline basis of polynomial degree p=3 and knot vector Ξ=$\{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$, where knots at the beginning and the end are repeated to make the basis interpolatory. The resulting group of basis functions is often referred to as a B-spline patch with open knot vectors.

Having constructed the corresponding basis functions, we can build a B-spline curve in d_s dimensions by a linear combination of basis functions

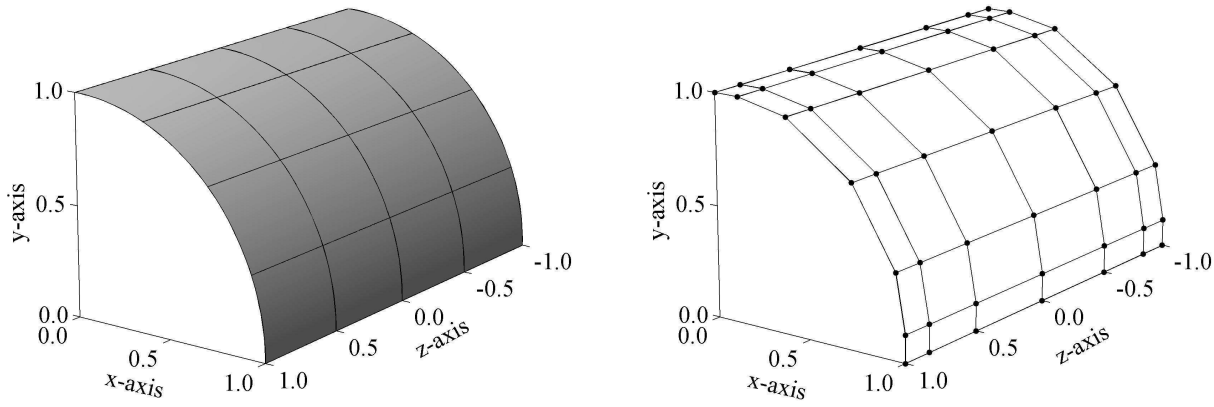
$$\boldsymbol{C}(\xi) = \sum_{i=1}^{n} \boldsymbol{P}_i \; N_{i,p}(\xi) \tag{6.4}$$

where coefficients $\boldsymbol{P}_i \in \mathbb{R}^{d_s}$ are called control points. Piecewise linear interpolation of the control points defines the control polygon. An example generated from the B-spline basis shown in Figure 6.1a is provided in Figure 6.1b.

6.1.2 Multivariate B-splines

Multivariate B-spline bases for the description of surfaces and solids can be easily constructed by taking the tensor product of corresponding univariate B-spline basis functions, which allows for a straightforward generalization of almost all univariate properties and algorithms [56, 154]. Note that we need two different dimension indices d_s and d_p for the physical space and the B-spline parameter space, respectively, where $d_s \geq d_p$. In d_s dimensions, multivariate B-spline basis functions are generated from d_p univariate knot vectors

$$\Xi^{\ell} = \{\xi_1^{\ell}, \xi_2^{\ell}, ..., \xi_{n_\ell + p_\ell + 1}^{\ell}\} \tag{6.5}$$



(a) NURBS elements. **(b)** Corresponding control mesh.

Figure 6.3: NURBS surface of a quarter of a cylinder, generated with the bivariate B-spline basis of Figure 6.2 and a set of suitable control points $\boldsymbol{P}_i$ and weights w_i.

where $\ell = 1, \ldots, d_p$. Parameter p_ℓ indicates the polynomial degree along parametric direction ℓ and n_ℓ is the associated number of basis functions. The resulting univariate B-spline basis functions of each direction ℓ can then be denoted by $N^{\ell}_{i_\ell,p_\ell}$, from which multivariate basis functions can be generated by

$$B_{\boldsymbol{i},\boldsymbol{p}}\left(\boldsymbol{\xi}\right) = \prod_{\ell=1}^{d} N^{\ell}_{i_\ell,p_\ell}(\xi^\ell) \tag{6.6}$$

Multi-index $\boldsymbol{i}=\{i_1, \ldots, i_{d_p}\}$ denotes the position in the tensor product structure, $\boldsymbol{p}=\{p_1, \ldots, p_{d_p}\}$ indicates the polynomial degree, and $\boldsymbol{\xi}=\{\xi^1, \ldots, \xi^{d_p}\}$ are the independent variables in each direction ℓ. An example of a two-dimensional knot vector structure and a uniform bivariate B-spline are given in Figures 6.2a and 6.2b, respectively..

Setting d_p=2 and d_p=3, we can now represent B-spline surfaces and solids, respectively, as a linear combination of multivariate basis functions

$$\boldsymbol{S}(\boldsymbol{\xi}) = \sum_{\boldsymbol{i}} \boldsymbol{P_i}\, B_{\boldsymbol{i},\boldsymbol{p}}\left(\boldsymbol{\xi}\right) \tag{6.7}$$

where the sum is taken over all combinations of multi-index $\boldsymbol{i}$. In the multivariate case, the control points $\boldsymbol{P_i} \in \mathbb{R}^{d_s}$ form the so-called control mesh.

6.1.3 Non-uniform rational B-splines

Many geometric entities being ubiquitous in engineering design, such as circles, ellipses, spheres or tori, cannot be represented exactly by piecewise polynomials. The most important of them, however, can be obtained through a projective transformation of a corresponding B-spline object in $\mathbb{R}^{d_s+1}$, yielding a non-uniform rational B-spline (NURBS) object [56, 82, 154, 165]. In the univariate case, NURBS basis functions $R_{i,p}(\xi)$ are given by

$$R_{i,p}\left(\xi\right) = \frac{w_i N_{i,p}(\xi)}{\sum_{j=1}^{n} w_j N_{j,p}(\xi)} \tag{6.8}$$

where $N_{i,p}(\xi)$ are polynomial B-spline basis functions, each of which is accompanied by a scalar weight w_i. In analogy with the previous section, multivariate NURBS basis functions can be obtained from a multivariate tensor product B-spline basis as

$$R_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi}) = \frac{w_{\boldsymbol{i}} B_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi})}{\sum_{j=1}^{n} w_{\boldsymbol{j}} B_{\boldsymbol{j},\boldsymbol{p}}(\boldsymbol{\xi})} \tag{6.9}$$

NURBS curves, surfaces and solids are then defined in the same manner as corresponding B-spline objects of Equations (6.4) and (6.7) as

$$\boldsymbol{S}(\boldsymbol{\xi}) = \sum_{\boldsymbol{i}} \boldsymbol{P}_{\boldsymbol{i}}\, R_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi}) \tag{6.10}$$

An example is given in Figure 6.3. The NURBS geometry of Figure 6.3a, which represents a quarter of a cylinder exactly, is generated by inserting the bivariate cubic B-spline basis of Figure 6.2 together with a suitable set of control points and weights into the NURBS Equations (6.9) and (6.10). Figure 6.3b shows the corresponding control mesh composed of the control points $\boldsymbol{P}_i$. Suitable control points and weights can be derived in the present simple case from [82, 154, 165] or for more complex examples from CAD tools such as Rhino [166, 183].

6.2 A concept for hierarchical refinement based on B-spline subdivision

In the following, we briefly review B-spline subdivision and show how this concept can be deployed to set up a hierarchical scheme for local refinement of B-spline basis functions, which combines an intuitive principle, full analysis suitability and straightforward implementation. We illustrate basic ideas of our hierarchical refinement strategy with a simple advection-diffusion example in one dimension.

6.2.1 Refinability of B-spline basis functions by subdivision

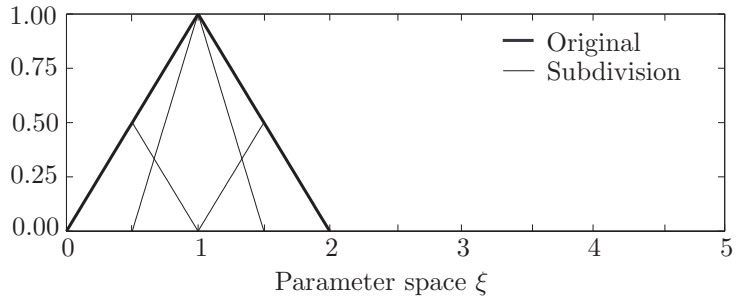
A further remarkable property of uniform B-splines is their natural refinement by subdivision. For a univariate B-spline basis function N_p of polynomial degree p, the subdivision property leads to the following two-scale relation [50, 212, 226]

$$N_p(\xi) = 2^{-p} \sum_{j=0}^{p+1} \binom{p+1}{j} N_p(2\xi - j) \tag{6.11}$$

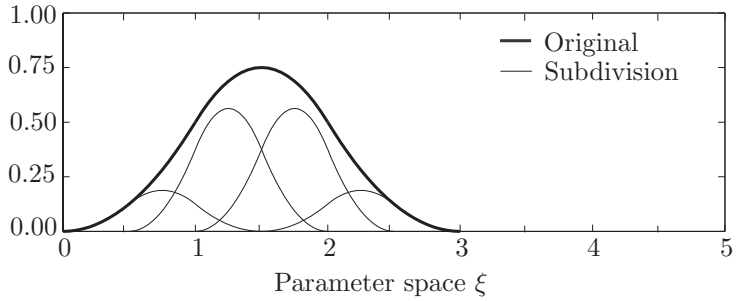
In other words, a B-spline can be expressed as a linear combination of contracted, translated and scaled copies of itself, which is illustrated for B-splines of different polynomial degrees in Figure 6.4. Equation (6.11) does not hold for non-uniform B-splines with repeated knots, but similar subdivision rules can be constructed [153, 169].

Due to their tensor product structure, the generalization of subdivision to multivariate B-splines is a straightforward extension of Equation (6.11) and can be written as

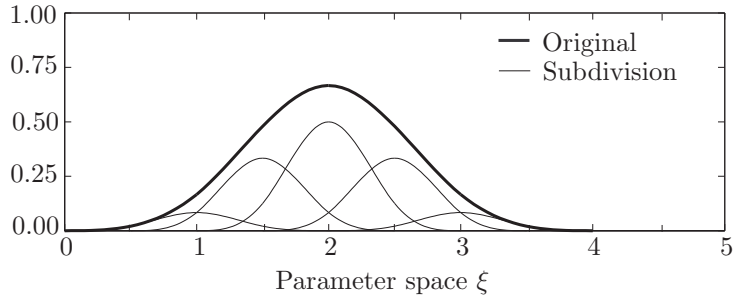
$$B_{\boldsymbol{p}}(\boldsymbol{\xi}) = \sum_{\boldsymbol{j}} \left(\prod_{\ell=1}^{d} 2^{-p_\ell} \binom{p_\ell + 1}{j_\ell} N_{p_\ell}(2\xi^\ell - j_\ell) \right) \tag{6.12}$$



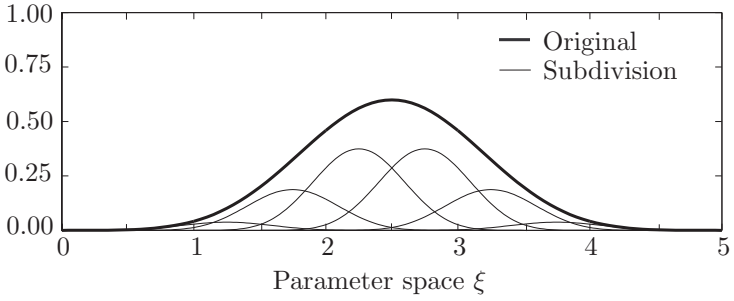
(a) Linear B-spline.



(b) Quadratic B-spline.

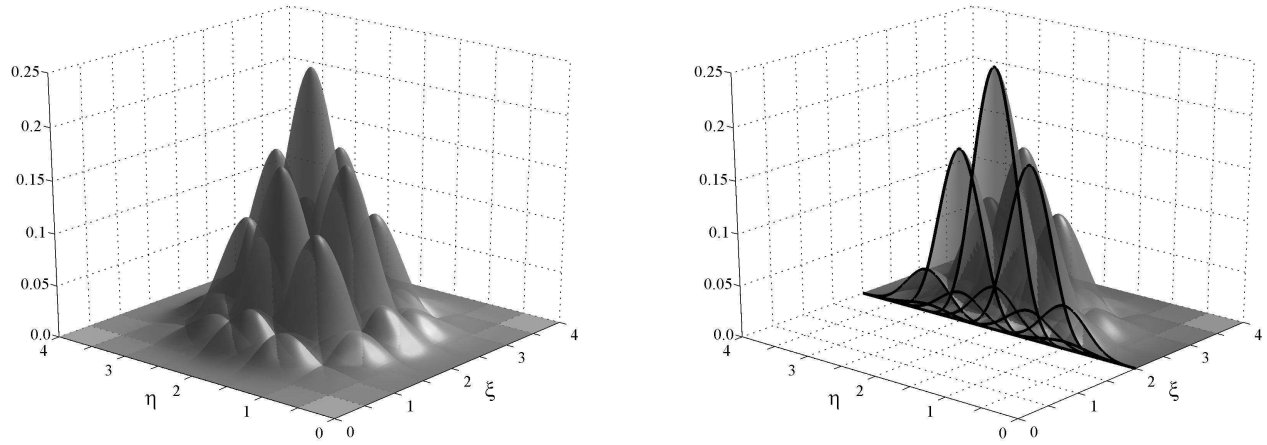


(c) Cubic B-spline.



(d) Quartic B-spline.

Figure 6.4: Subdivision of an original uniform B-spline into p+2 contracted B-splines of half the knot span width, illustrated for polynomial degrees p=*1* through 4.



(a) Contracted, translated and scaled B-splines. **(b)** Cut along ξ=2.0.

Figure 6.5: Subdivision of the bivariate cubic B-spline shown in Figure 6.2.

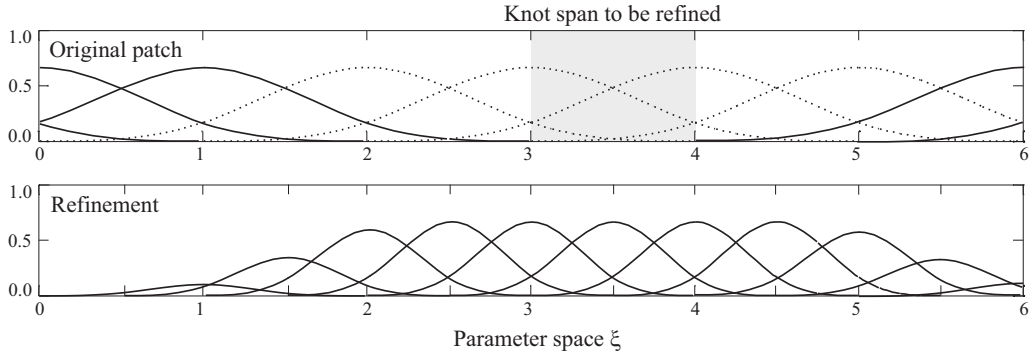
Following Section 6.1.2, multi-indices $\boldsymbol{j}=\{i_1,\ldots,i_{d_p}\}$, $\boldsymbol{p}=\{p_1,\ldots,p_{d_p}\}$ and $\boldsymbol{\xi}=\{\xi^1,\ldots,\xi^{d_p}\}$ denote the position in the tensor product structure, the polynomial degree and the independent variables in each direction ℓ of the d_p-dimensional parameter space. Figure 6.5 illustrates the new basis functions resulting from the multivariate two-scale relation Equation (6.12) applied to the bivariate cubic B-spline of Figure 6.2b. The most widely known application of Equations (6.11) and (6.12) is the development of highly efficient subdivision algorithms for the fast and accurate approximation of smooth surfaces by control meshes in computer graphics [153, 169, 212, 226].

6.2.2 Construction of adaptive hierarchical approximation spaces

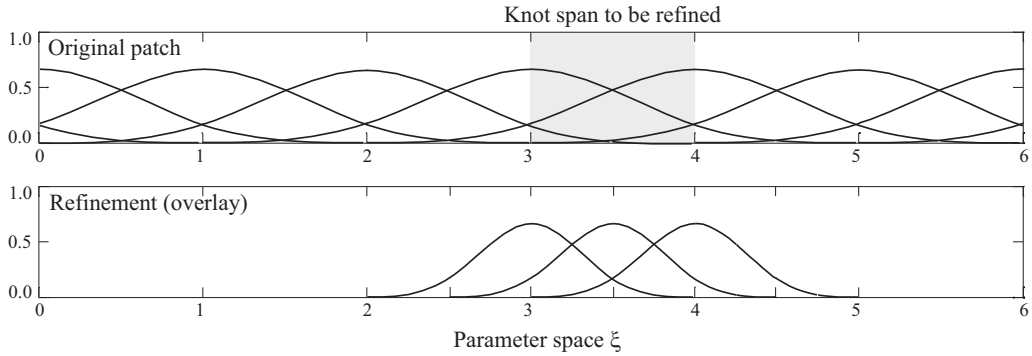
In the following, we will derive step by step a hierarchical scheme for local refinement of B-splines in one dimension, which combines concepts from B-spline subdivision, the *hp-d* adaptive approach [158, 178, 179] and existing hierarchical refinement techniques for B-spline finite elements [88, 106, 119] and for standard nodal based FEA [65, 122, 217]. Our main goal is to arrive at a local refinement strategy, which maintains theoretical consistency, can be straightforward generalized to two and three dimensions and NURBS, but at the same time allows an intuitive understanding of the underlying theoretical principles and can be implemented in arbitrary dimensions with manageable coding efforts.

6.2.2.1 Element and basis viewpoints on refinement

Traditionally, refinement in FEA adopts an element viewpoint, since the centerpiece of standard mesh refinement consists of the geometric division of elements [98, 122]. Furthermore, an element by element view accommodates most error estimators [4, 193], which specify the error element wise, and naturally corresponds to the way finite element procedures are traditionally implemented. In isogeometric analysis, however, it is often more suitable to look at the complete basis, since an element centered viewpoint may obstruct an intuitive understanding of refinement principles for basis functions, which extend over several knot span elements



(a) Subdivision refinement, splitting original B-splines with support in the element (dotted lines) according to the two-scale relation. The refined basis consists of all functions shown with solid lines.



(b) Hierarchical refinement, inspired by the *hp-d* adaptive approach [178]. The combination of the original patch and three contracted B-splines yields the refined basis.

Figure 6.6: Nucleus operation: the refinement of one knot span element.

[56, 98]. Therefore, we will adopt the element viewpoint, wherever we feel that aspects can be reasonably explained from there. Wherever we feel that it might become too restrictive or interferes with the consistency of the hierarchical methods at issue, we switch to the more comprehensive basis viewpoint.

6.2.2.2 Two-level hierarchical refinement for one element

In a first step, let us define a nucleus operation, from which we start our development: the refinement of one knot span element. Figure 6.6 exhibits a portion of a B-spline patch, where the element in the center should be refined. A straightforward approach, introduced for B-spline FEA by Kraft [119] and Höllig [106] and recently extended to isogeometric analysis by Vuong et al. [210], is the application of the two-scale relation Equation (6.11) to all basis functions with support in the knot span element under consideration. Contracted B-splines resulting from the subdivision of different B-spline basis functions, but with the same support, are superposed by adding their scaling factors. This leads to full hierarchical B-spline basis functions in the center of the refinement, while hierarchical B-splines at the boundary are gradually decreased as shown in Figure 6.6a. However, the direct subdivision

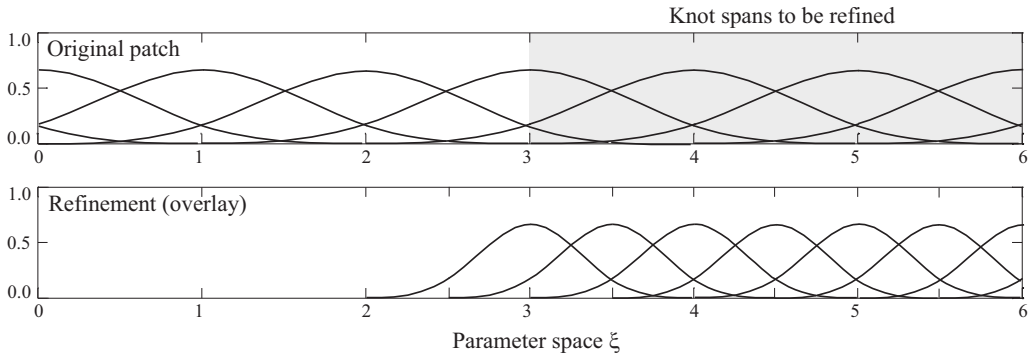


Figure 6.7: Hierarchical refinement in the sense of the *hp-d* approach for the three rightmost knot span elements in a row. The overlay is generated by repeating the nucleus operation of Figure 6.6b.

based refinement strategy results in a large spread of the refinement beyond the bounds of the knot span element under consideration, which obstructs the localization of refinement. Due to the increase of the spread with p, this is especially true for higher polynomial degrees. In addition, it requires a correct book-keeping and addition of scaling factors, which considerably complicates its implementation.

Therefore, we follow a different approach here and borrow the main idea of the *hp-d* adaptive approach, which was originally introduced for the p-version of the FEM [158] and successfully applied to B-spline bases in [178]. In an *hp-d* sense, we add an overlay of some B-splines of contracted knot span width to the original B-spline basis. At this point, no changes in the original basis functions are required, since we can infer from Equation (6.11) that single B-splines of contracted knot span width are linearly independent to original B-splines of full knot span width. The resulting basis is the combination of B-splines shown in Figure 6.6b, where original and overlay basis functions are plotted on separate levels, which reflects the two-level hierarchy between the original basis and its refinement overlay. Furthermore, we do not change the amplitude of contracted B-splines, thus ignoring the presence of scaling factors in Equation (6.11). We will show that this simplification can be maintained, when we generalize this refinement strategy to higher dimensions and NURBS later on.

6.2.2.3 Two-level hierarchical refinement for several elements

The refinement rule introduced in Figure 6.6b for one element adds an overlay consisting of the contracted B-spline central to the element at issue as well as its right and left neighbor. Let us proceed one step further to the refinement of several knot span elements in a row. Figure 6.7 illustrates the two-level hierarchical basis, which results from a repetition of the nucleus operation illustrated in Figure 6.6b for the three rightmost elements in the patch. We do not add contracted B-splines a second time, if already generated from the refinement operation of a neighboring element.

The reason for the specific choice of the nucleus operation becomes clear by consideration of the number of contracted B-splines. If we take fewer B-splines for each element refinement than shown in Figure 6.6b and omit the left and right neighbor, we would not obtain a complete row of contracted B-splines in Figure 6.7, since every second contracted B-spline would be missing. If we take more, the refinement would again spread out further beyond the leftmost

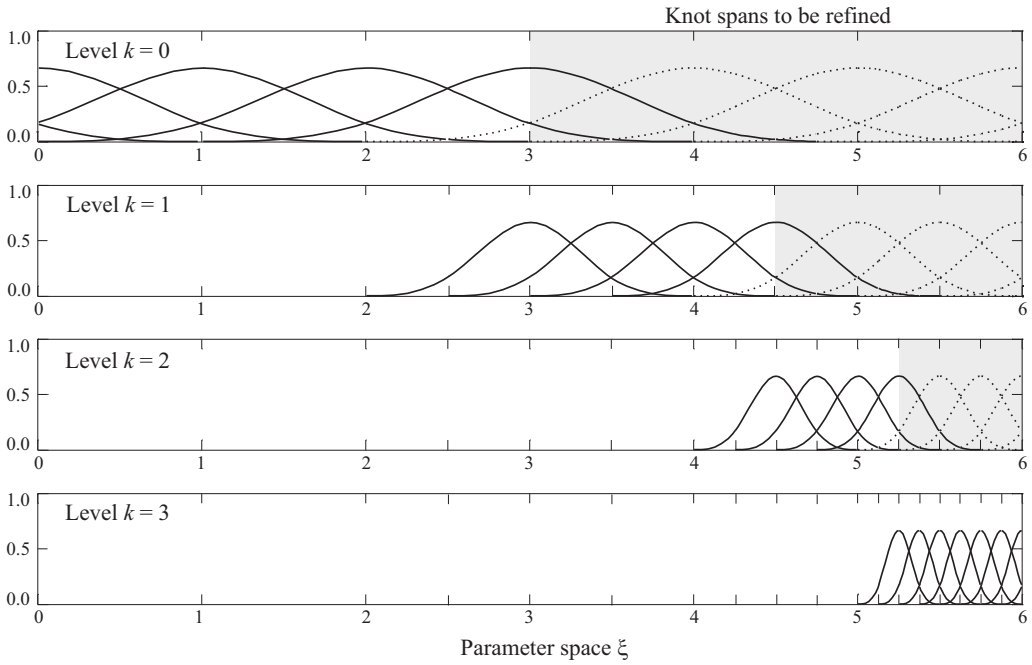


Figure 6.8: Hierarchical multi-level refinement: B-splines of level k plotted in dotted line can be represented by a linear combination of B-splines of the next level $k+1$ according to the two-scale relation Equation (6.11) and therefore need to be removed from the basis.

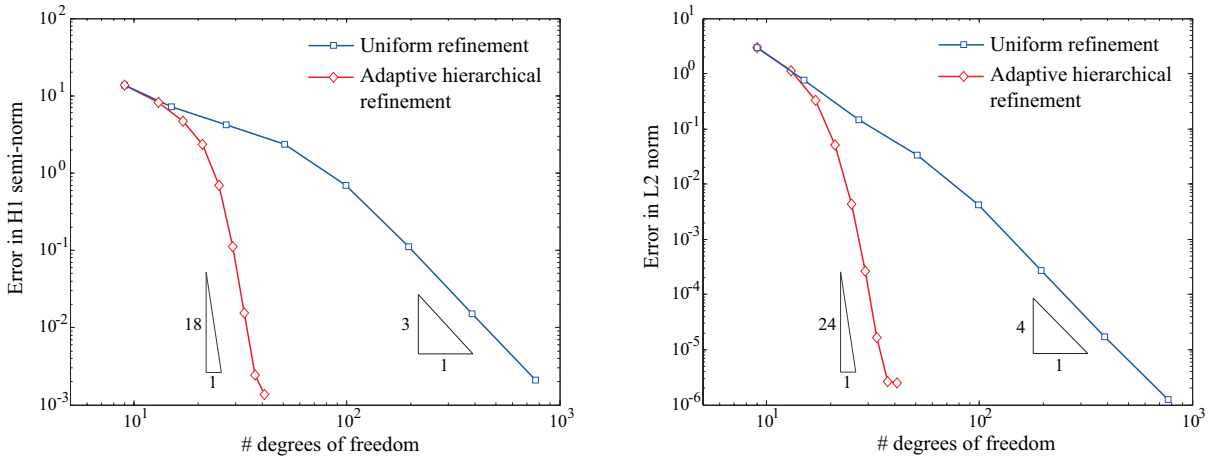
element at issue, which would counteract the localization of refinement. The refinement rule of Figure 6.6b is valid for p=3, but can be easily transferred to B-spline bases of other polynomial degrees by looking for the minimum number of contracted B-splines per element, with which a complete row of contracted B-splines in the overlay level can be achieved, when several elements are refined.

6.2.2.4 Multi-level hierarchical refinement

In order to increase the degree of local refinement, we can repeat the procedure described in the previous paragraphs several times. In doing so, we proceed from the two-level hierarchy of a single refinement step to a general multi-level hierarchy, consisting of several overlay levels. Let us introduce the level counter k, where k=0 denotes the original B-spline patch. In each refinement step, the nucleus operation is applied to elements of the currently finest level k to produce a new overlay level $k+1$. Hierarchically contracted B-splines of the new level $k+1$ are found by bisecting the knot span width with respect to level k, so that the specific width h_k of each level can be found by the relation

$$h_k = 2^{-k}\, h, \quad 1 \leq k \leq m \tag{6.13}$$

where h denotes the original knot span width of the unrefined patch and m the total number of levels in the hierarchy. The multi-level refinement procedure is illustrated in Figure 6.8, where the nucleus operation is successively applied to the three rightmost knot span elements of each level k. The resulting grid consists of a nested sequence of bisected knot span elements, and multiple hierarchical overlay levels of repeatedly contracted uniform B-splines.



(a) H^1 semi-norm. **(b)** L^2 norm.

Figure 6.9: Convergence of cubic hierarchical B-spline bases for the 1D advection-diffusion example.

6.2.2.5 Recovering linear independence

In order to guarantee full analysis suitability of the hierarchically refined B-spline basis, we have to ensure its linear independence. Comparing the different levels in the hierarchy of Figure 6.8, one can immediately observe that each overlay level k+1 consists of more than p+2 consecutive refined B-splines. As a consequence, their linear combination is capable of representing some of the B-spline basis functions of the previous level k according to the two-scale relation Equation (6.11). Therefore, we need to identify all B-spline basis functions that are representable by a combination of refined B-spline basis functions of the next level k+1 and remove them from the hierarchical basis. In Figure 6.8, basis functions to be taken out are shown with dotted lines, while the final linear independent hierarchical B-spline basis consists of all basis functions shown with solid lines.

6.2.3 A simple model problem in 1D

In the following, we test the efficiency of the hierarchically refined B-spline basis with a standard steady advection-diffusion problem in 1D, governed by the following equation and Dirichlet boundary conditions

$$a\,\frac{\partial u}{\partial x} + D\,\frac{\partial^2 u}{\partial x^2} = 0 \tag{6.14a}$$

$$u(x=0) = 0; \quad u(x=L) = 1 \tag{6.14b}$$

Parameters u, a=100, D=1 and L=3 denote the unknown concentration, the velocity, the diffusion coefficient and the length of the domain, respectively. Dirichlet constraints are specified at both ends. Here, the nature of the problem is dominated by advection, indicated by the high global Péclet number Pe=$aL/D \gg 1$. This leads to an almost immediate jump of the solution from 0 to 1 at the right hand end of the 1D domain Ω, involving very high gradients. An in-depth discussion of this problem and its exact solution can be found in [67, 223].

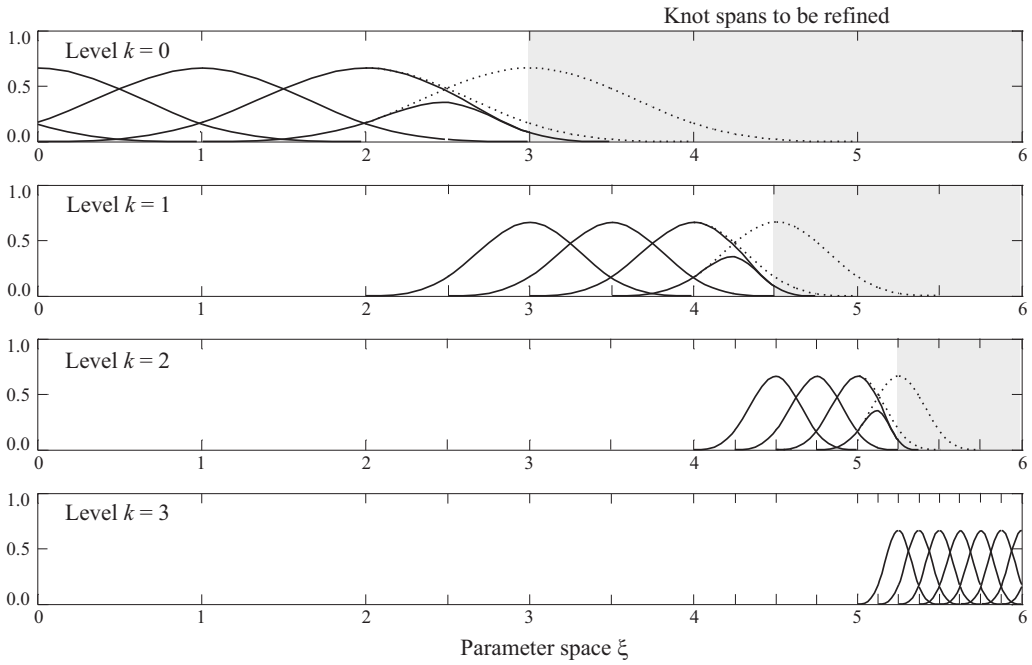
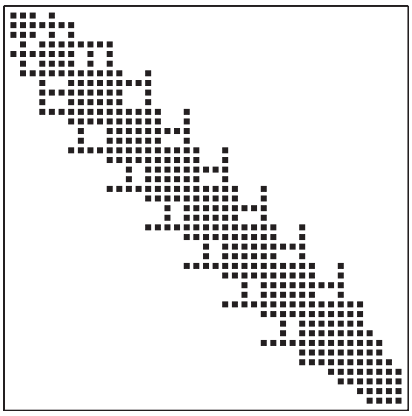


Figure 6.10: The subtraction procedure removes multiplicities of hierarchical B-splines according to the two-scale relation, which further reduces the coupling of B-splines throughout the hierarchy.

We apply a standard Galerkin discretization, where the dominance of the non-symmetric advection operator over the diffusion operator in Equation (6.14a) leads to spurious oscillations. While these issues are usually addressed by consistent stabilization techniques [36, 53, 67, 148], we use a sequence of cubic hierarchical bases in the sense of Figure 6.8 to obtain an accurate solution. In each refinement step, we generate an additional overlay level by adaptively refining the three rightmost elements. Since they are not interpolatory at the ends, the hierarchical basis requires weak enforcement of Dirichlet constraints, for which a simple penalty method [84, 220] is applied here. Figures 6.9a and 6.9b show the convergence in the H^1 semi-norm and L^2 norm, respectively, obtained with uniform h-refinement and adaptive hierarchical refinement. Uniform refinement doubles the number of equidistant knot span elements in each refinement step, which leads to optimal rates of convergence as predicted by the mathematical theory for uniform h-refinement [197, 221]. Due to their adaptivity, the hierarchical bases achieve rates of convergence, which are far better. To arrive at the final error level in both the H^1 and L^2 cases, the hierarchical bases require about one order of magnitude less degrees of freedom than uniform h-refinement. After seven refinement steps, the largest part of the error does not stem from the excessively refined right boundary anymore, so that the convergence rate levels off.

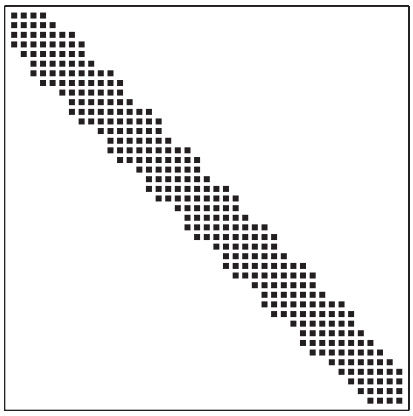
# of hierarchical levels k	0	1	2	3	4	5	6	7	8
Without subtraction	42.0	84.8	85.6	84.5	88.3	106.3	197.6	388.6	774.1
With subtraction	42.0	51.2	44.2	41.4	44.8	67.0	126.3	250.8	502.3

Table 6.1: Condition number of the stiffness matrix for different numbers of hierarchical levels without and with subtraction procedure.



Number of dofs: 41
Number of non-zero entries: 399
Maximum bandwidth: 17

(a) Without subtraction procedure.



Number of dofs: 41
Number of non-zero entries: 307
Maximum bandwidth: 9

(b) With subtraction procedure.

Figure 6.11: Sparsity pattern of a stiffness matrix without and with subtraction procedure.

6.2.4 Condition number and sparsity of the stiffness matrix

In the hierarchically refined basis of Figure 6.8, some basis functions occur explicitly as refinement functions on some level k, but are also contained implicitly in some B-spline basis function of higher level according to the two-scale relation Equation (6.11). We can achieve a further improvement of the conditioning and sparsity of the stiffness matrix, if we detect and remove these multiplicities. In principle, this can be achieved by checking each hierarchical B-spline of the next level $k+1$ to determine if it has common support with a B-spline of the current level k. If it does, we check whether the former is part of the linear combination of subdivision B-splines that result from the two-scale relation Equation (6.11) applied to the latter. If this is also true, we subtract the hierarchical B-spline of the next level $k+1$, multiplied by the corresponding scaling factor, from the B-spline of the current level k. Applying this subtraction procedure to the example basis of Figure 6.8, we arrive at the improved hierarchical basis of Figure 6.10, where initial basis functions are shown as dotted lines, while the final basis functions after subtraction are shown as solid lines. We observe that the overlapping of hierarchical levels is further reduced.

Analogous to the previous sections, we create two sequences of hierarchically refined bases with and without subtraction procedure, respectively, which are based on a refinement of the last three elements of each hierarchical level, and apply them to the advection-diffusion problem of Equation (6.14). To exclude any effect from weak boundary conditions, we replace uniform boundary basis functions by interpolatory basis functions created from open knot vectors, so that Dirichlet constraints can be satisfied strongly. The sparsity pattern of the stiffness matrices without and with subtraction procedure and the evolution of the corresponding condition numbers with increasing levels of hierarchical refinement are illustrated in Figure 6.11 and Table 6.1, respectively. The subtraction procedure has a beneficial effect on the structure of the sparse matrix, reducing both bandwidth and number of matrix entries. Note that the ordering of matrices was optimized by the symmetric reverse Cuthill-McKee algorithm [95]. The condition number of the matrix is improved by the subtraction of mul-

tiplicities, but still remains in the order of magnitude of its counterpart without subtraction in the one-dimensional case. We note that in higher dimensions, the difference in the condition number might be more pronounced, since there are many more basis functions that have common support and belong to different hierarchical levels.

From an implementation point of view, the detection of multiplicities requires complex algorithms, which obstruct automation of the refinement process. Its advantages are likely to be moderate, since the performance of iterative solvers is dominated by the condition number of the matrix, while the bandwidth plays a subordinate role. Therefore, we omit the subtraction procedure in the following, when we will focus on efficient and easy-to-implement hierarchical refinement schemes for isogeometric analysis in multiple dimensions.

6.3 Hierarchical refinement of axis-aligned B-splines in multiple dimensions

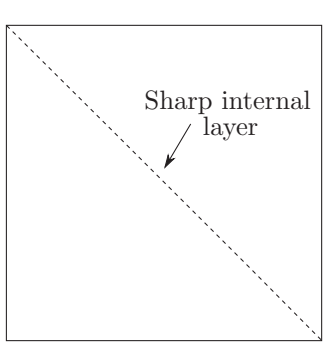
Due to their tensor-product structure, the concept of hierarchical refinement in 1D directly carries over to multi-dimensional B-splines. We discuss its implementation in the framework of tree-like data structures and suggest corresponding algorithms. To keep things simple at this stage, we confine ourselves to axis-aligned B-spline discretizations of cuboidal domains.

6.3.1 Transition from the 1D concept to multiple dimensions

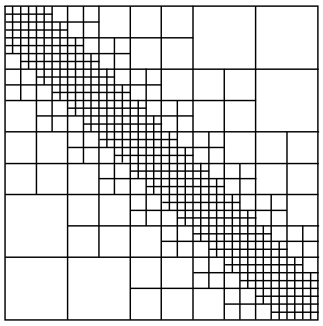
The tensor product structure of multi-dimensional B-splines permits a straightforward generalization of the one-dimensional hierarchical refinement concept presented in Section 6.2 to multiple dimensions. Assume a d_p-dimensional B-spline patch, which is generated according to Section 6.1.2 by d_p univariate knot vectors in each parametric direction ξ^ℓ. An adaptive multi-level B-spline basis can be generated by successively applying the 1D procedures of the previous section, since the tensor product structure allows for a decoupling of refinement operations in each parametric direction ℓ and subsequent d_p-dimensional assembly by multiplication. We will illustrate this by commenting briefly on each refinement step:

- *Nucleus operation* (refinement of a single d_p-dimensional element): Depending on the polynomial degree p_ℓ of each parametric direction, we choose a suitable number of contracted B-splines for each direction ℓ according to the principles outlined in Section 6.2.2. Assuming p=3, the nucleus operation in each parametric direction ℓ corresponds to Figure 6.6b , from which d_p-dimensional hierarchical B-splines can be generated by multiplying the one-dimensional functions in the sense of Figure 6.5.

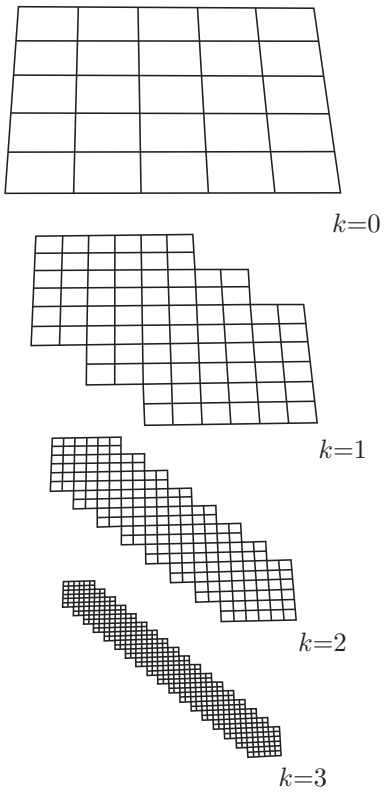
- *Multi-level hierarchy*: The local increase of refinement follows the same concept as described in Sections 6.2.2.3 and 6.2.2.4, where the repetition of the nucleus operation in each overlay level k and the repetition of hierarchical refinement with successively contracted B-splines for the generation of the next overlay level k+1 lead to a multi-level B-spline basis, which naturally accommodates adaptivity in d_p dimensions.



(a) System sketch.



(b) Hierarchical mesh.



(c) Multi-level structure of hierarchically contracted knot spans.

Figure 6.12: Multi-level hierarchical B-splines of p=3 for adaptive refinement along a diagonal internal layer in a square domain. The original patch of level k=0 consists of 5×5 knot span elements.

- *Recovering linear independence*: A linear combination of multi-dimensional hierarchical B-splines of the next level k+1 can represent B-splines of the current level k in the sense of the multi-dimensional two-scale relation Equation (6.12). These linear dependencies need to be identified through the multi-level hierarchy and eliminated by a removal of corresponding higher-level B-splines. In analogy with section 6.2.2.5, this can be achieved by checking in each parametric direction, if the required p+2 contracted B-splines of level k+1 exist in the hierarchical structure.

- *Dirichlet constraints*: Dirichlet boundary conditions can be incorporated weakly by variational methods [19, 78, 84] or strongly by a least squares fit of boundary basis functions [56, 97]. Homogeneous boundary conditions can be imposed strongly by simply removing all basis functions with support at the Dirichlet boundary from the basis.

Multi-dimensional hierarchical refinement is illustrated for the example of a two-dimensional square domain, which should be refined along its diagonal as shown in Figure 6.12a, where we assume a steep gradient due to an internal layer. Figure 6.12b shows the hierarchical mesh, which represents the element structure, over which numerical integration is carried out, so that the coupling of basis functions from different levels can be taken into account. Basis

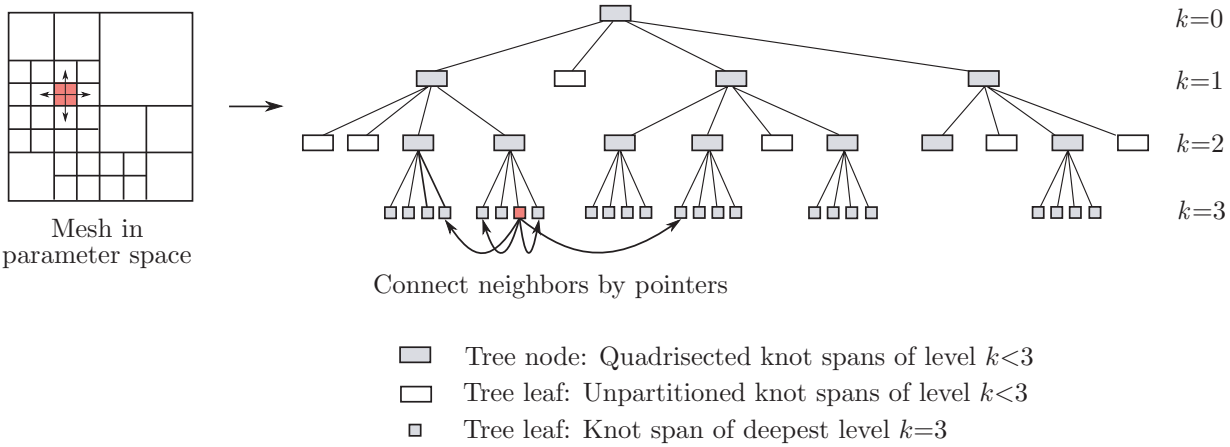


Figure 6.13: Quadtree example illustrating the hierarchical data organization of part of an adaptive mesh. The neighboring relations within each hierarchical level are established by pointers, which are shown here for one element of the finest level.

functions of different levels are defined over knot spans, which are hierarchically contracted in the sense of Equation (6.13). The corresponding multi-level overlay structure is illustrated in Figure 6.12c, where the sequence of hierarchical knot spans is plotted.

6.3.2 Geometry representation and hierarchical mapping

In the general case, tensor-product B-splines are mapped from a regular grid in the parameter space to an arbitrarily shaped physical geometry via control point values according to Equation (6.7). In the special case of an axis-aligned regular B-spline mesh, we can use the simplicity of its cuboidal geometry to describe the mapping from parameter space $\boldsymbol{\xi}$ to physical space $\boldsymbol{x}$ analytically by the following transformation

$$\boldsymbol{x} = \boldsymbol{x}_0(\boldsymbol{\xi}) + \boldsymbol{H}\,\boldsymbol{\xi} \tag{6.15}$$

where $\boldsymbol{x}_0$ denotes the origin of the physical domain in the parameter space, and $\boldsymbol{H}$ is a diagonal matrix, containing the physical knot span width h_{ξ^ℓ} of each parametric direction ℓ. The Jacobian determinant directly follows as

$$J = \det \boldsymbol{H} \tag{6.16}$$

Due to Equations (6.15) and (6.16), the B-spline basis can be disconnected completely from the geometry and serves exclusively for the approximation of the solution fields.

Numerical integration is performed over the hierarchical mesh (cf. Figure 6.12b). Depending on the hierarchical level k of the respective element, an additional mapping that involves the following Jacobian

$$j = \left(\frac{1}{2^k}\right)^{d_p} \tag{6.17}$$

is required, where d_p is the number of dimensions of the B-spline discretization. In the scope of the present paper, integrals are evaluated by Gauss integration, which places $(p+1)^{d_p}$ integration points in each element of the hierarchical mesh.

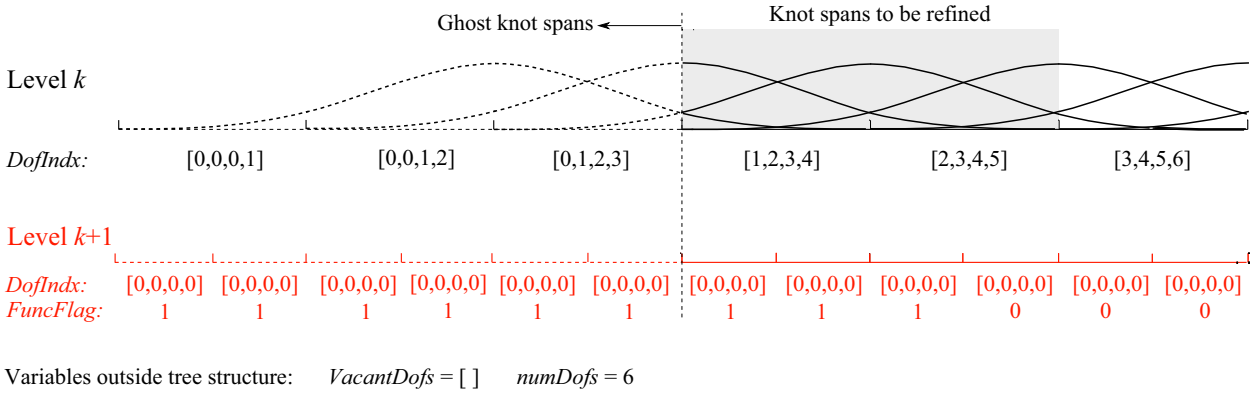


Figure 6.14: Illustration of Algorithm 1: Apply nucleus operation to the first two knot span elements, which creates new knot spans of level k+1 and flag knot spans, where contracted basis functions start. Also split and flag all related ghost knot spans to ensure a correct treatment of boundary basis functions.

6.3.3 Efficient implementation in tree data structures

A considerable advantage of hierarchical refinement in the present form is the preservation of its intuitive multi-level concept irrespective of the dimensionality of the B-spline basis, which allows for an easy generalization of corresponding algorithms in the framework of the tensor product structure. Additionally, its direct correspondence to standard multi-level data structures [172] constitutes the basis for its straightforward and efficient implementation.

6.3.3.1 Binary, quad- and octrees

Hierarchical data structures provide a natural way to decompose and organize spatial data according to different levels of complexity and offer fast access to relevant parts of a dataset [171, 172]. For their efficient implementation, binary, quad- and octrees are usually employed in one, two and three dimensions, respectively [14, 41, 140]. Tree data structures are also often the starting point for efficient massively parallel simulations on very large architectures [5, 14, 29, 41]. The quadtree concept shown in Figure 6.13 illustrates the analogy between an adaptive hierarchical quadrilateral mesh and the two-dimensional tree. In our implementation of hierarchical B-spline refinement, the tree is the fundamental entity, where each node or leaf holds all the information of the corresponding knot span on the respective hierarchical level and of the basis functions defined therein. Additionally, we equip each node or leaf with pointers that connect it with all direct neighbors of the same hierarchical level (see Figure 6.13). These "horizontal" neighboring relations are frequently needed during refinement to establish contracted basis functions, check for linear dependencies and correctly assign degrees of freedom.

6.3.3.2 Evaluation of basis functions

Hierarchical B-splines N^k of knot span width $h/2^k$ are generated by contraction of unrefined B-splines. Thus, they do not have to be implemented separately, but can be directly computed on each overlay level k in each parametric direction ℓ from their original unrefined counterparts N^0 as follows

$$N^k(\xi^\ell) = N^0(2^k\xi^\ell) \tag{6.18a}$$

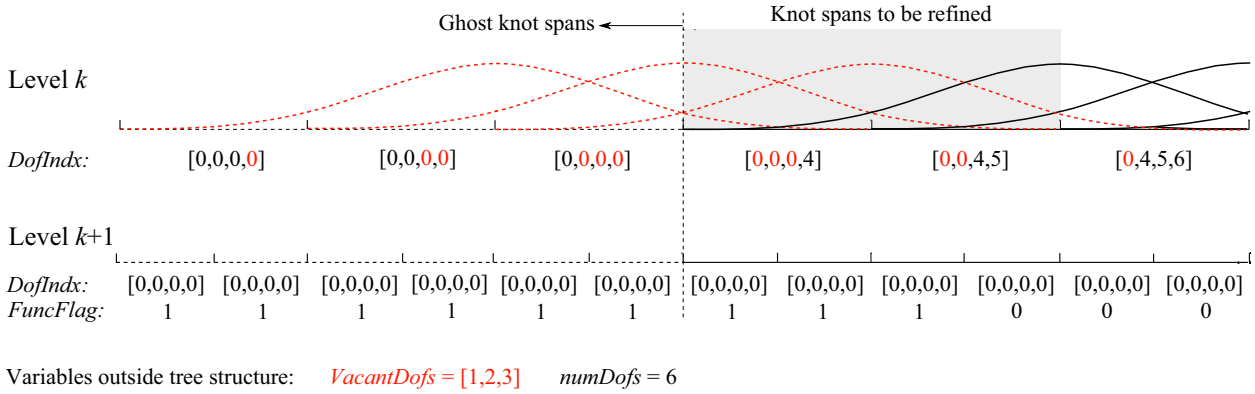


Figure 6.15: Illustration of Algorithm 2: Identify linear dependencies, remove corresponding degrees of freedom and store them in VacantDofs for later reassignment.

$$\frac{\partial N^k(\xi^\ell)}{\partial \xi^\ell} = \frac{1}{2^k} \frac{\partial N^0(2^k \xi^\ell)}{\partial \xi^\ell} \tag{6.18b}$$

According to their tensor-product structure, multidimensional B-splines and their first derivatives can be subsequently assembled by simple multiplication of their components from each parametric direction.

6.3.3.3 Degree of freedom organization

Handling degrees of freedom within the hierarchical tree structure is complicated by the removal of basis functions during the refinement process and therefore requires special care. The degrees of freedom attributed to the basis functions with support in a knot span element are contained in an array structure, denoted as *DofIndx* here, and stored in the corresponding node or leaf of the tree. In particular, the last entry of *DofIndx* corresponds to the B-spline whose support starts in the current knot span and continues over the successive $p+1$ knot spans in positive parametric direction. A zero in *DofIndx* indicates that the corresponding basis function does not exist in the basis. This idea can be easily generalized to d_p dimensions, where the support of the B-spline that starts in the current d_p-dimensional knot span continues over a regular polytope spanned by the successive $(p+1)^{d_p}$ knot spans (a line in 1D, a square in 2D, a cube in 3D, etc.). A 1D example of this numbering concept, which is similar to the one devised by HÖLLIG [106], is given in Figure 6.14.

During refinement, contracted B-splines of the next hierarchical level $k+1$ are first initiated via a Boolean, called *FuncFlag* here, to flag the respective leaf, where the new basis function starts. This allows us to check for linear dependencies first, to remove corresponding B-splines of level k by setting their *DofIndx* entries to zero and to buffer their degree of freedom numbers in the array *VacantDofs*. Later on, we can reassign these numbers to basis functions of the new hierarchical level $k+1$, until the buffer *VacantDofs* is empty. In order to carry out the same operations on boundary knot spans, we introduce ghost knot spans [14, 138] that are taken into account only during the refinement procedure, but not during analysis (see Figures 6.14 through 6.16).

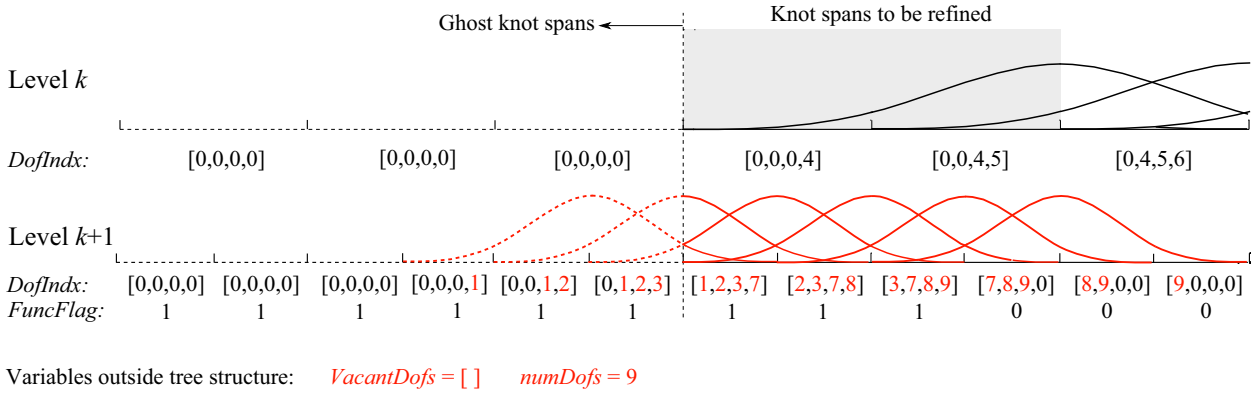


Figure 6.16: Illustration of Algorithm 3: On the basis of FuncFlag, add contracted basis functions of level k+1 that have support outside the ghost knot spans. Assign degrees of freedom by filling DofIndx in positive parameter direction. Consume degrees of freedom buffered in *VacantDofs* first.

6.3.3.4 Basic algorithmic sketch

We roughly outline the basic algorithmic ideas of a single hierarchical refinement step. As input, we assume the tree structure of the current hierarchical level k and the result of an error estimator, which specifies for each leaf of the tree if it is to be refined or not. For simplicity, we consider here only the leaves of the currently finest level k. The refinement procedure consists of three parts. The first part is outlined in pseudocode in Algorithm 1 and illustrated in Figure 6.14 by a small example. It carries out the nucleus operation for each element to be refined, creates new leaves of level k+1 and fills in corresponding contracted basis functions via *FuncFlag*. The second part is given in pseudocode in Algorithm 2 and illustrated in Figure 6.15. It removes linear dependencies between basis functions of level k and k+1. The third part is outlined in pseudocode in Algorithm 3 and illustrated in Figure 6.16. It assigns degrees of freedom for the new basis functions of level k+1. As output, we receive the refined B-spline patch, organized in a tree structure of level k+1.

6.4 Hierarchical refinement of NURBS for arbitrary geometries

Up to this point, we have dealt with B-splines over structured grids for the discretization of axis-aligned cuboidal domains. The concept of subdivision can also be applied to NURBS bases, which discretize arbitrarily shaped domains. A hierarchical refinement scheme for adaptive analysis with NURBS can be derived with minimal effort on the basis of the algorithms and data structures described in Section 6.3.3.

Subdivision rules for univariate NURBS are derived by inserting the two-scale relation of Equation (6.11) into the construction rule for NURBS basis functions, Equation (6.8), which yields the following two-scale relation for NURBS

$$R_{i,p}(\xi) = \frac{w_i \;\; 2^{-p} \sum_{j=0}^{p+1} \binom{p+1}{j} N_p(2\xi - j)}{\sum_{j=1}^{n} w_j N_{j,p}(\xi)} \tag{6.19}$$

According to the isogeometric paradigm [56, 109], the geometry is described exactly by the

Data: Tree data structure, deepest level of leaves is k; result of the error estimator for each knot span element of level k;
Result: Adds new leaves of level k+1 to the tree; initialize index structure *DofIndx*; mark all new leaves with *FuncFlag*, where a new basis function starts;

```
// Loop over all knot span elements of level k (currently deepest level);
// Arguments of for-loops are in the sense of iterators, pointing to leaves in the tree;
for i_ele_k ← 1 to n_ele_k do
    if error estimator requires refinement in i_ele_k then
        // Apply nucleus operation to current i_ele_k;
        // Loop over all elements affected: In case of p=3, these are the current element
        // i_ele_k and its surrounding direct neighbors (see Figure 6.6b);
        // if i_ele_k is located at the boundary, also refine all neighboring ghost
        elements;
        for j_ele ← 1 to n_ele_affected do
            // Create new knot span elements of level k+1;
            if element j_ele is unrefined then
                Append 2/4/8 new leaves of level k+1 in the tree structure
                (bi-, quadri- or octasection of element in 1/2/3D, respectively);
                Initialize in each new leaf DofIndx = [0]; FuncFlag = false;
                Connect neighboring leaves of level k+1 by pointers and update pointers
                in all surrounding other leaves, if existing;
            end
            // Loop over all new leaves of j_ele;
            for i_leaf_k+1 ← 1 to n_leaves do
                // Fill in hierarchical basis functions of level k+1, if required by the
                // nucleus operation (see Figure 6.6b and Section 6.2.2.3);
                if a basis function starts in i_leaf_k+1 then
                    Set i_leaf_k+1.FuncFlag = true;
                end
                // Ensure the proper handling of boundary basis functions;
                if i_leaf_k+1 contains a ghost element then
                    Set i_leaf_k+1.FuncFlag = true;
                end
            end
        end
    end
end
```

Algorithm 1: Find elements and basis functions of the next hierarchical level k+1

original unrefined NURBS basis, so that geometry refinement in the framework of isogeometric analysis is not required. Therefore, we keep the weights w_j and corresponding control points $\boldsymbol{P}_j$ unchanged and always take the sum of the original B-splines $N_{j,p}$ in the denominator of Equation (6.19). Nonetheless, using the refined NURBS basis for enhancing the geometry representation would be of course possible [47, 48].

A multi-dimensional subdivision rule for NURBS can be derived analogously by substituting Equation (6.12) into Equation (6.9)

$$R^{\boldsymbol{h}}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi}) = \frac{w_{\boldsymbol{i}} \sum_{\boldsymbol{j}} \left(\prod_{\ell=1}^{d} 2^{-p_\ell} \binom{p_\ell+1}{j_\ell} N_{p_\ell}(2\xi^\ell - j_\ell) \right)}{\sum_{\boldsymbol{j}} w_{\boldsymbol{j}} B_{\boldsymbol{j},\boldsymbol{p}}(\boldsymbol{\xi})} \tag{6.20}$$

```
Data: Tree data structure, deepest level of leaves is k+1;
Result: Degrees of freedom of linearly dependent basis functions of level k are erased
        from DofIndx in the tree and stored in VacantDofs for later reassignment;

// Loop again over all knot span elements of level k;
for i_ele_k ← 1 to n_ele_k do
    if i_ele_k has leaves of level k+1 then
        // Implementation sketch of conditional clauses:
        // - start from the leaves of level k+1 of the current element i_ele_k;
        // - regular polytope at issue consists of (p+2)^d successive leaves of level k+1;
        // - it is spanned by (p+2) leaves of level k+1 in d positive parametric directions;
        // - walk through the polytope using pointers that connect neighboring leaves;
        // - determine existence of neighboring element by checking corresponding
        pointer;
        // Note that due to the ghost element concept (see Figures 6.14 through 6.16),
        this
        // strategy also works for boundary basis functions;
        if at least one of the leaves of the polytope does not exist then break;
        if FuncFlag == false in at least one of the leaves of the polytope then break;
        // Since in each leaf of the polytope, a basis function of level k+1 starts, the
        // basis function of level k starting in i_ele_k has to be removed and its degree
        // of freedom needs to be remembered for reassignment later on;
        Identify degree of freedom dof that starts in i_leaf_k and append to VacantDofs
        // Loop over polytope spanned by (p+1)^d leaves of level k and erase dof;
        for j_ele_k ← 1 to n_ele_polytope do
            Set j_ele_k.DofIndx ( position of dof ) = 0;
        end
    end
end
```

Algorithm 2: Remove linear dependencies between current level k and next level $k+1$

where the multi-index notation exactly follows the one introduced in Section 6.1.2 for multivariate B-splines and Section 6.2.1 for multivariate B-spline subdivision. Since the geometry is not refined, B-splines $B_{\boldsymbol{j},\boldsymbol{p}}$ and corresponding weights in the denominator of Equation (6.20) refer again always to the original B-spline patch.

On the basis of Section 6.4.1, we introduce some adaptations to the hierarchical refinement scheme for B-splines to also accommodate NURBS. First, we separate Equations (6.19) and (6.20) in a B-spline part (numerator) and a rational part (denominator), which are treated separately. The numerator carries out hierarchical refinement on B-spline level, so that we can make full use of the concepts and algorithms introduced in Sections 6.2 and 6.3. Thus, the resulting hierarchically refined NURBS basis is also constructed from a nested sequence of bisected knot span elements, over which multiple hierarchical overlay levels of repeatedly contracted B-splines are defined.

As shown in the previous section, the denominator is always computed with the original

Data: Tree data structure, deepest level of leaves is k+1; list of unassigned degrees of freedom in *VacantDofs*; total number of degrees of freedom *numDofs*;
Result: In leaves of level *k+1* with *FuncFlag*==true, a degree of freedom is assigned to the basis function starting there;

```
// Loop over all new knot span elements of level k+1;
for i_leaf_k+1 ← 1 to n_leaves_k+1 do
    // Prevent dof assignment to basis functions that consist of ghost knot spans only;
    // Consider polytope spanned by (p+1)^d leaves of level k+1;
    if all leaves of the polytope are ghost knot spans then break;
    else if i_leaf_k+1.FuncFlag == true then
        if VacantDofs is empty then
            // Loop over polytope spanned by (p+1)^d leaves of level k+1 and assign new dof;
            for j_leaf_k ← 1 to n_polytope do
                Set i_leaf_k+1.DofIndx ( position of numDofs ) = numDofs;
            end
            numDofs++;
        else
            // Assign degree of freedom from VacantDofs;
            for j_leaf_k ← 1 to n_polytope do
                Set i_leaf_k+1.DofIndx ( position of numDofs ) = VacantDofs.First();
            end
            Erase first entry of VacantDofs;
        end
    end
end
```

Algorithm 3: Activate new basis functions

B-spline patch B^0

$$\mathrm{sum}(\boldsymbol{\xi}) = \sum_{\boldsymbol{j}} w_{\boldsymbol{j}} B^0_{\boldsymbol{j}}(\boldsymbol{\xi}) \tag{6.21}$$

where multi-index $\boldsymbol{j}$ includes all B-splines with support at the parameter space location $\boldsymbol{\xi}$. The basis functions $R_{\boldsymbol{i}}$ of the hierarchical NURBS basis and its derivatives follow as

$$R_{\boldsymbol{i}}(\boldsymbol{\xi}) = \frac{B_{\boldsymbol{i}}(\boldsymbol{\xi})}{\mathrm{sum}(\boldsymbol{\xi})} \tag{6.22a}$$

$$\frac{\partial R_{\boldsymbol{i}}(\boldsymbol{\xi})}{\partial \xi^\ell} = \frac{\partial B_{\boldsymbol{i}}(\boldsymbol{\xi})/\partial \xi^\ell \, \mathrm{sum}(\boldsymbol{\xi}) - B_{\boldsymbol{i}}(\boldsymbol{\xi}) \, \partial \mathrm{sum}(\boldsymbol{\xi})/\partial \xi^\ell}{\mathrm{sum}(\boldsymbol{\xi})^2} \tag{6.22b}$$

where we additionally drop the weights $w_{\boldsymbol{i}}$ in the numerator of Equation (6.22a) for further simplification. Standard rules for generating higher order derivatives [154] can be adapted in the same way. The geometry mapping by way of the Jacobian matrix and determinant are computed throughout the hierarchical refinement procedure from the unrefined NURBS patch

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{\boldsymbol{j}} \frac{w_{\boldsymbol{j}} B^0_{\boldsymbol{j}}(\boldsymbol{\xi})}{\mathrm{sum}(\boldsymbol{\xi})} \boldsymbol{P}_{\boldsymbol{j}} \tag{6.23}$$

with the initial set of weights $w_{\boldsymbol{j}}$ and control points $\boldsymbol{P}_{\boldsymbol{j}}$.

In the following two chapters, we will illustrate how the concepts and algorithms for local hierarchical refinement of B-splines and NURBS can be integrated into the finite cell method and isogeometric analysis to derive efficient adaptive schemes.

Chapter 7

Adaptive analysis based on local hierarchical refinement of NURBS

In the following, the versatility of local hierarchical refinement in the framework of NURBS based isogeometric analysis is demonstrated for a series of fluid and solid mechanics problems in two and three dimensions. B-spline and NURBS basis functions exhibiting higher order continuity have been shown to be ideal candidates for approximating these problems in the framework of the finite element method, for example in the context of structural vibrations [55, 56], incompressibility [9, 77, 199], shells [27, 75, 117], fluid-structure interaction [20, 23], turbulence [6, 18, 21], phase-field analysis [35, 96], contact [200, 201], fracture [206] and optimization [60, 211]. Therefore, we will not discuss their general solution characteristics, but directly concentrate on their hierarchical refinement.

Using the concepts and algorithms introduced in Section 7 in conjunction with a simple gradient based error estimator, B-spline and NUBRS discretizations are adaptively refined by overlays of hierarchically contracted basis functions. According to the isogeometric paradigm that assumes that the initial geometry representation is already exact [54, 56, 183], the initial NURBS patch is always used for geometry related operations throughout the refinement process. Our in-house isogeometric code has been set from the scratch by the author is an adaptation of the finite cell implementation outlined in Section 5.4.3 [177]. In the present section, all examples are discretized by cubic B-splines or NURBS and the corresponding system of equations is solved by an iterative GMRES solver with ILU preconditioning [134] provided by the library framework Trilinos [203].

7.1 Error estimation and automatic refinement

In the following, elements to be refined are selected automatically by means of a suitable error indicator ε. All meshes shown in the following are generated automatically with the help of a simple gradient based error estimator, which aims at capturing steep gradients in the solution

$$\varepsilon_e = \frac{1}{V_e} \left(\int_{\Omega_e} |\nabla \boldsymbol{u}|^2 \ \ \mathrm{d}\Omega_e \right)^{1/2} \tag{7.1}$$

where $\boldsymbol{u}$ is the solution field evaluated from the current mesh of hierarchical level k. The

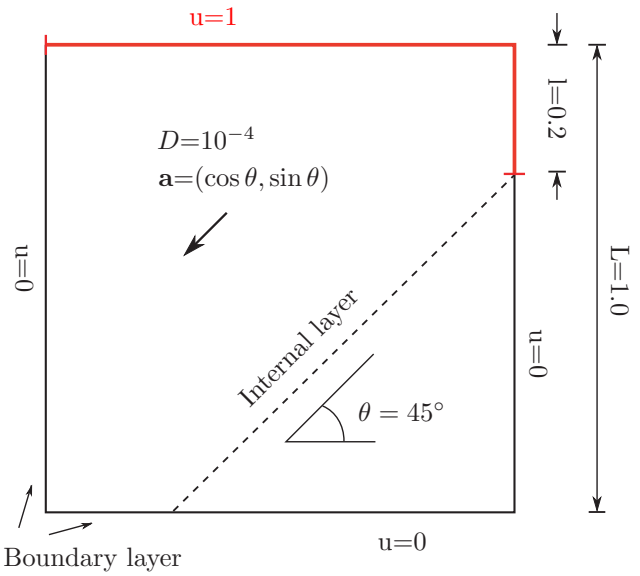


Figure 7.1: Advection skew to the mesh in 2D: Problem definition.

indicator is evaluated over the domain Ω_e of each element and subsequently normalized with respect to the corresponding element volume V_e. If ε_e is larger than its average

$$\varepsilon_e > \frac{C}{n_{ele}} \sum_{i=1}^{n_{ele}} \varepsilon_e^i \tag{7.2}$$

with n_{ele} being the number of all elements in the current mesh, the element is refined. We introduce an additional constant C to empirically fine-tune the threshold for the specific problem. For more elaborate error indicators and estimators, see for example [4].

7.2 Adaptive isogeometric analysis based on cuboidal geometries

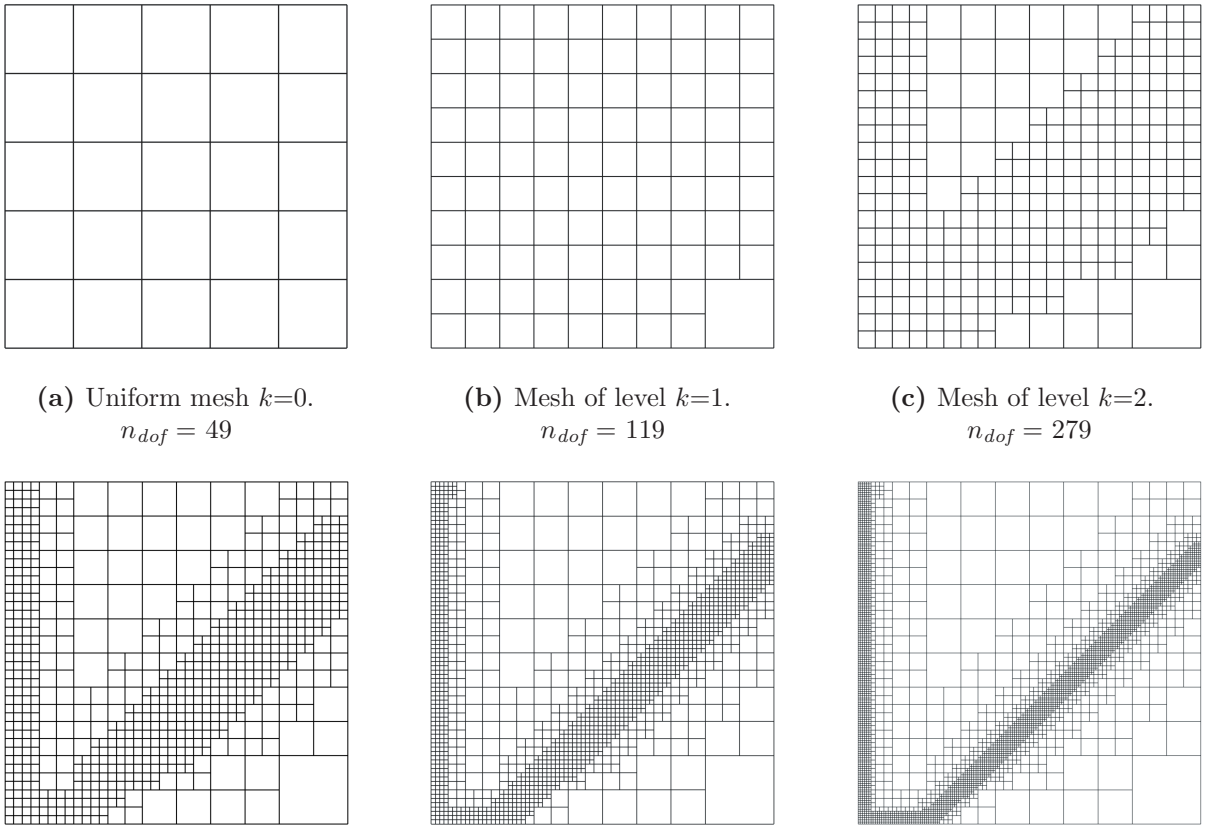
We will first focus on problems over cuboidal axis-aligned domains, which can be discretized exactly by B-splines defined over a structured grid of knot span elements. As outlined in Section 7.3.2, this allows for a particularly simple geometry handling.

7.2.1 Advection skew to the mesh in 2D

The first model problem is illustrated in Figure 7.1a and involves the solution of the linear advection-diffusion equation

$$\boldsymbol{a} \cdot u - \nabla u \cdot (D \nabla u) = f \tag{7.3}$$

where u denotes the solution, $\boldsymbol{a}$ is the velocity, D is the diffusivity and f is a source term. In particular, the velocity is inclined to the mesh at 45° and the diffusivity is chosen to be extremely small, so that the problem is dominated by advection, resulting in a very high global Peclet number of Pe=10^4. Thus, we expect sharp interior and boundary layers, which require



(a) Uniform mesh k=0. $n_{dof} = 49$

(b) Mesh of level k=1. $n_{dof} = 119$

(c) Mesh of level k=2. $n_{dof} = 279$

(d) Mesh of level k=3. $n_{dof} = 612$

(e) Mesh of level k=4. $n_{dof} = 1{,}319$

(f) Mesh of level k=5. $n_{dof} = 3{,}017$

Figure 7.2: Sequence of hierarchical meshes generated by an automatic refinement scheme that makes use of a simple gradient-based error indicator and the algorithms described in sub-section 4.4.

stable numerical techniques in addition to increased resolution to be accurately captured. The problem is a well-studied benchmark [19, 36], examined for uniform k-refinement in [56] and local T-spline refinement in [22, 69].

We investigate the adaptive resolution of the internal and boundary layers with the present hierarchical refinement approach, starting from a 5×5 grid of cubic B-splines. We satisfy boundary conditions weakly at the inflow boundary with Nitsche's method [78, 80, 99] and strongly at the outflow boundaries [19]. Furthermore, we apply standard SUPG stabilization [19, 36, 67] in addition to refinement. The corresponding stabilization parameter is chosen to be $\tau = h_a/(2\,|\boldsymbol{a}|)$, where h_a is the element length of in flow direction. In the current example, $h_a = \sqrt{2}\ 2^{-k}h$, where k is the finest level of hierarchical refinement present in the element under consideration and h is the original knot span spacing of the unrefined discretization. Employing an automatic refinement scheme based on the error indicator of Equation (7.1) and the algorithms described in Section 6.3, we obtain a sequence of hierarchical meshes shown in Figure 7.2. The hierarchy of knot spans defining B-splines of each level k is shown in Figure 7.3. It can be observed that the refinement captures the location of the internal as well as the boundary layers very well.

An overkill solution obtained with a uniform cubic B-spline discretization of 160×160 elements and SUPG stabilization and the adaptive solution obtained with a refined mesh of

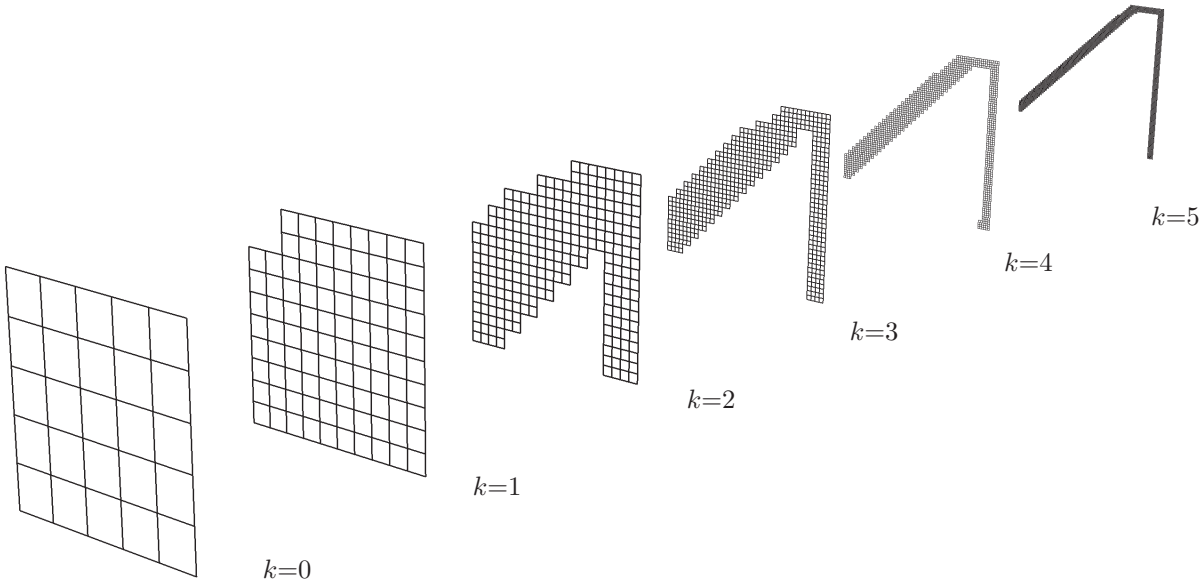
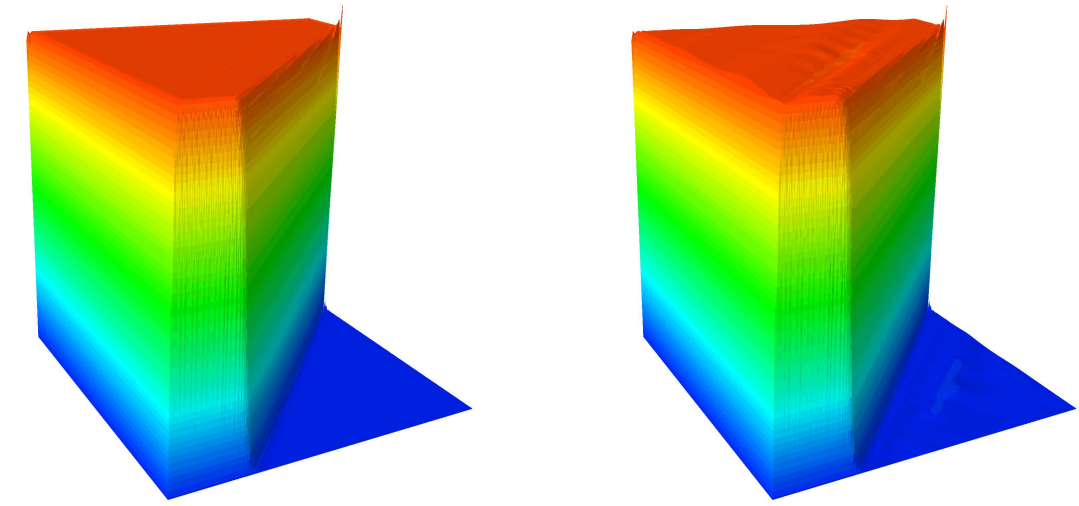


Figure 7.3: Sequence of knot span elements for each hierarchical level k, corresponding to the hierarchical meshes for the 2D advection problem.

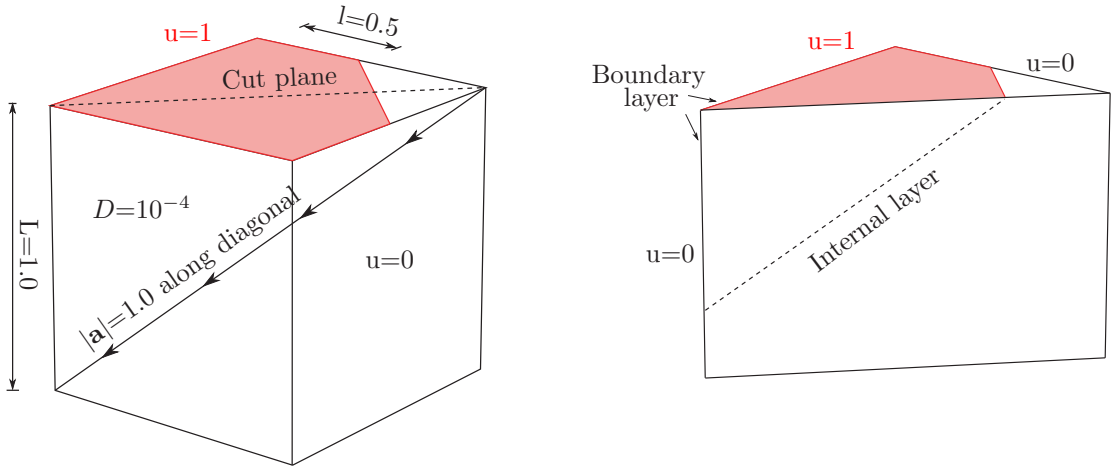


(a) Solution obtained with a uniform discretization of 160×160 cubic elements.

(b) Solution obtained with the adaptive hierarchical mesh of k=5.

Figure 7.4: Solution of the advection dominated problem in 2D.

hierarchical level $k = 5$ are shown in Figure 7.4. We can observe over- and under-shooting of the adaptive solution along the internal layer as also reported in [22, 69] for T-spline refinement. Nonetheless, the comparison of adaptive and uniform solution fields demonstrates that hierarchical refinement leads to a qualitatively similar result. While both meshes provide a comparable element size around the location of the layers, the adaptive mesh with 3,017 dofs requires only about 12% of the degrees of freedom of the uniform mesh with 26,244 dofs. Finally, we would like to point out the high quality of the refinement in terms of locality, as the hierarchical elements of the finest level show no propagation through the mesh. This is an



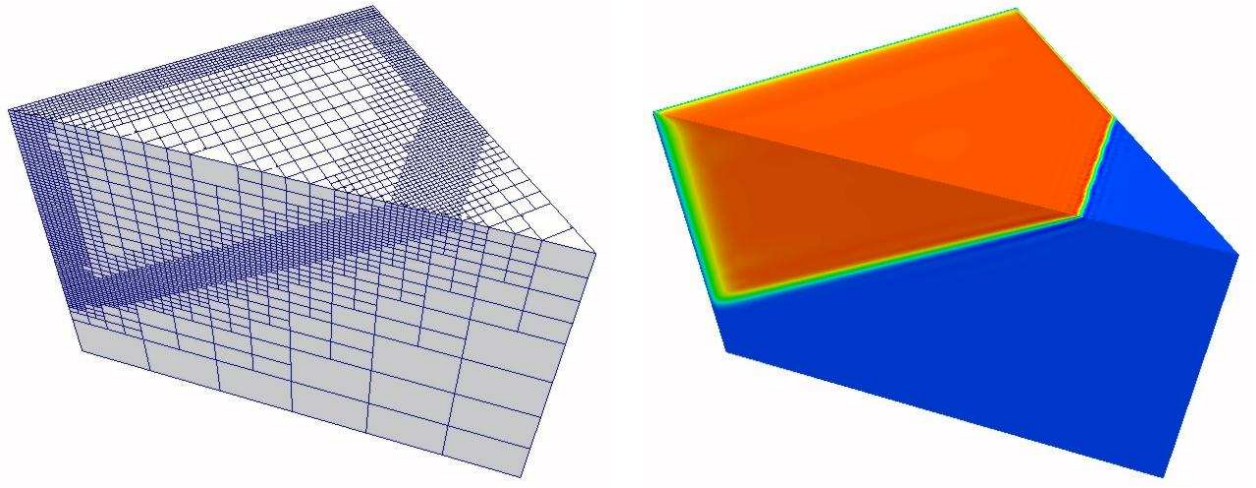
(a) Problem definition. **(b)** System cut along diagonal plane.

Figure 7.5: Advection skew to the mesh in 3D.

improvement in comparison to recent works on adaptive isogeometric analysis [22, 69], where automatic local refinement employing cubic T-splines was reported to result in globally refined T-meshes when applied to a very similar advection-diffusion problem.

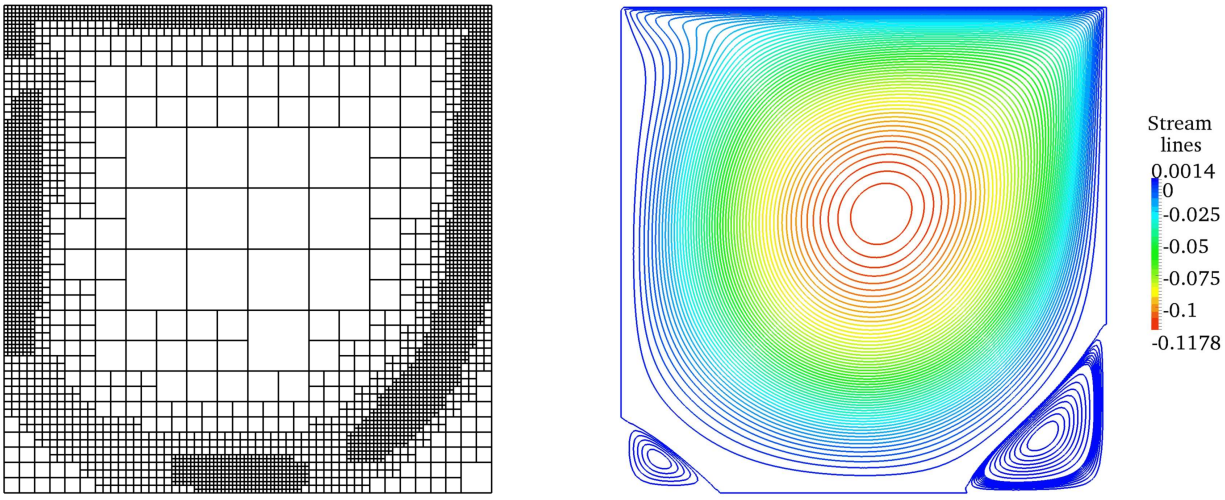
7.2.2 Advection skew to the mesh in 3D

Having in mind the 2D problem of Figure 7.1, we can construct an analogous advection-diffusion problem in three dimensions, whose details are given in Figure 7.5a. It exhibits the same challenges in terms of strong advection dominance (global Péclet number κ=10^4), a sharp internal layer and several boundary layers (see Figure 7.5b). Following the previous sub-section, we apply SUPG stabilization, the error indicator of Equation (7.1) and weak and



(a) Hierarchical mesh of level k=4. **(b)** Accurate approximation of sharp layers.

Figure 7.6: Adaptive mesh and corresponding solution for the 3D advection problem. To reveal internal features, only one half of the domain is shown according to Figure 7.6b.



(a) Mesh of level k=4. **(b)** Isolines of the stream function Ψ.

Figure 7.7: The lid driven cavity problem: Adaptive hierarchical mesh and corresponding solution.

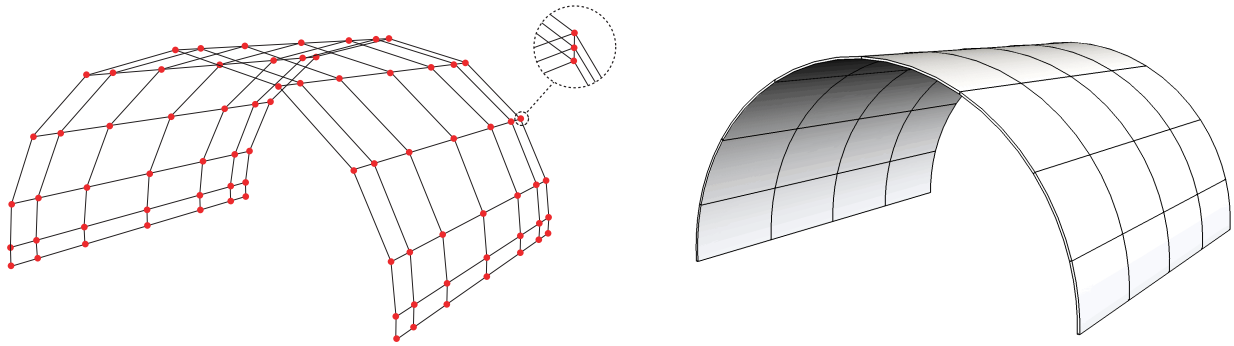
strong boundary conditions at inflow and outflow boundaries, respectively. Figure 7.6 shows the resulting hierarchical mesh of level k=4, generated automatically from an initial 6×6×6 cubic B-spline grid, and the corresponding solution field. The mesh captures the location of internal and boundary layers accurately and its refinement is local without propagating throughout the entire domain. Similar to the 2D case, the solution field exhibits slight under- and overshooting along the internal layer. While the adaptive mesh of Figure 7.6a features only 116,314 degrees of freedom, a uniform cubic B-spline discretization, which uses a global mesh size corresponding to the finest elements of the adaptive mesh, requires a resolution of 96×96×96 knot span elements, which amounts to 941,192 degrees of freedom. In Section 4, we demonstrated that our hierarchical refinement approach can be generalized easily to three dimensions from algorithmic and implementation points of view. From the point of view of computational efficiency, the present result confirms that hierarchical refinement in 3D also yields high quality adaptive meshes and accurate solution fields, comparable to the 2D case of the previous sub-section.

7.2.3 Lid driven cavity

To demonstrate the versatility of the hierarchical refinement approach in the framework of a further physical model, we consider the two-dimensional stationary incompressible Navier-Stokes equation in stream function formulation [44]

$$-\nu\Delta^2\Psi + \Psi_y\left(\Delta\Psi\right)_x - \Psi_x\left(\Delta\Psi\right)_y = f \tag{7.4}$$

where Ψ denotes the stream function, ν the viscosity, and f the applied body force. A standard benchmark is the lid driven cavity problem [44, 93], which models flow in a unit square domain $\Omega = (0,1)^2$, whose upper boundary moves to the right, whereas the rest of the boundaries are fixed. Corresponding boundary conditions are Ψ=0 on all boundaries, Ψ_x=0 and Ψ_y=0 on the left, right and bottom boundaries ('no slip' requirement), Ψ_x=0 and Ψ_x=1 on the top boundary (the driven surface). At all boundaries, we impose Ψ strongly, while its derivatives



(a) Control mesh. **(b)** NURBS mesh.

Figure 7.8: Geometry description of half a cylinder. The structure is modeled as a continuum with cubic NURBS in surface directions and 3 quadratic NURBS basis functions through the thickness.

are imposed weakly by a variant of Nitsche's method [19, 99]. The nonlinearities in Equation (7.4) are handled by simple fixed point iteration [194].

The solution characteristics of the driven cavity problem are strongly determined by the choice of viscosity. We set ν=1.25 $\cdot 10^{-3}$, which results in a Reynolds number of Re=800. A separation of vortices in the lower edges can thus be expected. Since we are also interested in accurately tracking the location of separation at Ψ=0, we add an additional term to the gradient-based error indicator of Equation (7.1) as follows

$$\varepsilon_e = \frac{1}{V_e}\left[\left(\int_{\Omega_e} |\nabla\Psi|^2 \, \mathrm{d}\Omega_e\right)^{1/2} + \left(\int_{\Omega_e} |\Psi|^2 \, \mathrm{d}\Omega_e\right)^{-1/2}\right] \tag{7.5}$$

so that ε_e also grows in the vicinity of Ψ=0. The resulting hierarchical mesh of level k=4, generated from an initial cubic B-spline discretization of 8×8 knot span elements, and the isolines of the corresponding stream function solution are plotted in Figure 7.7. They show that the refinement can be concentrated around the boundary layers and locations of flow separation. The adaptive mesh of Figure 7.7a with 3,517 dofs requires about 80% fewer degrees of freedom than a corresponding uniform discretization with 16,641 that uses the size of the smallest element of Figure 7.7a as the global mesh size.

7.3 Adaptive isogeometric analysis based on NURBS geometries

According to the isogeometric paradigm, the NURBS basis of the geometry description should be also used for analysis, thus superseding geometry approximation and mesh generation of standard FEM [56, 183]. However, in most cases, the complexity in geometry and in the solution of the physical model do not coincide, so that the NURBS basis can represent the geometry exactly with a limited number of basis functions, while analysis requires local refinement to achieve satisfactory accuracy. The hierarchical refinement approach for NURBS based isogeometric analysis introduced in Section 5 maintains the initial geometry description by a given set of control points, weights and NURBS basis functions throughout the refinement

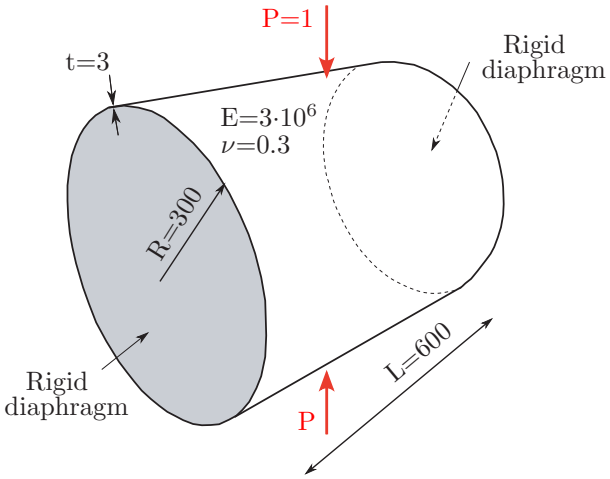


Figure 7.9: The pinched cylinder.

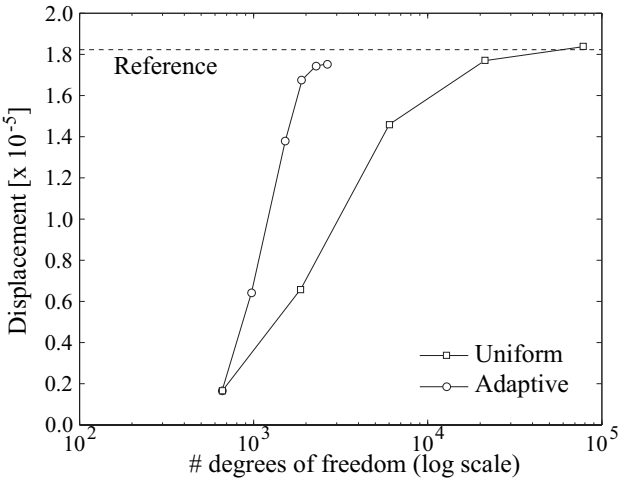
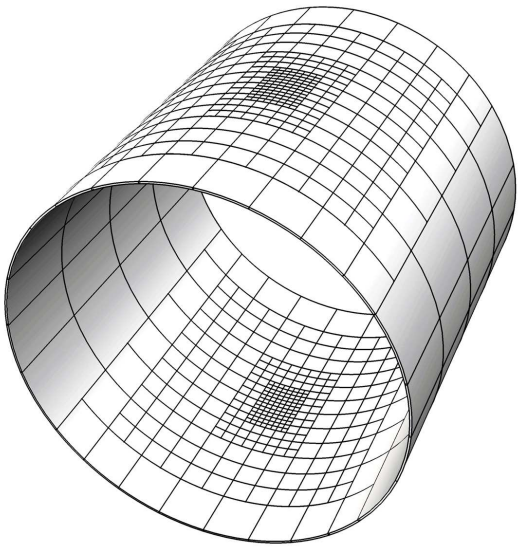
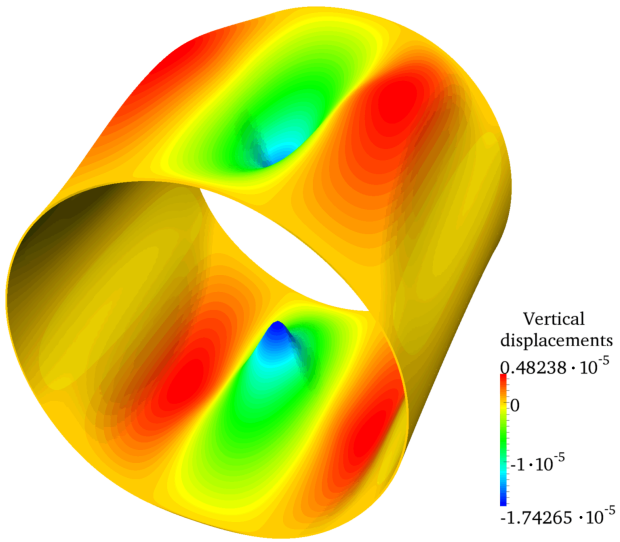


Figure 7.10: Convergence of the vertical displacement under the load.



(a) Hierarchical mesh of level k=4.



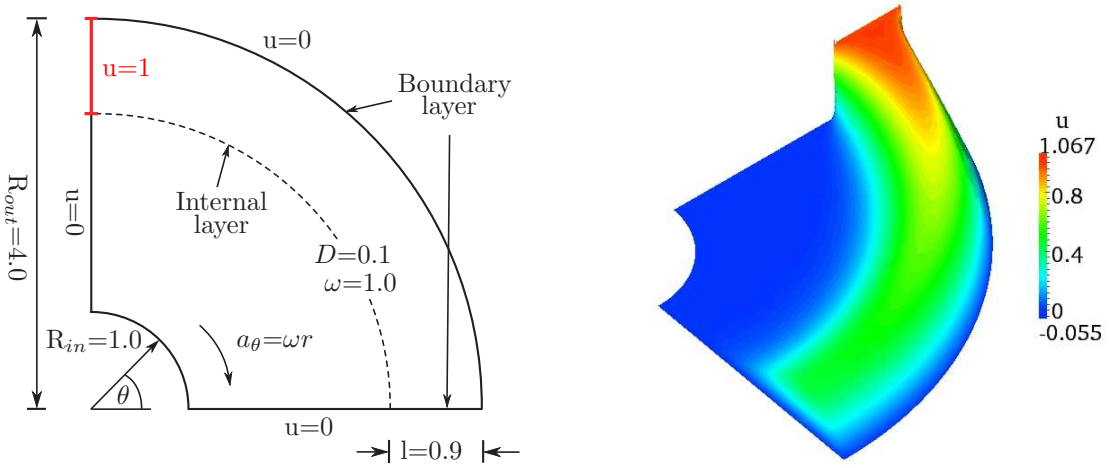
(b) Corresponding vertical displacements plotted on the deformed structure.

Figure 7.11: Adaptive mesh and solution for the pinched cylinder problem. The results for the complete structure are obtained by mirroring the results obtained for the half cylinder.

process, while introducing additional levels of hierarchical basis functions, which achieve an adaptive refinement of the solution fields only.

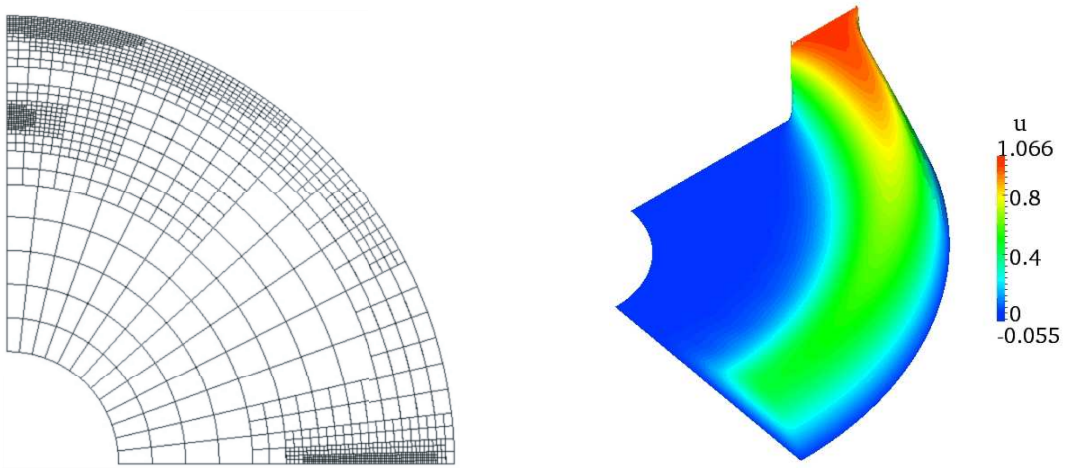
7.3.1 The pinched cylindrical shell

The first NURBS example is the pinched cylinder problem from the shell obstacle course. It constitutes a standard benchmark with great relevance to the assessment of shell analysis



(a) System sketch. **(b)** Solution from uniform discretization.

Figure 7.12: Advection-diffusion in an annular section.



(a) Hierarchical mesh of level k=4. **(b)** Corresponding solution.

Figure 7.13: Adaptive mesh and solution for the 2D advection-diffusion problem.

procedures and has therefore been discussed extensively in the literature [25, 56, 52, 131]. Its geometry is given here by the control mesh and the corresponding mapped NURBS mesh of Figure 7.8, modeling one half of the cylinder. All geometrical mapping procedures throughout the adaptive analysis will revert to this model, which will not be refined, since it represents the geometry exactly. Following [56, 161], the shell is modeled as a three-dimensional solid and no shell assumptions are employed (see Figure 7.8a). The polynomial degree is cubic in the surface directions, whereas only one quadratic NURBS element is used through the thickness, which has been shown to be sufficient to obtain sufficiently accurate results [31, 56]. The problem definition of Figure 7.9 illustrates the external loading by two opposite concentrated forces, which results in highly localized deformation under the loads. At the longitudinal edges of the half cylinder, we apply symmetry boundary conditions. Automatic hierarchical refinement based on the concepts of Section 6.3 and 6.4 with the error indicator of Equation

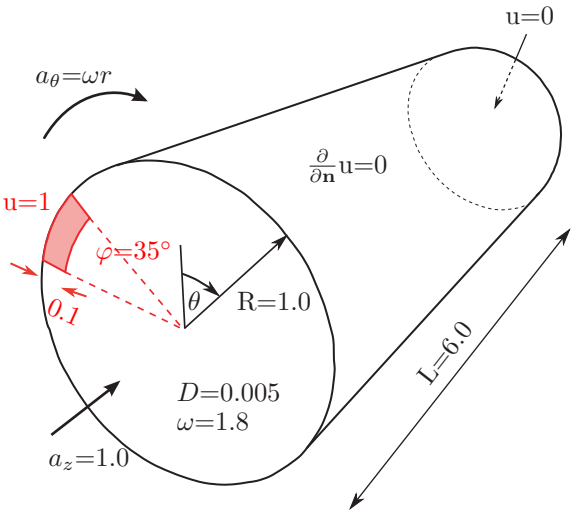


Figure 7.14: Problem sketch.

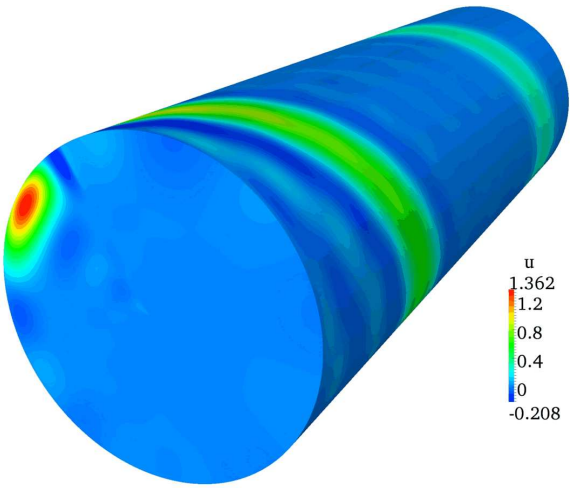
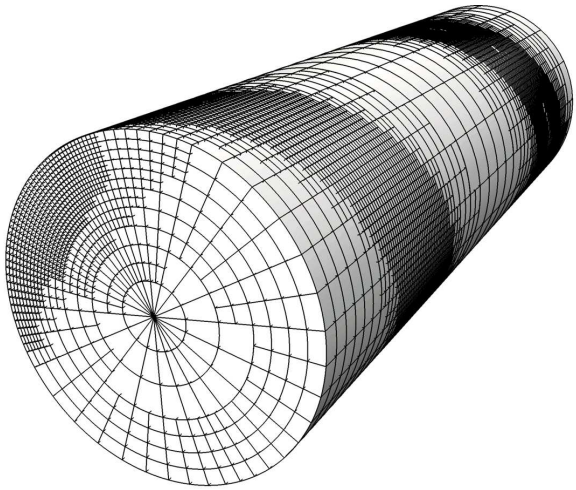
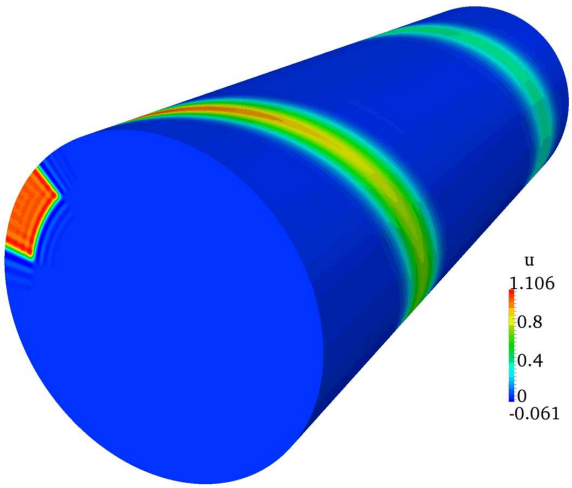


Figure 7.15: Initial solution.



(a) Hierarchical NURBS mesh of level k=3.

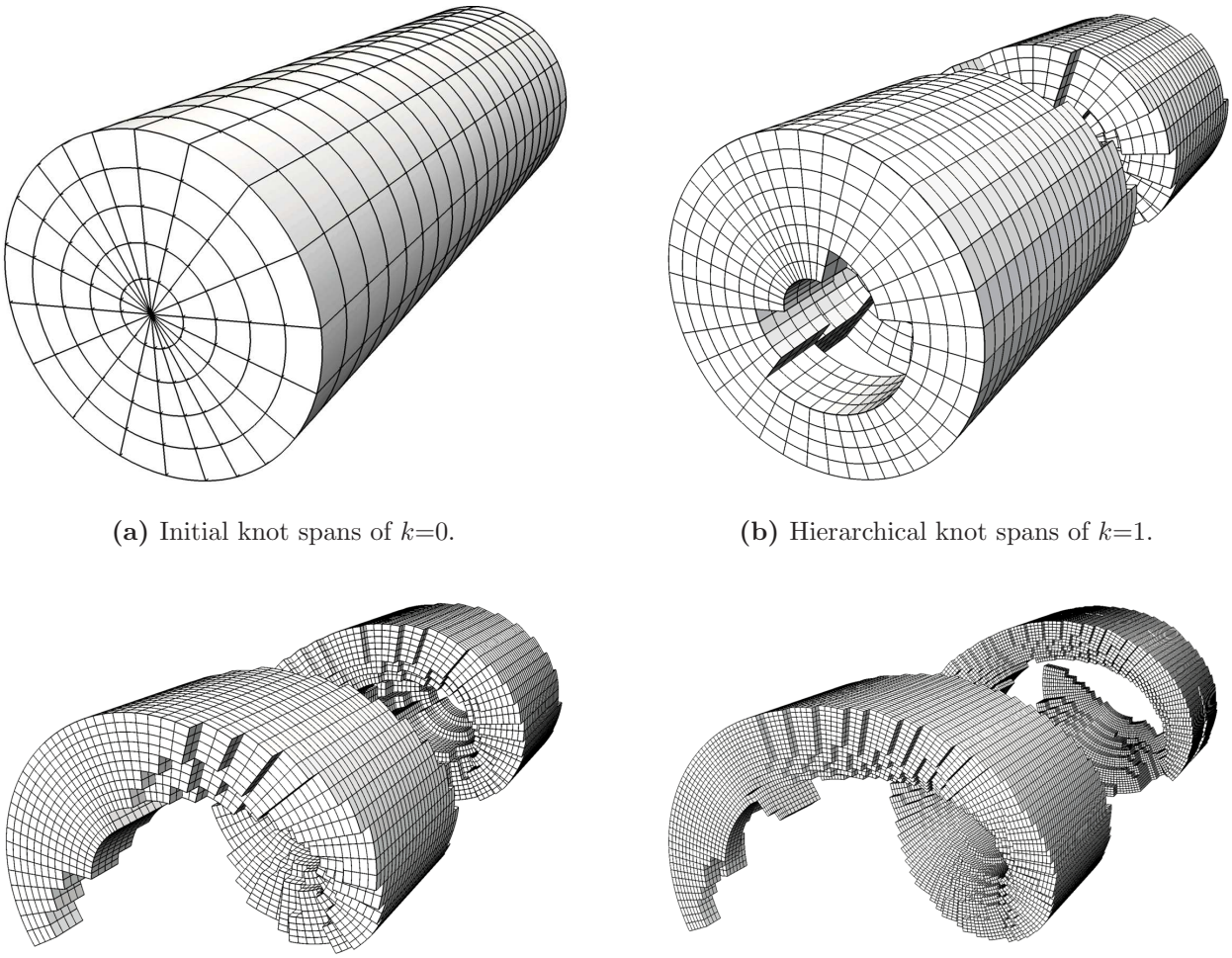


(b) Corresponding solution.

Figure 7.16: Adaptive mesh and solution for the 3D advection-diffusion problem.

(7.1) is applied to the initial discretization of 8×4×1 NURBS elements shown in Figure 7.8b. It carries out local refinement only in the surface directions, but keeps one quadratic NURBS element in the thickness direction, so that the shell is not refined through-the-thickness.

Convergence in terms of vertical displacements under the applied point load is shown in Figure 7.10, for which the analytical solution is given as $1.82488 \cdot 10^{-5}$ [52, 56]. Uniform refinement, which applies global h-refinement in surface directions to the initial discretization of Figure 27b, converges to a slightly softer solution than the reference, which agrees well with previous results [22, 56]. In contrast, adaptive hierarchical refinement converges to a slightly smaller displacement. This can be attributed to the influence of the unrefined parts of the shell, which are not sufficiently resolved, but not detected by the simple gradient-based error indicator due to their relatively small gradients. Nonetheless, hierarchical refinement achieves a comparable level of accuracy with about one order of magnitude fewer degrees of



(a) Initial knot spans of k=0.

(b) Hierarchical knot spans of k=1.

(c) Hierarchical knot spans of k=2.

(d) Hierarchical knot spans of k=3.

Figure 7.17: Sequence of knot span elements, over which contracted basis functions of successive hierarchical levels k are defined. The combination of all levels results in the hierarchical NURBS mesh of Figure 7.16a.

freedom than uniform refinement. The hierarchical mesh of level k=4 and the corresponding displacement plot on the deformed structure are given in Figure 7.11, where deformations are largely magnified for better visibility of the deformation pattern.

7.3.2 Advection-diffusion in an annular section

We focus again on linear advection-diffusion described by Equation (7.3). The present 2D example is defined over an annular quarter section, which is described exactly by a 10×13 NURBS mesh. The two-dimensional flow field corresponds to a rotational vortex with tangential velocity a_θ=ωr and radial velocity a_r=0. A sketch of the problem definition is given in Figure 7.12a. Boundary conditions are prescribed weakly at inflow and strongly at outflow boundaries [19]. Over part of the inflow, the concentration u is set to 1, creating boundary and internal layers, which is illustrated by the reference solution of Figure 7.12b, obtained from a

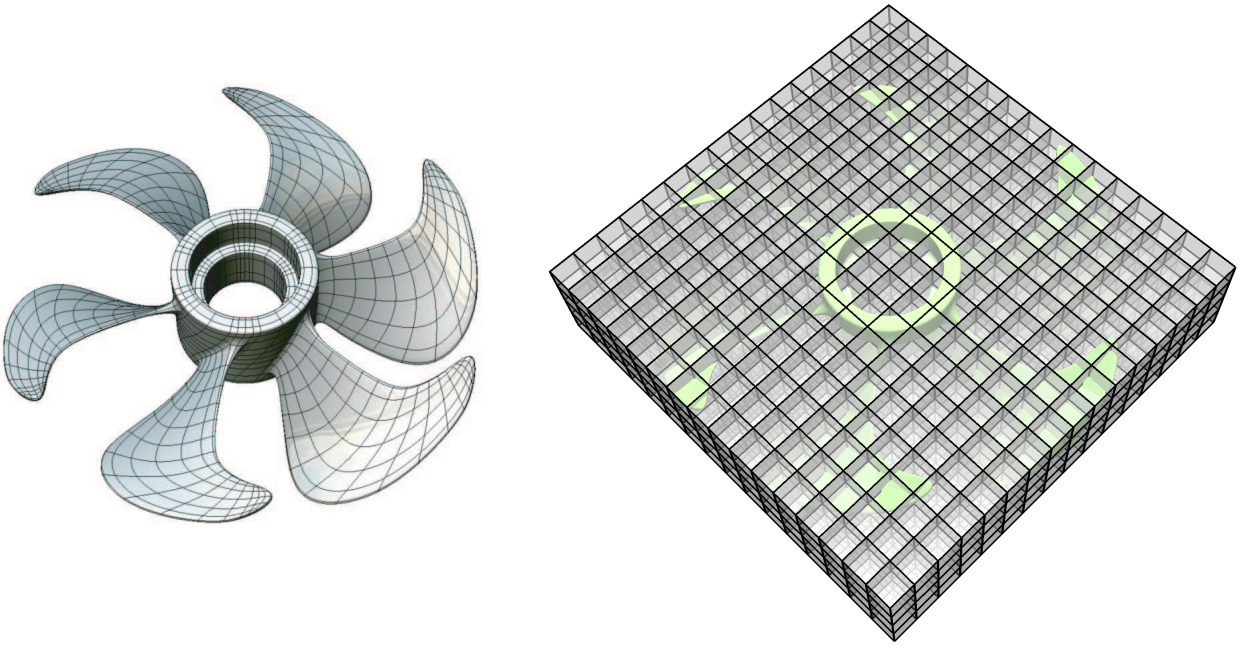
uniform standard Galerkin discretization of 160×208 cubic NURBS elements. Applying automatic hierarchical refinement based on the gradient-based error indicator of Equation (7.1), we obtain an adaptive mesh of level k=4, plotted in Figure 7.13a. The corresponding solution in Figure 7.13b shows qualitatively no difference in terms of boundary and internal layers as well as absolute values in comparison to Figure 7.12b. However, the uniformly refined solution requires 33,810 degrees of freedom, whereas the adaptive mesh requires only 1,387, but both provide the same resolution of the concentration jumps at the boundaries.

7.3.3 Advection-diffusion in a rotating cylinder

Finally, we consider advection-diffusion in a three-dimensional cylinder that rotates around its axis with tangential velocity a_θ=ωr and radial velocity a_r=0. At the same time, we assume flow of constant axial velocity a_z, which results in a helical pattern of the concentration that emerges from the fixed local inflow boundary condition u=1. The geometry of the cylinder is described exactly by two equal NURBS patches, each of which covers one half of the cylinder and consists of 5×10×20 cubic standard Galerkin NURBS elements in (r, θ, z)-directions, respectively (see Figure 7.17a). A sketch of the problem and the initial isogeometric solution are shown in Figure 7.14 and 7.15, respectively. The initial NURBS discretization is unable to accurately resolve the boundary and internal layers along the plume. Automatic hierarchical refinement based on the gradient-based error indicator of Equation (7.1) results in an adaptive mesh of level k=3, plotted in Figure 7.16a. The corresponding solution in Figure 7.16b exhibits a considerably improved resolution of boundary and internal layers. Figure 7.17 illustrates the sequence of knot spans, over which the contracted hierarchical basis functions are defined. The finest level k=3 shown in Figure 7.17d demonstrates that the automatic refinement procedure is able to accurately track the revolving plume. A uniform discretization that yields a plume resolution with the same element size as the smallest element of the adaptive mesh requires a globally refined mesh of 40×80×160 NURBS elements with 1,391,380 degrees of freedom, whereas the present adaptive mesh requires only 125,271 dofs.

7.4 Hierarchical refinement, the finite cell method, and T-spline CAD surfaces

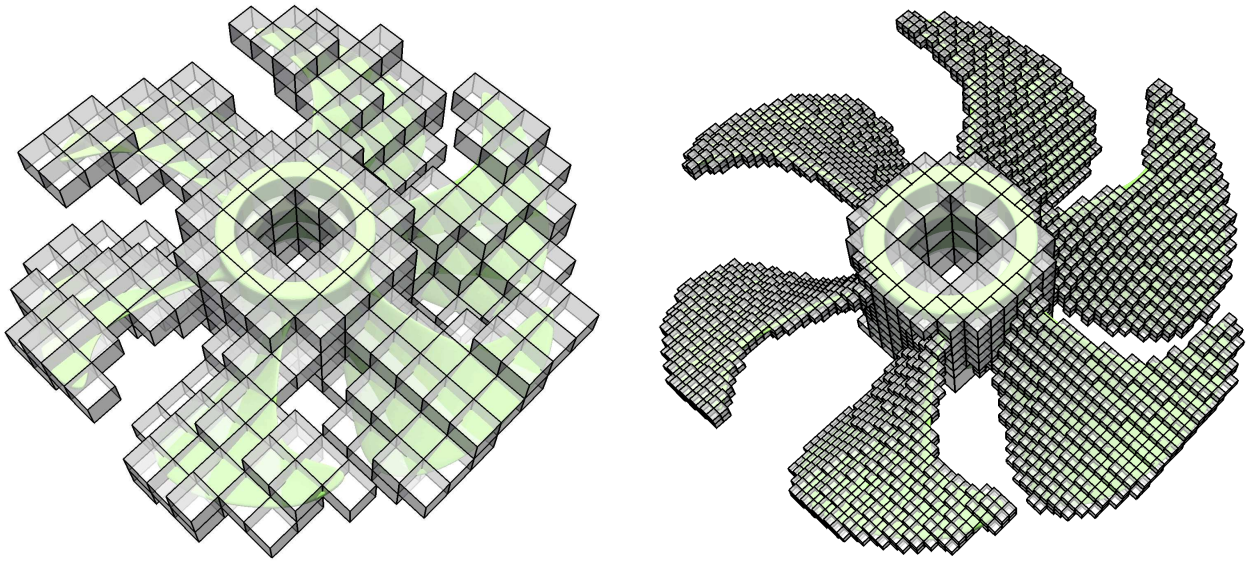
The translation of complex CAD based geometrical models into conforming finite element discretizations is computationally expensive, hard to be fully automated and often leads to error-prone meshes, which have to be improved manually by the user. Combining simplicity and reliability with a high degree of automation, embedded domain methods such as the B-spline version of the finite cell method introduced in Sections 2 to 5 open the door for a seamless isogeometric design-through-analysis procedure for complex engineering parts and assemblies, which bypasses all problems of mesh generation [182]. Its potential for the analysis of complex CAD based geometrical models is illustrated in the following by a a ship propeller and a rim of car wheel, whose geometry is provided by T-spline CAD surfaces. We furthermore demonstrate that hierarchical refinement of B-splines can considerably increase the flexibility of the method by adaptively resolving local features in geometry and solution fields, while thanks to its straightforward implementation, the key benefit of full automation can be maintained.



(a) Bézier elements of a T-spline surface (Output from CAD package Rhino with T-spline plug-in).

(b) The complete structure is immersed in a bounding box of 16×16×4 axis-aligned cubic B-spline elements.

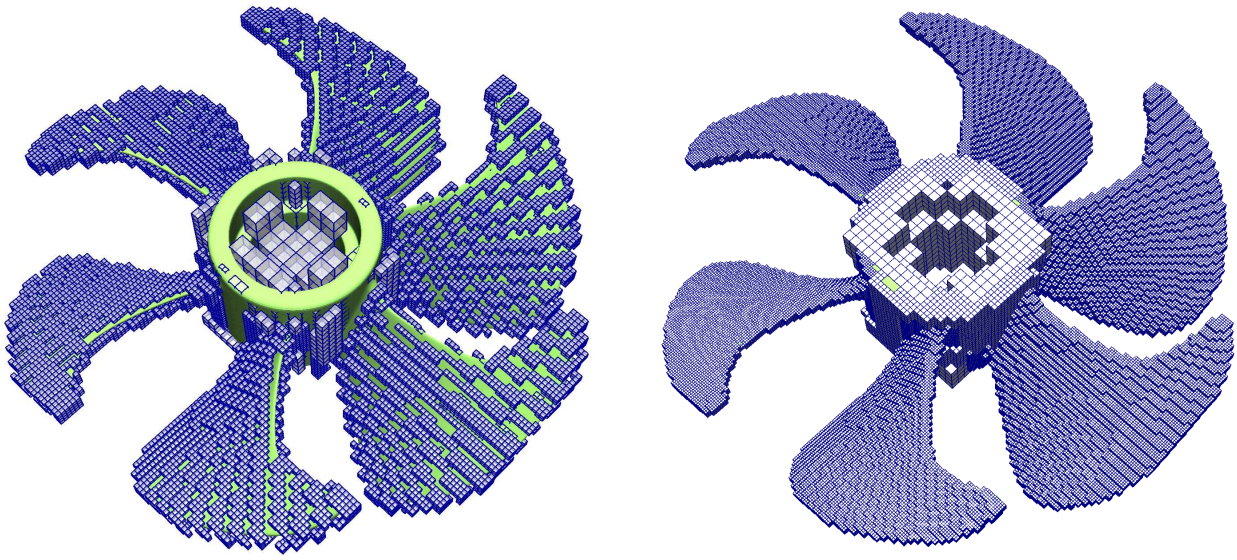
Figure 7.18: Propeller example: CAD geometry description and embedded domain discretization.



(a) Deletion of elements without support in the propeller domain creates a reduced set of elements, which homogeneously resolve the structure irrespective of the local thickness.

(b) Hierarchical refinement of the propeller blades achieves a homogeneous through-the-thickness resolution. Corresponding elements are octasected twice.

Figure 7.19: Propeller example: The role of hierarchical refinement



(a) 18,499 sub-cells of level k=1. **(b)** 226,272 sub-cells of level k=2.

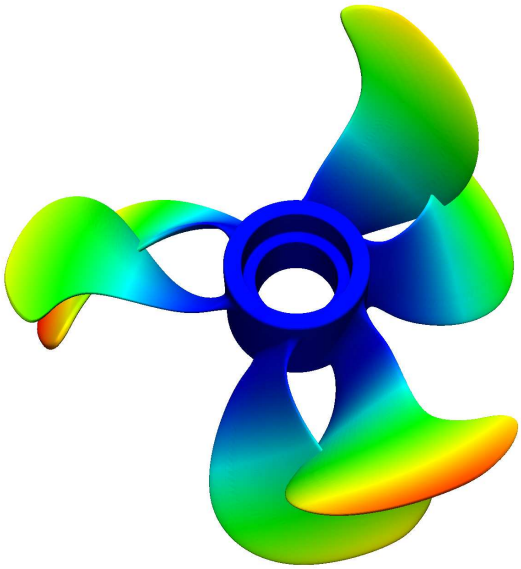
Figure 7.20: Propeller example: Sub-cell partitioning of elements cut by the geometric boundary. The partitioning scheme shown in Figure 2.2 is carried out up to level k=2.

7.4.1 Free vibration analysis of a ship propeller

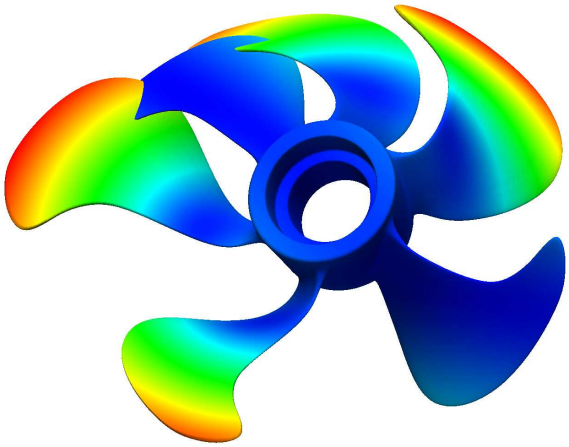
The geometry of the propeller is given by a smooth, watertight T-spline surface (i.e., no gaps or overlaps). It is exported from the CAD package Rhino [166] with the T-spline plug-in [205] in the form of Bézier elements as shown in Figure 7.18a. Its maximum diameter and height is 0.695 m and 0.334 m, respectively, and it is made out of steel with Young's modulus $2.1{\cdot}10^{11}$ N/m^2, Poisson's ratio 0.28 and density 7,850 kg/m^3. The thickness of the cylindrical hub in the center is about four times larger than the average thickness of the surrounding propeller blades. The structure can neither be characterized as a typical shell nor as a true solid. Configurations like this usually require specialized and time consuming meshing procedures to produce good quality discretizations.

We apply the B-spline version of the finite cell method to illustrate the discretization procedure in the framework of the immersed boundary concept and its combination with hierarchical refinement. First, the complete structure is embedded in a regular grid of axis-aligned B-splines as illustrated in Figure 7.18b. In the present examples, we apply uniform B-splines of polynomial degree p=3, which offer higher order approximation, but still are computationally efficient due to their relatively local support. Second, all those knot span elements without support in the propeller domain Ω_{phys} are eliminated from the discretization, which leads to a reduced set of elements displayed in Figure 7.19a. The decision whether an element is to be kept or not is based on a simple point location query, which checks if at least one integration point is located in Ω_{phys}. In particular, this discertization procedure can be fully automated irrespective of the geometric complexity involved.

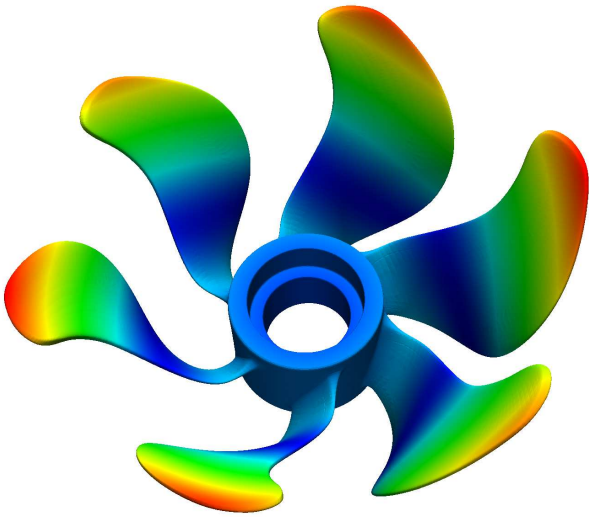
We have successfully tested two options to establish an efficient and reliable point location query procedure within a CAD based geometrical model. A straightforward approach is the voxelization of the embedding domain [38, 57], where a Boolean flag indicates for each voxel whether it is located in- or outside. This transfer of the CAD into a voxel model permits an



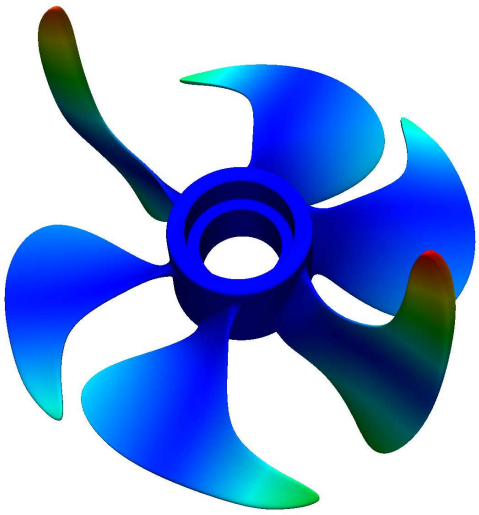
(a) The first, second and third mode shapes exhibit a rotational symmetry around the center, corresponding to the three pairs of opposing propeller blades. We display mode 2.



(b) The fourth and fifth mode shapes exhibit a symmetry with respect to the two axis of the propeller plane. We display mode 5.

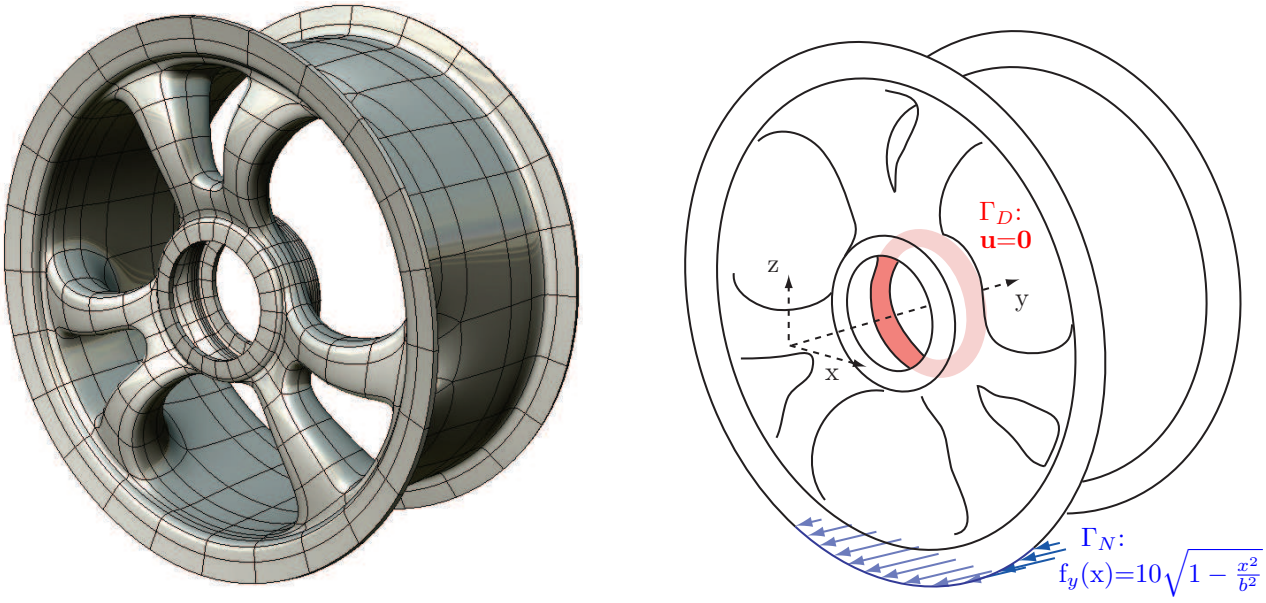


(c) The sixth mode shape is single and shows bending of the blades out of the propeller plane.



(d) The seventh mode shape shows the bending of opposing blades out of the propeller plane.

Figure 7.21: Free vibration analysis of the ship propeller. Most of the activity occurs in the propeller blades, whose resolution has been adaptively increased by hierarchical refinement. The color scale refers to the absolute displacement (dark blue - no displacement, red - highest displacement).



(a) Bézier elements of a T-spline surface (Output from CAD package Rhino with T-spline plug-in).

(b) Horizontal loading on the outer edge of the rim, fixture at the hub of the wheel. (Radius R=239 mm and b=R $\sin \pi/5$)

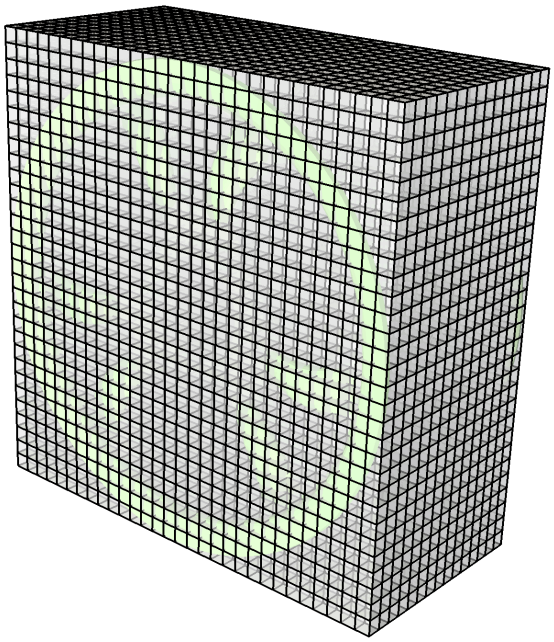
Figure 7.22: Rim of a car wheel: Geometry description and boundary conditions.

analysis in the sense of Section 4.4. A suitable alternative is provided by search algorithms based on special space-partitioning data structures such as k-d trees [30, 58, 171, 172], which are able to directly operate on a surface parameterization. As a proof of concept, we connected our in-house finite cell implementation to the library *VFlib*[7], which provides an efficient k-d tree based point location query for an STL surface parameterization of a CAD model, and successfully generated finite cell discretizations directly based on CAD data.

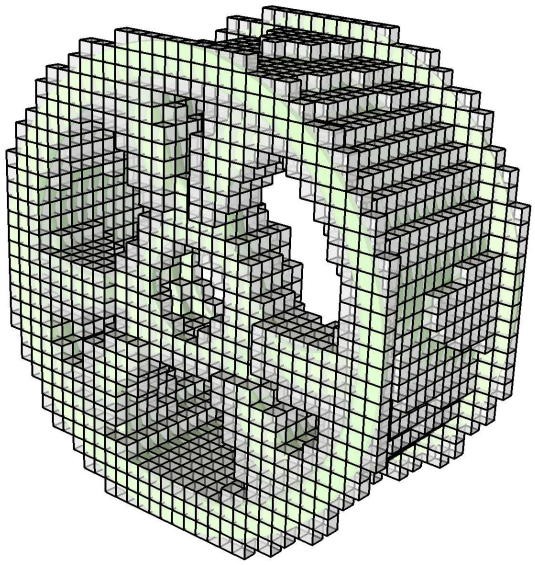
An axis-aligned discretization with elements of the same size does not account for the inhomogeneous thickness of the different regions of the structure. In a third step, we therefore apply two levels of hierarchical refinement to the propeller blades, while we leave the discretization of the central hub as it is, in order to achieve a homogeneous resolution of the two different thicknesses. Whereas a uniformly refined mesh of the propeller that provides the same resolution of the blades has 80,922 dofs, the adaptive mesh shown in Figure 7.19b has only 53,052 dofs. In a fourth step, we equip each element cut by the geometric boundary by additional sub-cells, which are organized in an octree of depth two, generated by the partitioning procedure described in sub-section 7.1. The sub-cells corresponding to the leaves of the first and second levels of the octree are shown in Figure 7.20a and 7.20b, respectively. It should be noted again that the blue sub-cells of Figure 7.20 do not affect B-spline basis functions, which are still defined over the set of elements shown in black lines in Figure 7.19b. Each sub-cell contains 4×4×4 Gauss points, leading to an aggregation of integration points in cut elements to accurately take into account the geometric boundary during numerical integration.

Following the finite cell method and the fictitious domain concept, which has been discussed in detail in Section 2.1, parameter α specified in Equation (2.3) penalizes the elasticity matrix

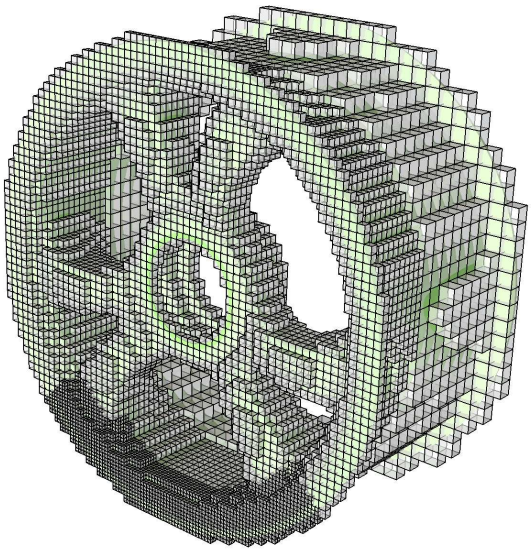
[7]Courtesy of Prof. Manfred Krafczyk and co-workers, Institut für rechnergestützte Modellierung im Bauingenieurwesen, Technische Universität Braunschweig; www.tu-braunschweig.de/irmb



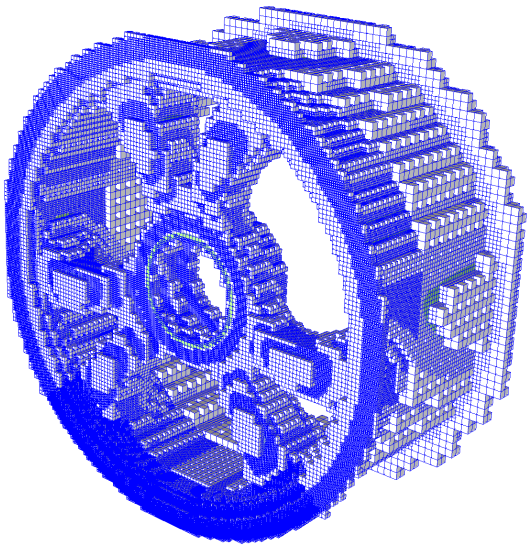
(a) The complete structure is immersed in a bounding box of 30×15×30 axis-aligned B-spline elements.



(b) Elements without support in the physical domain of the rim are deleted.



(c) Hierarchical refinement adaptively increases the resolution, where high gradients of the solution are expected.



(d) Cut elements are adaptively partitioned twice for accurate integration of the geometric boundary (473,641 sub-cells).

Figure 7.23: Immersed boundary discretization of the rim of a car wheel with the B-spline version of the finite cell method and hierarchical refinement for adaptive stress-displacement analysis.

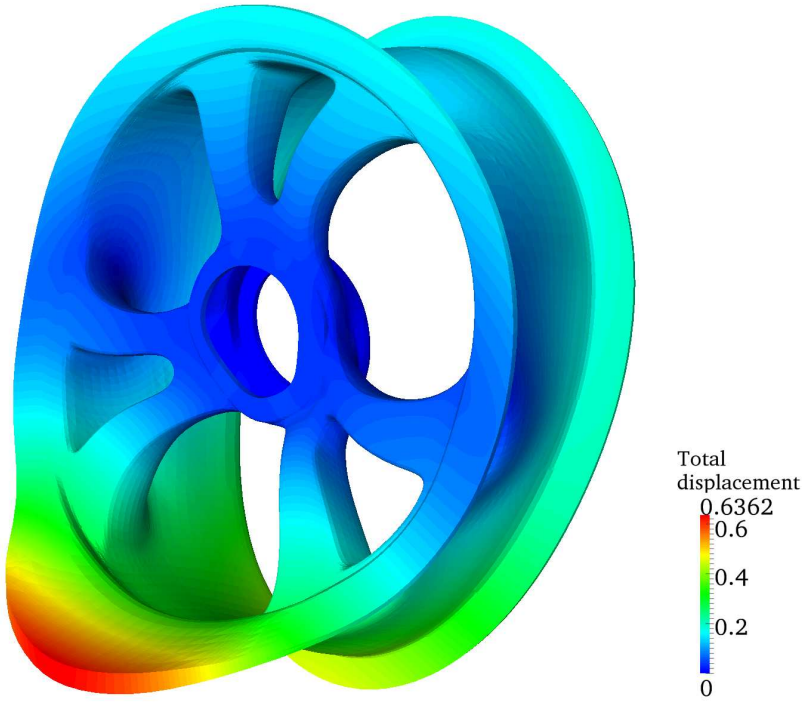


Figure 7.24: Total displacements in mm, plotted on the deformed structure. Steep gradients of the solution around the loading area on the lower edge are adaptively resolved by hierarchical refinement.

in the fictitious domain. It is introduced into the weak form of linear elasticity given in Equation (2.1) via the stress-strain relation of Equation (2.2). The linear elastic free vibration problem additionally considers the influence of the mass of the structure characterized by its homogeneous density ρ. The corresponding mathematical derivation and its numerical treatment within the finite element method are reported in detail for example in [17, 108, 222]. In analogy to the stiffness integral of Equation (2.1), parameter α of Equation (2.3) is also applied to penalize the density ρ in the fictitious domain in order to maintain consistency with the original problem. The weak form of the free vibration problem in the framework of the finite cell method follows as

$$\int_{\Omega} \boldsymbol{\sigma} \,:\, (\nabla_{sym}\, \delta\boldsymbol{u})\; dV \;-\; \lambda \int_{\Omega} (\alpha\rho)\, \delta\boldsymbol{u} \cdot \boldsymbol{u}\, dV \;=\; 0 \tag{7.6}$$

In the corresponding finite cell discretization, the contribution to the stiffness matrix and the mass matrix are penalized by factor α in the sense of Equation (2.3), if a point location query indicates that an integration point is located outside the propeller domain Ω_{phys}. For the free vibration analysis of the propeller, we choose a moderately small $\alpha{=}10^{-3}$, which prevents an extreme ill-conditioning of the system, so that an iterative Eigenvalue solver can be applied.

The hierarchically refined mesh of Figure 7.19b is analysis suitable and is used in combination with the sub-cells of Figure 7.20 to conduct a free vibration analysis of the structure, where the mass matrix is lumped according to the row sum method [108]. For the solution of the Eigenvalue problem, we applied the library package ARPACK [8]. Figure 7.21 illustrates the first seven mode shapes. The three lowest Eigenforms, each of which corresponds to one

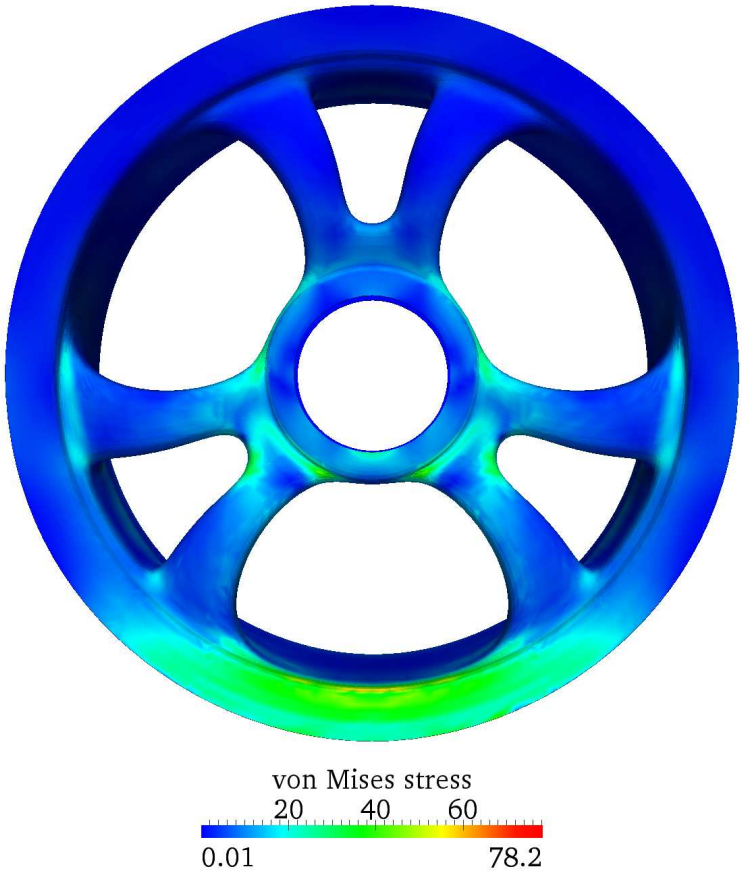


Figure 7.25: Von Mises stress in the rim of a car wheel (Front view).

pair of opposing blades, exhibit a rotational symmetry in the propeller plane around the center of the hub. The next two mode shapes are the same, but the second is rotated 90° compared to the first within the plane of the propeller. They are thus symmetric with respect to the two axes that span the propeller plane. The sixth Eigenform is single and denotes a rotationally symmetric bending of the blades out of the propeller plane. The seventh mode shape shows an out-of-plane bending of two opposite blades. The reflection of geometric symmetries in the Eigenforms corresponds very well to engineering experience and indicates a good quality of the first modes. Unfortunately, we found that higher mode shapes were corrupted.

7.4.2 Stress-displacement analysis of a rim of a car wheel

The second example is a rim of a car wheel. Analogous to the propeller, its geometry is originally described by a T-spline surface, which is exported from the CAD package Rhino [166] using the T-spline plug-in [205] in the form of Bézier elements as shown in Figure 7.22a. Its maximum diameter and height are 239.26 mm and 223.28 mm, respectively, and its material is aluminium with Young's modulus $7.0{\cdot}10^4$ N/mm^2 and Poisson's ratio 0.34. We apply an ellipsoidal line load at the bottom outer edge of the rim in the direction of the global y-axis, and homogeneous Dirichlet constraints at the hub fix the structure, simulating the suspension of the wheel. Horizontal loadings typically occur due to centrifugal forces, when the vehicle

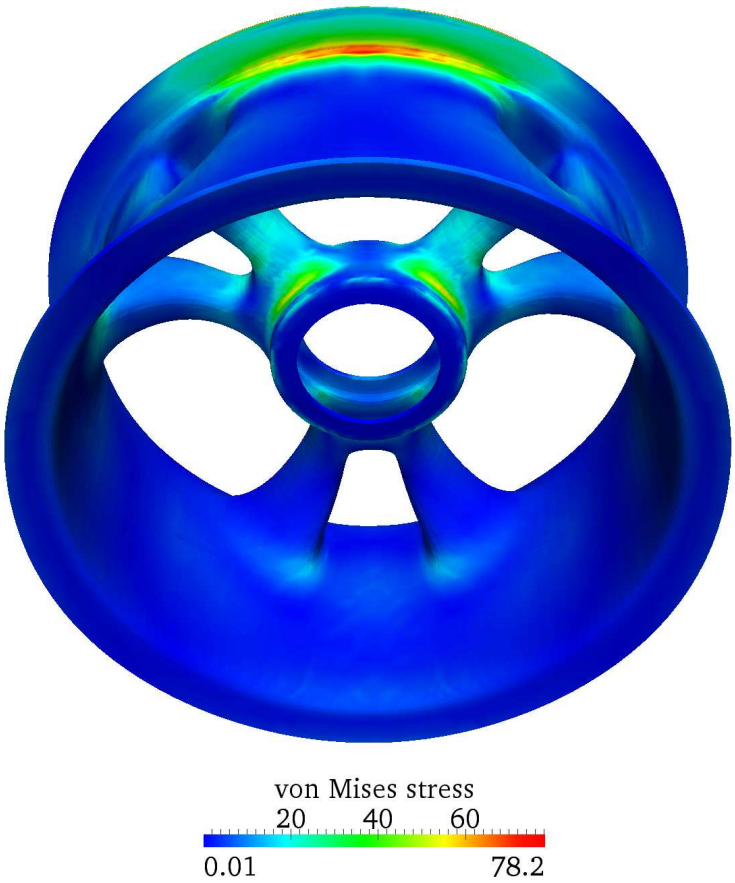


Figure 7.26: Von Mises stress in the rim of a car wheel (Bottom-up view).

enters into a curve, and are transferred from the road surface onto the outer edge of the rim by the tire. The boundary conditions are further specified in Figure 7.22b.

The discretization procedure in the framework of the immersed boundary approach is illustrated in Figure 7.23. In a first step, the structure is completely immersed into a cuboidal bounding box of axis-aligned knot span elements, over which uniform B-splines are defined (Figure 7.23a). In a second step, elements located completely outside the physical domain Ω_{phys} of the wheel are erased from the discretization (Figure 7.23b). The location of the elements is determined by the straightforward point location query based procedure described in the previous sub-section. In a third step, we apply hierarchical refinement to adaptively resolve those parts of the structure, where we expect larger gradients in the solution fields (Figure 7.23c). The first level of refinement addresses the complex geometric parts of the spokes, the central hub and the upper part of the tire bearing. The second level of refinement is applied to the lower part of the tire bearing, where the loading is imposed. In a last step, we apply the simple partitioning procedure of sub-section 7.1 to elements cut by the geometric boundary, which generates two levels of sub-cells that accurately resolve the geometric boundary by integration points (Figure 7.23d). The stiffness contribution of integration points located outside the rim domain are penalized by factor $\alpha{=}10^{-6}$ in the sense of Equation (2.3).

Dirichlet constraints are considered in a weak sense by applying Nitsche's method, whose variational formulation requires an integration over the Dirichlet boundary. Taking advantage

of the axis-aligned Dirichlet surface Γ_D shown in Figure 7.22b, corresponding integrals can be evaluated by a projection of Gauss points onto Γ_D in each sub-cell that cuts Γ_D. The discrete system of equations is passed to the direct solver Pardiso [113]. The resulting deformation of the rim is illustrated in Figure 7.24, where the displacements imposed on the structure are magnified by a factor of 300 for better visibility. It can be observed that the solution in the local vicinity of the loading area at the outer edge of the rim exhibits rapid changes, which confirm the necessity of local hierarchical refinement there. The adaptive mesh of Figure 7.23c with 105,807 dofs considerably decreases the computational effort in comparison to uniform refinement, which requires 622,989 dofs for the same resolution of the loading area. Figures 7.25 and 7.26 show von Mises stresses, plotted on the boundaries of the undeformed structure. Large stresses occur predominantly around the lower tire bearing, the two lower spokes and the central hub, thus tracing the flow of forces from the loading area to the support, while the rest of the structure remains unstressed. According to the symmetry in geometry and boundary constraints, the stress solution is completely symmetric and exhibits the typical stress concentration phenomena at reentrant sharp curves (see front view of Figure 7.25). Due to the superposition of tensile membrane and bending stress, the maximum von Mises stress occurs at the sharp reentrant bend, where the loaded boundary ring bends into the main cylinder of the tire bearing (bottom-up view of Figure 7.26).

Chapter 8

Summary and conclusions

The present work intends to contribute to a more efficient interaction between geometrical models and finite element analysis to alleviate the discretization challenge for complex geometries. In this context, the main focus was placed on two specific concepts: the finite cell method and isogeometric analysis.

In the framework of *the finite cell method*, four new aspects were discussed: The independence of the finite cell concept from a special high-order basis; the B-spline version of the finite cell method, which applies high-order and high-continuity B-spline bases within the FCM concept; the extension of the finite cell method to geometrically nonlinear problems; and the combination of the finite cell method with the *hp-d*-adaptive approach as a possibility for local mesh refinement. The main results of this part can be summarized as follows:

- The finite cell method can be characterized as a seamless geometry-through-analysis procedure, which combines the fictitious domain approach with a high-order approximation basis, adaptive integration and weak imposition of unfitted Dirichlet constraints.
- Basis functions of the p-version of the finite element method and B-splines constitute suitable high-order Ansatz spaces, which successfully instantiate the characteristic FCM properties, i.e. smooth extension of solution fields beyond the physical domain and high rates of convergence under p-refinement. In addition, both FCM versions were shown to yield optimal rates of convergence under h-refinement and a high quality of stresses on geometric boundaries that cut through finite cells.
- For geometrically nonlinear problems, the straightforward extension of the standard FCM formulation using the same nonlinear kinematics over the complete domain leads to the loss of uniqueness of the deformation map in the penalized fictitious domain.
- A possible way out is a modified geometrically nonlinear FCM formulation, based on repeated deformation resetting, which assumes for the fictitious domain the deformation-free reference configuration after each Newton iteration. The uniqueness issue can be completely circumvented, since the deformation map of the fictitious domain does not need to be computed. Deformation resetting is equivalent to performing repeated linear analyses in the fictitious domain, which considerably reduces the computational cost.

- For the imposition of unfitted Dirichlet boundary conditions in a geometrically nonlinear context, a simple penalty formulation, which can be easily linearized in a consistent manner, yields excellent stress and displacement results on the Dirichlet boundary and preserves high rates of convergence.
- Both FCM versions inherit the robustness of the corresponding high-order basis. They are able to accurately represent very large deformation states despite the presence of the fictitious domain and severe distortion of the physical parts of the finite cell meshes.
- The finite cell method provides a seamless geometry-through analysis procedure for voxel based geometrical models, since it is able to directly operate on the voxel information without any preprocessing, segmentation of the voxel data or time-consuming mesh generation. This was illustrated by the large deformation analysis of a CT-based metal foam sample, for which accurate results could be achieved with both FCM versions.
- While the overall solution characteristics were found to be equivalent, a detailed comparison reveals specific strengths of each FCM version: For knot span and p-version discretizations with a comparable number of degrees of freedom, the B-spline version tends to yield more accurate results due to the higher-order continuity of its basis. The p-version tends to be computationally more efficient due to a lower degree of population and bandwidth of the stiffness matrix.
- Hierarchical *hp-d* adaptivity constitutes a suitable method for local mesh refinement in the framework of the finite cell method. In particular, it preserves the easy meshing property as a key advantage of the FCM concept.
- While an *hp-d*-adaptive p-version of the FCM can be based on overlays of linear hierarchical basis functions, the *hp-d*-adaptive B-spline version can use overlays of hierarchical subdivision B-splines. In a proof of concept, both FCM versions were applied for local mesh refinement around material interfaces and singular points, which led to a considerable improvement of convergence rates and motivates future work beyond this thesis.

In the framework of *isogeometric analysis*, a hierarchical refinement procedure based on the intuitive concept of B-spline subdivision was derived, which combines full analysis suitability with direct generalization to higher dimensions, in particular 3D, and a straightforward implementation in tree data structures. Theoretical concepts, algorithmic and implementation aspects for hierarchical refinement of B-splines as well as their extension to NURBS were discussed in detail. The main results of this part can be summarized as follows:

- Hierarchical refinement of B-splines and NUBRS as described in this work was successfully tested as a basis for adaptive tensor-product isogeometric analysis of some elementary fluid and structural analysis problems.
- Hierarchical refinement of B-splines leads to adaptive meshes with highly localized refinement, and no propagation away from the areas of interest was experienced. A comparison of adaptive hierarchical with standard uniform refinement shows that the computational effort in terms of degrees of freedom can be in general reduced about one order of magnitude at a comparable level of accuracy.

- The isogeometric paradigm assumes that the initial geometry description is already exact. Therefore, the initial unrefined NURBS patch can be maintained for the description of the geometry throughout the refinement process in the sense of a sub-parametric finite element scheme. Under these premises, only minor adjustments are necessary to directly generalize concepts and algorithms of hierarchical B-spline refinement to NURBS.
- All beneficial characteristics fully carry over to three-dimensional solid NURBS discretizations, for which other refinement techniques such as T-splines have not been shown to be applicable yet.
- The B-spline version of the FCM was applied for embedded domain analysis of a ship propeller and a rim of a car wheel, whose geometry description was given by complex CAD based T-spline surfaces. The examples demonstrate the potential of embedded domain analysis for a full automation of the discretization process for CAD based geometrical models irrespective of their geometric complexity.
- In this context, hierarchical refinement of NURBS can considerably increase the flexibility of B-spline based embedded domain methods in terms of adaptive resolution of local features in geometry and solution fields, while due to its simplicity and straightforward implementation, the key advantage of automated mesh generation for complex geometries can be fully maintained.

The results of the first part show that the finite cell method is able to fully profit from the advantages of its high-order approximation basis, such as high-rates of convergence or robustness against severe mesh distortion, while at the same time it is able to seamlessly integrate a range of important sub-methods, such as algorithms for geometric nonlinearity, Nitsche's method or *hp-d* adaptivity. We therefore believe that the finite cell method is a very flexible tool that offers an unprecedented combination of high accuracy at a moderate computational effort, full automation of the discretization process irrespective of the geometric complexity involved, and applicability to almost any geometrical model. Based on the results of the second part, we think that hierarchical refinement in this form has great potential to establish itself as a straightforward, efficient and versatile technique for local refinement in NURBS based isogeometric analysis. In addition, we are convinced that embedded domain methods such as the B-spline version of the finite cell method can open the door for a seamless IGA design-through-analysis procedure for complex engineering parts and assemblies, where a plethora of highly relevant and very promising aspects still need to be examined. Examples comprise the performance optimization of adaptive integration schemes, which could dramatically increase the computational efficiency, the analysis of topology changes and moving boundaries, for which embedded domain methods offer significant advantages over ALE-type approaches, or the introduction of embedded domain suitable coupling schemes for multiphysics problems, which stand at the forefront of today's challenges in computational science and engineering.

Bibliography

[1] Abedian A, Parvizian J, Düster A, Khademyzadeh H, Rank E. The Finite Cell Method for Elasto-Plastic Problems. In: *Proc. of the 10th Int. Conference on Computational Structures Technology*, Valencia, 2010.

[2] Agoston MK. *Computer Graphics and Geometric Modeling: Mathematics.* Springer: London, 2005.

[3] Agoston MK. *Computer Graphics and Geometric Modeling: Implementation and Algorithms.* Springer: London, 2005.

[4] Ainsworth M, Oden JT. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 1997; **142**:1–88.

[5] Akcelik V, Bielak J, Biros G, Epanomeritakis I, Fernandez A, Ghattas O, Kim EJ, Lopez J, O'Hallaron D, Tu T, Urbanic J. High-resolution forward and inverse earthquake modeling on terascale computers. In: *Proceedings of ACM/IEEE SC 2003*, Phoenix, 2003.

[6] Akkerman I, Bazilevs Y, Calo VM, Hughes TJR, Hulshoff S. The role of continuity in residual-based variational multiscale modeling of turbulence. *Computational Mechanics*, 2008; **41**:371–378.

[7] Allaire G, Jouve F, Toader A. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 2004; **194**:363–393.

[8] ARPACK. Software for the solution of large scale eigenvalue problems developed by R. Lehouq, D. Sorensen, K. Maschhoff, C. Yang; http://www.caam.rice.edu/software/ARPACK/ (October 11th, 2011)

[9] Auricchio F, da Veiga LB, Lovadina C, Reali A. The importance of the exact satisfaction of the incompressibility constraint in nonlinear elasticity: mixed fems versus NURBS-based approximations. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:314–323.

[10] Baaijens FPT. A fictitious domain/mortar element method for fluid-structure interaction. *International Journal of Numerical Methods in Fluids*, 2001; **35**:743–761.

[11] Babuška I. The finite element method with penalty. *Mathematics of Computation*, 1972; **27**(122):221–228.

[12] Babuška I, Guo BQ. Approximation properties of the *hp*-version of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 1996; **133**:319–346.

[13] Babuška I, Banerjee U, Osborn JE. Meshless and generalized finite element methods: A survey of some major results. In: M. Griebel, M.A. Schweitzer (Eds.), *Meshfree Methods for Partial Differential Equations, Lecture Notes in Computational Science and Engineering, vol. 26.*, Springer: Berlin, 2002.

[14] Bangerth W, Burstedde C, Heister T, Kronbichler M. Algorithms and Data Structures for Massively Parallel Generic Adaptive Finite Element Codes. Accepted for publication in *ACM Transactions on Mathematical Software*, 2011.

[15] Banhart J. Manufacture, characterization and application of cellular metals and metal foams. *Progress in Material Science*, 2001; **46**:559–632.

[16] Bastian P, Engwer C. An unfitted finite element method using discontinuous Galerkin. *International Journal of Numerical Methods in Engineering*, 2009; **79**:1557–1576.

[17] Bathe K-J. *Finite Element Procedures.* Prentice Hall: Upper Saddle River, 1996.

[18] Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 2007; **197**:173–201.

[19] Bazilevs Y, Hughes TJR. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 2007; **36**:12–26.

[20] Bazilevs Y, Calo VM, Hughes TJR, Zhang Y. Isogeometric fluid-structure interaction: theory, algorithms and computations. *Computational Mechanics*, 2008; **43**:3–37.

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

[22] Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJR, Lipton S, Scott MA, Sederberg TW. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:229–263.

[23] Bazilevs Y, Hsu MC, Kiendl J, Wuechner R, Bletzinger KU. 3D Simulation of Wind Turbine Rotors at Full Scale. Part II: Fluid-Structure Interaction. *International Journal of Numerical Methods in Fluids*, 2011; **65**:236–253.

[24] Becker R, Burman E, Hansbo P. A hierarchical NXFEM for fictitious domain simulations. *International Journal of Numerical Methods in Engineering*, 2010; **86**:54–559.

[25] Belytschko T, Stolarski H, Liu WK, Carpenter N, Ong JSJ. Stress projection for membrane and shear locking in shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 1985; **51**:221–258.

[26] Belytschko T, Liu WK, Moran B. *Nonlinear Finite Elements for Continua and Structures.* Wiley: New York, 2006.

[27] Benson DJ, Bazilevs Y, Hsu MC, Hughes TJR. Isogeometric shell analysis: The Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:276–289.

[28] Bertsekas DP. *Constrained Optimization and Lagrange Multiplier Methods.* Athina Scientific: Nashua, 1996.

[29] Bielak J, Ghattas O, Kim EJ. Parallel octree-based finite element method for large-scale earthquake ground motion simulation. *Computer Modeling in Engineering and Sciences*, 2005; **10**(2):99–112.

[30] Bindick S, Stiebler M, Krafczyk M. Fast kd-tree-based hierarchical radiosity for radiative heat transport problem. *International Journal for Numerical Methods in Engineering*, 2011; **86**:1082–1100.

[31] Bischoff M, Wall WA, Bletzinger K-U, Ramm E. Models and finite elements for thin-walled structures. In: Stein E, de Borst R, Hughes TJR (eds). *Encyclopedia of Computational Mechanics, Vol. 2. Solids, Structures and Coupled Problems*, Wiley: New York, 2004.

[32] Bischoff M, Ramm E. *Advanced computational structural mechanics.* Lecture notes, Institute of Structural Mechanics, Universität Stuttgart, 2007.

[33] Bishop J. Rapid stress analysis of geometrically complex domains using implicit meshing. *Computational Mechanics*, 2003; **30**:460–478.

[34] Bonet J, Wood R. *Nonlinear Continuum Mechanics for Finite Element Analysis.* Cambridge University Press: Cambridge, 2008.

[35] Borden MJ, Verhoosel CV, Scott MA, Hughes TJR, Landis CM. A phase-field description of dynamic brittle fracture. Preprint submitted to *Computer Methods in Applied Mechanics and Engineering*, 2011.

[36] Brooks AN, Hughes TJR. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 1982; **32**:199–259.

[37] Bungartz HJ, Griebel M. Sparse grids. *Acta Numerica*, 2004; **13**:147–269.

[38] Bungartz HJ, Griebel M, Zenger C. *Introduction to Computer Graphics.* Charles River Media: Hingham, 2004.

[39] Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:2680–2686.

[40] Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Applied Numerical Mathematics*, 2011; doi:10.1016/j.apnum.2011.01.008.

[41] Burstedde C, Wilcox LC, Ghattas O. p4est: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees. *SIAM Journal on Scientific Computing*, 2011; **33**(3):1103–1133.

[42] Cai Q, Kollmannsberger S, Mundani RP, Rank E. The finite cell method for solute transport problems in porous media. In: *Proc. of the 16th Int. Conference on Finite Elements in Flow Problems*, Munich, 2011.

[43] Canuto C, Hussaini MY, Quarteroni A, Zang TA. *Spectral Methods. Fundamentals in Single Domains.* Springer: Heidelberg, 2006.

[44] Carey GF, Utku M. Stream function finite element solution of Stokes flow with penalties. *International Journal for Numerical Methods in Fluids*, 1984; **4**:1013–1025.

[45] Chapman B, Jost G, van der Pas R. *Using OpenMP. Portable Shared Memory Parallel Programming.* MIT Press: Cambridge, 2008.

[46] Chen HF, Han FS. Compressive behavior and energy absorbing characteristic of open cell aluminum foam filled with silicate rubber. *Scripta Materialia*, 2003; **49**:583–586.

[47] Chen W, Cai Y, Zheng J. Generalized hierarchical NURBS for interactive shape modification. In: *Proceedings of the 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applations in Industry*, 2008.

[48] Chen W, Cai Y, Zheng J. Freeform-based form feature modeling using a hierarchical & multi-resolution NURBS method. In: *Proceedings of the 9th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applations in Industry*, 2010.

[49] Chilton L, Suri M. On the selection of a locking-free *hp* element for elasticity problems. *International Journal for Numerical Methods in Engineering*, 1997; **40**:2045–2062.

[50] Chui CK. *An introduction to wavelets.* Academic Press: London, 1992.

[51] Cluff DRA, Esmaeili S. Compressive properties of a new metal-polymer hybrid material. *Journal of Material Science*, 2009; **44**:3867–3876.

[52] Cirak F, Ortiz M, Schröder P. Subdivision surfaces: A new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 2000; **47**(12):2039–2072.

[53] Codina R. On stabilized finite element methods for linear systems of convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 2000; **188**(1-2):61–82.

[54] Cohen E, Martin T, Kirby RM, Lyche T, Riesenfeld R. Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:335–356.

[55] Cottrell JA, Reali A, Bazilevs Y, Hughes TJR. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 2006; **195**:5257–5296.

[56] Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric Analysis: Toward Integration of CAD and FEA.* Wiley: New York, 2009.

[57] Common Versatile Multi-purpose Library for C++, developed at the University of Geneva; http://tech.unige.ch/cvmlcpp/ (October 13^{th}, 2011)

[58] de Berg M, Cheong O, van Kreveld M, Overmars M. *Computational Geometry: Algorithms and Applications.* Springer: Heidelberg, 2008.

[59] Del Pino S, Pironneau, O (2003) A fictitious domain based general pde solver. In: Numerical methods for scientific computing: Variational problems and applications. Kuznetsov Y, Neittanmaki P, Pironneau O (eds), CIMNE, Barcelona

[60] Dedè L, Borden MJ, Hughes TJR. Isogeometric Analysis for topology optimization with a phase field model. *ICES report 11-29*, The University of Texas at Austin, October 2011.

[61] Demkowicz L. *Computing with hp-Adaptive Finite Elements; Volume 1: One and Two Dimensional Elliptic and Maxwell Problems.* Chapman&Hall/CRC:Boca Raton, 2006.

[62] Demkowicz L, Kurtz J, Pardo D, Paszynski M, Rachowicz W, Zdunek A. *Computing with hp-Adaptive Finite Elements; Volume 2: Three Dimensional Elliptic and Maxwell Problems with Applications.* Chapman&Hall/CRC:Boca Raton, 2007.

[63] Deng J, Chen F, Li X, Hu C, Tong W, Yang Z, Feng Y. Polynomial splines over hierarchical T-meshes. *Graphical Models*, 2008; **70**:76–86.

[64] de Souza Neto EA, Perić D, Owen DRJ (2008) Computational Methods for Plasticity: Theory and Applications. Wiley, New York

[65] Deuflhard P, Leinen P, Yserentant H. Concepts of an adaptive hierarchical finite element code. *Impact of Computing in Science and Engineering*, 1989; **1**:3–35.

[66] Dolbow J, Harari I. An efficient finite element method for embedded interface problems. *International Journal for Numerical Methods in Engineering*, 2009; **78**:229–252.

[67] Donea J, Huerta A. *Finite Element Methods for Flow Problems.* Wiley: Chichester, 2003.

[68] Dong S, Yosibash Z. A parallel spectral element method for dynamic three-dimensional nonlinear elasticity problems. *Computers & Structures*, 2009; **87**(1-2):59–72.

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

[70] Düster A. High order finite elements for three-dimensional, thin-walled nonlinear continua. PhD thesis, Lehrstuhl für Bauinformatik, Technische Universität München, Shaker: Aachen, 2001.

[71] Düster A, Bröker H, Rank E (2001) The *p*-version of the finite element method for three-dimensional curved thin walled structures. *International Journal for Numerical Methods in Engineering*, 2001; **52**:673–703.

[72] Düster A, Hartmann S, Rank E. *p*-fem applied to finite isotropic hyperelastic bodies. *Computer Methods in Applied Mechanics and Engineering*, 2003; **192**:5147–5166.

[73] Düster A, Niggl E, Rank E. Applying the *hp-d* version of the FEM to locally enhance dimensionally reduced models. *Computer Methods in Applied Mechanics and Engineering*, 2007. **196**:3524–3533.

[74] Düster A, Parvizian J, Yang Z, Rank E. The Finite Cell Method for Three-Dimensional Problems of Solid Mechanics. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:3768–3782.

[75] Echter R, Bischoff M. Numerical efficiency, locking and unlocking of NURBS finite elements. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:374–382.

[76] Ehlers W. *Elemente der nichtlinearen Kontinuumsthermodynamik.* Lecture notes, Institute of Structural Mechanics, Universität Stuttgart, 2006. (in German)

[77] Elguedj T, Bazilevs Y, Calo VM, Hughes TJR. $\bar{B}$ and $\bar{F}$ projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:2732–2762.

[78] Embar A, Dolbow J, Harari I. Imposing Dirichlet boundary conditions with Nitsche's method and spline-based finite elements. *International Journal for Numerical Methods in Engineering*, 2010; **83**:877–898.

[79] Evans JA, Bazilevs Y, Babuška I, Hughes TJR. n-Widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 2009; **198**(21–26):1726–1741.

[80] Evans JA, Hughes TJR. Explicit trace inequalities for isogeometric analysis and parametric finite elements. *ICES report 11-17*, The University of Texas at Austin, May 2011.

[81] Farhat C, Hetmaniuk U. A fictitious domain decomposition method for the solution of partially axisymmetric acoustic scattering problems. Part I: Dirichlet boundary conditions. *International Journal for Numerical Methods in Engineering*, 2002; **54**:1309–1332.

[82] Farin G. *Curves and surfaces for CAGD.* Morgan Kaufmann Publishers: San Francisco, 2002.

[83] Felippa CA. *Introduction to finite element methods.* University of Colorado, Boulder. http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/Home.html (October 10^{th} , 2010)

[84] Fernandez-Mendez S, Huerta A. Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 2004; **193**:1257–1275.

[85] Fiedler T, Solórzano E, Garcia-Moreno F, Öchsner A, Belova IV, Murch GE. (2009) Computed tomography based finite element analysis of the thermal properties of cellular aluminium. *Materialwissenschaft und Werkstofftechnik*, 2009; **40**:139–143.

[86] FlagShyp 08 Version 2.30. Nonlinear FE solver developed by J. Bonet and R. Wood. http://www.flagshyp.com

[87] Flemisch B, Wohlmuth B. Stable Lagrange multipliers for quadrilateral meshes of curved interfaces in 3D. *Computer Methods in Applied Mechanics and Engineering*, 2007; **196**:1589–1602.

[88] Forsey D, Bartels RH. Hierarchical B-Spline refinement. *Computer Graphics* (SIGGRAPH '88 Proceedings), 1988; **22**(4):205–212.

[89] Forsey D, Bartels RH. Surface fitting with hierarchical B-Splines. *ACM Transactions on Graphics*, 1995; **14**(2):134–161.

[90] Franke D, Düster A, Nübel V, Rank E. A comparison of the *h*-, *p*-, *hp*-, and *rp*-version of the FEM for the solution of the 2D Hertzian contact problem. *Computational Mechanics*, 2010; **45**:513–522.

[91] Gerstenberger A, Wall WA. An eXtended Finite Element Method/Lagrange multiplier based approach for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:1699–1714.

[92] Gerstenberger A, Wall WA. An embedded Dirichlet formulation for 3D continua. *International Journal for Numerical Methods in Engineering*, 2010; **82**:537–563.

[93] Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 1982; **48**:387–411.

[94] Glowinski R, Kuznetsov Y. Distributed Lagrange multipliers based on fictitious domain method for second order elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 2007; **196**:1498–1506.

[95] Golub G, Van Loan CF. *Matrix Computations.* Johns Hopkins University Press: Baltimore, 1996.

[96] Gomez H, Calo VM, Bazilevs Y, Hughes TJR. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:4333–4352.

[97] Govindjee S, Strain J, Mitchell TJ, Taylor R. Convergence of an Efficient Local Least-Squares Fitting Method for Bases with Compact Support. Manuscript submitted to *Computer Methods in Applied Mechanics and Engineering*, 2011.

[98] Grinspun E, Krysl P, Schröder P. CHARMS: A Simple Framework for Adaptive Simulation. *ACM Transactions on Graphics*, 2002; **21**(3), 281–290.

[99] Hansbo A, Hansbo P. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 2002; **191**:537–552.

[100] Harari I , Dolbow J. Analysis of an efficient finite element method for embedded interface problems. *Computational Mechanics*, 2010; **46**:205–211.

[101] Haslinger J, Kozubek T, Kunisch K, Peichl G. Shape Optimization and Fictitious Domain Approach for Solving Free Boundary Problems of Bernoulli Type. *Computational Optimization and Applications*, 2003; **26**:231–251.

[102] Haslinger J, Renard, Y. A New Fictitious Domain Approach Inspired by the Extended Finite Element Method. *SIAM Journal on Numerical Analysis*, 2009; **47**:1474–1499.

[103] Heisserer U. *High-order finite elements for material and geometrical nonlinear finite strain problems.* PhD thesis, Lehrstuhl für Computation in Engineering, Technische Universität München. Shaker: Aachen, 2008.

[104] Heisserer U, Hartmann S, Düster A, Yosibash Z. On volumetric locking-free behavior of p-version finite elements under finite deformations. *Communications in Numerical Methods in Engineering*, 2008; **24**:1019–1032.

[105] Höllig K, Reif U, Wipper J. Weighted extended b-spline approximation of Dirichlet problems. *SIAM Journal on Numerical Analysis*, 2001; **39**:442–462.

[106] Höllig K. *Finite Element Methods with B-Splines.* Society for Industrial and Applied Mathematics: Philadelphia, 2003.

[107] Holzapfel G. *Nonlinear Solid Mechanics. A Continuum Approach for Engineering.* Wiley: New York, 2000.

[108] Hughes TJR. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Dover: New York, 2000.

[109] Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 2005; **194**:4135–4195.

[110] Ibrahimbegovic A. *Nonlinear Solid Mechanics: Theoretical Formulations and Finite Element Solution Methods*, Springer: Heidelberg, 2009.

[111] Intel Corporation. Math Kernel Library version 10.1 (MKL); http://software.intel.com/en-us/articles/intel-mkl (October 11^{th}, 2011)

[112] Intel Corporation. C/C++ compiler suite 11.1 (icc); http://software.intel.com/en-us/articles/intel-compilers (October 11^{th}, 2011)

[113] Intel Corporation. PARDISO solver version 10.1, part of the Intel's MKL library; http://software.intel.com/en-us/articles/intel-mkl (October 11^{th}, 2011)

[114] Jhaver R, Tippur H. Processing, compression response and finite element modeling of syntactic foam based interpenetrating phase composite (IPC). *Materials Science and Engineering: A*, 2009; **499**:507–517.

[115] Kagan P, Fischer A. Integrated mechanically based CAE system using B-spline finite elements. *Computer Aided Design*, 2000; **32**(8-9):539–552.

[116] Keyak J, Falkinstein Y. Comparison of in situ and in vitro CT-scan-based finite element model predictions of proximal femoral fracture load. *Medical Engineering & Physics*, 2003; **25**:781–787.

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

[118] Kobbelt L. Multiresolution techniques. In: Farin G, Hoschek J, Sim MS (eds). *Handbook of Computer Aided Geometric Design.* Morgan Kaufmann Publishers: San Francisco, 2002, 343–361.

[119] Kraft R. Adaptive and linearly independent multilevel B-splines. In: *Surface Fitting and Multiresolution Methods,* Méhauté AL, Rabut C, Schumaker LL (eds). Vanderbilt University Press, 1997; 209–218.

[120] Krause R, Mücke R, Rank E. *hp*-version finite elements for geometrically non-linear problems. *Communications in Numerical Methods in Engineering*, 1995; **11**:887–897.

[121] Krause R, Rank E. Multiscale computations with a combination of the *h*- and *p*-versions of the finite-element method. *Computer Methods in Applied Mechanics and Engineering*, 2003. **192**:3959–3983.

[122] Krysl P, Grinspun E, Schröder P. Natural hierarchical refinement for finite element methods. *International Journal for Numerical Methods in Engineering*, 2003; **56**:1109–1124.

[123] Lapack Version 3.2.2. Linear Algebra Package; http://www.netlib.org/lapack/ (October 11^{th}, 2011)

[124] Legay A, Wang HW, Belytschko T. Strong and weak arbitrary discontinuities in spectral finite elements. *International Journal for Numerical Methods in Engineering*, 2005; **64**:991–1008.

[125] Li X, Deng J, Chen F. Polynomial splines over general T-meshes. *Visual Computer*, 2010; **26**:277–286.

[126] Li X, Zheng J, Sederberg TW, Hughes TJR, Scott MA. On Linear Independence of T-splines. *ICES report 10-40*, The University of Texas at Austin, October 2010.

[127] Lipton S, Evans JA, Bazilevs Y, Elguedj T, Hughes TJR. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:357–373.

[128] Livermore Software Technology Corporation. LS-Dyna 971 R5 user's manual. Livermore, CA

[129] Löhner R, Cebral RJ, Camellia FE, Appanaboyinaa S, Baum JD, Mestreau EL, Soto OA. Adaptive embedded and immersed unstructured grid techniques. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:2173–2197.

[130] Lui S (2009) Spectral domain embedding for elliptic PDEs in complex domains. *Journal of Computational and Applied Mathematics*, 2009; **225**:541–557.

[131] MacNeal RH, Harder RL. A Proposed Standard Set of Problems To Test Finite Element Accuracy. *Finite Elements in Analysis and Design*, 1985; **1**:3–20.

[132] MacNeal-Schwendler Corporation. MSC Nastran Numerical Methods User's Guide. Santa Ana, CA, 2001.

[133] Malvern LE. *Introduction to mechanics of a continuous medium.* Prentice Hall: Englewood Cliffs, 1969.

[134] Meister A. *Numerik linearer Gleichungssysteme: Eine Einführung in moderne Verfahren.* Vieweg & Teubner: Wiesbaden, 2011. (in German)

[135] Mergheim J, Steinmann P. A geometrically nonlinear FE approach for the simulation of strong and weak discontinuities. *Computer Methods in Applied Mechanics and Engineering*, 2006; **195**:5037–5052.

[136] Miehe C. *Einführung in Numerische Methoden der Mechanik.* Lecture notes, Institute of Applied Mechanics, Universität Stuttgart, 2005. (in German)

[137] Miehe C. *Geometrical methods of nonlinear continuum mechanics.* Lecture notes, Institute of Applied Mechanics, Universität Stuttgart, 2007.

[138] Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 2005; **37**:239–261.

[139] Mok DP, Wall WA, Bischoff M, Ramm E. Algorithmic aspects of deformation dependent loads in non-linear static finite element analysis. *Engineering Computations*, 1999; **16**:601–618.

[140] Mundani RP, Bungartz HJ, Rank E, Niggl A, Romberg R. Extending the p-Version of Finite Elements by an Octree-Based Hierarchy. In: Widlund OB, Keyes DE (eds.), *Lecture Notes in Computational Science and Engineering*, Vol. 55, Springer: New York, 2006, pp. 699–706.

[141] Neittaanmäki P, Tiba D. An embedding domains approach in free boundary problems and optimal design. *SIAM Journal on Control and Optimization*, 1995; **33**(5):1587–1602.

[142] Netgen Version 4.9.13. Tetrahdral mesh generator developed by J. Schöberl; http://sourceforge.net/projects/netgen-mesher (October 11^{th}, 2011)

[143] Nguyen-Thanh N, Nguyen-Xuan H, Bordas SP, Rabczuk T. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**(21-22):1892–1908.

[144] Nguyen-Thanh N, Kiendl J, Nguyen-Xuan H, Wüchner R, Bletzinger KU, Bazilevs Y, Rabczuk T. Rotation free isogeometric thin shell analysis using PHT-splines. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**(47-48):4310–3424.

[145] Noël AT, Szabó B (1997) Formulation of geometrically non-linear problems in the spatial reference frame. Int J Numer Meth Eng 40:1263–1280

[146] Novelline R (1997) Squire's Fundamentals of Radiology. Harvard University Press, Cambridge

[147] Nübel V, Düster A, Rank E. An rp-adaptive finite element method for elastoplastic problems. *Computational Mechanics*, 2007; **39**:557–574.

[148] Oñate E. Derivation of stabilized equations for numerical solution of advective-diffusive transport and fluid flow problems. *Computer Methods in Applied Mechanics and Engineering*, 1998; **151**(1-2):233–265.

[149] Parussini L, Pediroda V. Fictitious Domain approach with *hp*-finite element approximation for incompressible fluid flow. *Journal of Computational Physics*, 2009; **228**:3891–3910.

[150] Parvizian J, Düster A, Rank E. Finite Cell Method: *h*- and *p*- extension for embedded domain methods in solid mechanics. *Computational Mechanics*, 2007; **41**:122–133.

[151] Parvizian J, Düster A, Rank E. Toplogy Optimization Using the Finite Cell Method. *Optimization and Engineering*, 2011; 10.1007/s11081-011-9159-x

[152] Peskin C. The Immersed Boundary Method. *Acta Numerica*, 2002; **11**:1–39.

[153] Peters J, Reif U. *Subdivision surfaces.* Springer: Heidelberg, 2008.

[154] Piegl LA, Tiller W. The NURBS Book (Monographs in Visual Communication). Springer, New York, 1997.

[155] Ramière I, Angot P, Belliard M. A fictitious domain approach with spread interface for elliptic problems with general boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 2007; **196**:766–781.

[156] Ramm E. *Baustatik II*, Lecture notes, Institute of Structural Mechanics, Universität Stuttgart, 2007. (in German)

[157] Rank E. Adaptive *h*-, *p*- and *hp*-versions for boundary integral element methods. *International Journal for Numerical Methods in Engineering*, 1989; **28**(6):1335–1349.

[158] Rank E. Adaptive remeshing and *h-p* domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 1992. **101**:299–313.

[159] Rank E. A zooming-technique using a hierarchical *hp*-version of the finite element method In: Whiteman J (ed.), *The Mathematics of Finite Elements and Applications - Highlights 1993*, Elsevier: Uxbridge, 1993.

[160] Rank E, Krause R. A multiscale finite-element-method. *Computers & Structures*, 1997. **64**:139–144.

[161] Rank E, Düster A, Nübel V, Preusch K, Bruhns OT. High order finite elements for shells. *Computer Methods in Applied Mechanics and Engineering*, 2005; **194**:2494–2512.

[162] Rank E, Düster A, Schillinger D, Yang Z. The Finite Cell Method: High order simulation of complex structures without meshing. In: Computational Structural Engineering. Yuan Y, Cui J, Mang H (eds), Springer: Heidelberg, 87–92, 2009.

[163] Rank E, Kollmannsberger S, Sorger C, Düster A. Shell Finite Cell Method: A High Order Fictitious Domain Approach for Thin-Walled Structures. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**:3200–3209.

[164] Rank E, Ruess M, Kollmannsberger S, Schillinger D, Düster A. Geometric modeling, Isogeometric Analysis and the Finite Cell Method. Preprint submitted to *Computer Methods in Applied Mechanics and Engineering*, 2011.

[165] Rogers DF. *An Introduction to NURBS with Historical Perspective.* Morgan Kaufmann Publishers: San Francisco, 2001.

[166] Rhino. CAD modeling and design toolkit. www.rhino3d.com (September 24^{th}, 2011)

[167] Rüberg T, Cirak F. Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. Accepted for publication in *Computer Methods in Applied Mechanics and Engineerg*, 2011.

[168] Ruess M, Tal D, Trabelsi N, Yosibash Z, Rank E. The finite cell method for bone simulations: Verification and validation. *Biomechanics and Modeling in Mechanobiology*, 2011; doi:10.1007/s10237-011-0322-2

[169] Sabin M. *Analysis and Design of Univariate Subdivision Schemes.* Springer: Heidelberg, 2010.

[170] Sadd MH. *Elasticity. Theory, Applications, and Numerics.* Academic Press: Oxford, 2009.

[171] Samet H. *The design and analysis of spatial data structures.* Addison Wesley: Reading, 1989.

[172] Samet H. *Foundations of Multidimensional and Metric Data Structures.* Morgan Kaufmann Publishers: San Francisco, 2006.

[173] Sanches R, Bornemann P, Cirak F. Immersed b-spline (i-spline) finite element method for geometrically complex domains. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**: 1432–1445.

[174] Schileo E, Dall'Ara E, Taddei F, Malandrino A, Schotkamp T, Viceconti M. An accurate estimation of bone density improves the accuracy of subject-specific finite element models. *Journal of Biomechanics*, 2008; **41**:2483–2491.

[175] Schillinger D, Kollmannsberger S, Mundani RP, Rank E. The Hierarchical B-Spline Version of the Finite Cell Method for Geometrically Nonlinear Problems of Solid Mechanics. In: *Proc. of the 4th European Conference on Computational Mechanics (ECCM 2010)*, Paris, France, 2010.

[176] Schillinger D, Kollmannsberger S, Mundani R-P, Rank E. The finite cell method for geometrically nonlinear problems of solid mechanics. *IOP Conference Series: Material Science and Engineering*, 2010; **10**:012170.

[177] Schillinger D, Mundani R-P. Finite Element Code Design with Sandia's Library Package Trilinos: Efficient data structures, solver interfaces and parallelization. In: *Tagungsband 22. Forum Bauinformatik 2010*, Technische Universität Berlin, 2010.

[178] Schillinger D, Rank E. An unfitted *hp*-adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**(47-48):3358–3380.

[179] Schillinger D, Düster A, Rank E. The *hp-d* adaptive Finite Cell Method for Geometrically Nonlinear Problems of Solid Mechanics. *International Journal for Numerical Methods in Engineering*, 2011; doi:10.1002/nme.3289.

[180] Schillinger D, Ruess M, Zander N, Bazilevs Y, Düster A, Rank E. Small and large deformation analysis with the *p*- and B-spline versions of the Finite Cell Method. Preprint submitted to *Computational Mechanics*, 2011.

[181] Schillinger D, Ruess M, Düster A, Rank E. The Finite Cell Method for large deformation analysis. *Proceedings in Applied Mathematics and Mechanics*, 2011; **11**:271–272.

[182] Schillinger D, Dedè L, Scott MA, Evans JA, Borden MJ, Rank E, Hughes TJR. An Isogeometric Design-through-analysis Methodology based on Adaptive Hierarchical Refinement of NURBS, Immersed Boundary Methods, and T-spline CAD Surfaces. To appear as *ICES report*, The University of Texas at Austin, January 2012.

[183] Schmidt R, Kiendl J, Bletzinger KU, Wüchner R. Realization of an integrated structural design process: analysis-suitable geometric modelling and isogeometric analysis. *Computing and Visualization in Science*, 2010; **13**(7):315–330.

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

[185] Scott MA, Li X, Sederberg TW, Hughes TJR. Local Refinement of Analysis-Suitable T-splines. *ICES report 11-06*, The University of Texas at Austin, March 2011.

[186] Sederberg TW, Zheng J, Bakenov A, Nasri A. T-splines and T-NURCCSs. *ACM Transactions on Graphics*, 2003; **22**(3):477–484.

[187] Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T. T-spline simplification and local refinement. *ACM Transactions on Graphics*, 2004; **23**(3):276–283.

[188] Sehlhorst HG, Jänicke R, Düster A, Rank E, Steeb H, Diebels S. Numerical investigations of foam-like materials by nested high-order finite element methods. *Computational Mechanics*, 2009; **45**:45–59.

[189] Simo JC, Taylor RL, Wriggers P. A note on finite-element implementation of pressure boundary loading. *Communications in Applied Numerical Methods*, 1991; **7**:513–525.

[190] Simulia. Analysis uses's manual for Abaqus/Standard v6.7. Providence, RI, 2008.

[191] SolidWorks. CAD modeling and design toolkit. www.solidworks.com (September 24^{th}, 2011)

[192] Šolín P, Segeth K, Doležel I. *Higher-order Finite Element Methods.* Chapman&Hall/CRC:Boca Raton, 2004.

[193] Stein E (ed). Error-controlled Adaptive Finite Elements in Solid Mechanics. Wiley: Chichester, 2002.

[194] Sueli E, Mayers DF. *An Introduction to Numerical Analysis.* Cambridge University Press: Cambridge, 2003.

[195] Sukumar N, Chopp DL, Moës N, Belytschko T. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering*, 2001; **190**:6183–6200.

[196] Suri M. Analytical and computational assessment of locking in the *hp* finite element method. *Computer Methods in Applied Mechanics and Engineering*, 1996; **133**:347–371.

[197] Szabó B, Babuška I. *Finite Element Analysis.* Wiley: New York, 1991.

[198] Szabó B, Düster A, Rank E. The *p*-version of the finite element method. In: *Encyclopedia of Computational Mechanics Vol. 1.* Stein E, de Borst R, Hughes TJR (eds), Wiley: Chichester, 119–139, 2004.

[199] Taylor RL. Isogeometric analysis of nearly incompressible solids. *International Journal for Numerical Methods in Engineering*, 2011; **87**(1-5):273–288.

[200] Temizer I, Wriggers P, Hughes TJR. Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**:1100–1112.

[201] Temizer I, Wriggers P, Hughes TJR. Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS. *ICES report 11-16*, The University of Texas at Austin, May 2011.

[202] Trabelsi N, Yosibash Z, Milgrom C. Validation of subject-specific automated *p*-FE analysis of the proximal femur. *Journal of Biomechanics*, 2009; **42**:234–241.

[203] Trilinos Version 10.6. Sandia National Laboratories, http://trilinos.sandia.gov/, Los Alamos, NM, 2011. (November 26^{th}, 2011)

[204] Truesdell C, Noll W. *The Non-Linear Field Theories of Mechanics.* Springer, Berlin, 1992.

[205] T-Splines, Inc. Surface modeling software based on T-splines. www.tsplines.com (September 24^{th}, 2011)

[206] Verhoosel CV, Scott MA, Borden MJ, Hughes TJR, de Borst R. Discretization of higher-order gradient damage models using isogeometric finite elements. *ICES report 11-12*, The University of Texas at Austin, May 2011.

[207] Vinci C. *Application of Dirichlet Boundary Conditions in the Finite Cell Method.* Master Thesis, Lehrstuhl für Computation in Engineering, Technische Universität München, 2009.

[208] Visual DoMesh 2008 v1.1. Mesh generator developed by C. Sorger, Chair for Computation in Engineering, Technische Universität München.

[209] Vos PEJ, van Loon R, Sherwin SJ. A comparison of fictitious domain methods appropriate for spectral/*hp* element discretisations. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:2275–2289

[210] Vuong AV, Giannelli C, Jüttler B, Simeon B. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 2011. **200**(49-52):3554–3567.

[211] Wall WA, Frenzel MA, Cyron C. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:2976–2988.

[212] Warren J, Weimer H. *Subdivision Methods for Geometric Design.* Morgan Kaufman Publishers: San Francisco, 2002.

[213] Wenisch P, Wenisch O. *Fast octree-based voxelization of 3D boundary representation-objects.* Technical report, Lehrstuhl für Bauinformatik, Technische Universität München, 2004.

[214] Wriggers P. *Nonlinear Finite Element Methods.* Springer: Berlin, 2008.

[215] Yosibash Z, Hartmann S, Heisserer U, Düster A, Rank E, Szanto M. Axisymmetric pressure boundary loading for finite deformation analysis using *p*-FEM. *Computer Methods in Applied Mechanics and Engineering*, 2007; **196**:1261–1277.

[216] Yosibash Z, Padan R, Joskowicz L, Milgrom C. A CT-based high-order finite element analysis of the human proximal femur compared to in-vitro experiments. *Journal of Biomechanical Engineering*, 2007; **129**:297–309.

[217] Yserantant H. On the multi-level splitting of finite element spaces. *Numerische Mathematik*, 1986; **49**:379–412.

[218] Zander N. *The Finite Cell Method for Linear Thermoelasticity.* Master Thesis, Lehrstuhl für Computation in Engineering, Technische Universität München, 2011.

[219] Zhang L, Gerstenberger A, Wang X, Liu WK. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 2004; *193*:2051–2067.

[220] Zhu T, Atluri SN. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. *Computational Mechanics*, 1998; **21**:211–222.

[221] Zienkiewicz OC, Taylor RL. *The Finite Element Method Vol. 1: The Basis.* Butterworth-Heinemann: Oxford, 2005.

[222] Zienkiewicz OC, Taylor RL . *The Finite Element Method Vol. 2: Solid Mechanics.* Butterworth-Heinemann: Oxford, 2005.

[223] Zienkiewicz OC, Taylor RL. *The Finite Element Method Vol. 3: Fluid Dynamics.* Butterworth-Heinemann: Oxford, 2005.

[224] Zohdi T, Wriggers P. Aspects of the computational testing of the mechanical properties of microheterogeneous material samples. *International Journal for Numerical Methods in Engineering*, 2001; *50*:2573–2599.

[225] Zohdi T, Wriggers P. *Introduction to Computational Micromechanics.* Springer: Berlin, 2005.

[226] Zorin D, Schröder P, DeRose T, Kobbelt L, Levin A, Sweldens W. *Subdivision for Modeling and Animation.* SIGGRAPH course notes, 2000.