Ingenieurfakultät Bau Geo Umwelt
Lehrstuhl für Computation in Engineering

Multi-level *hp*-FEM: dynamically changing high-order mesh refinement with arbitrary hanging nodes

Nils Dietrich Zander, M.Sc.

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

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. habil. Fabian Duddeck
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. rer. nat. Ernst Rank
	2.	Prof. Zohar Yosibash, D.Sc. Ben-Gurion University of the Negev, Israel
	3.	Prof. Alessandro Reali, Ph.D. University of Pavia, Italien

Die Dissertation wurde am 27.10.2016 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 03.02.2017 angenommen.

Zusammenfassung

In den letzten Dekaden wurden die mathematischen Eigenschaften der *hp*-Version der Finiten Elemente Methode ausführlich untersucht. Hierbei wurde unter anderem bewiesen, dass diese Diskretisierungsmethode sogar bei singulären Problemen eine exponentielle Konvergenz erzielt. Trotz der hieraus resultierenden Vorteile ist diese Methode in der heutigen Ingenieurspraxis nicht weit verbreitet. Ein Grund hierfür ist das Problem der sogenannten hängenden Knoten, das zu einer hohen Komplexität in der Formulierung und Implementierung der Methode führt.

Das Ziel der in der vorliegenden Arbeit vorgestellten Multi-Level *hp*-Methode ist das Problem der hängenden Knoten zu lösen, ohne dabei die hohe Lösungsqualität von *hp*-Methoden zu senken. Hierzu wird die klassischerweise eingesetzte *refine-by-replacement* Verfeinerungsstrategie durch einen *refine-by-superposition* Ansatz ersetzt. Die Multi-Level *hp*-Methode setzt dieses alternative Verfeinerungskonzept ein, indem ein grobes high-order Grundnetz mit einem feinen high-order Overlaynetz superpositioniert wird. Diese alternative *hp*-Formulierung umgeht das Problem der hängenden Knoten, da die globale Kontinuität der Lösung mittels homogener Dirichlet-Randbedingungen auf dem Overlaynetz sichergestellt werden kann.

Diese Arbeit zeigt, dass sich durch diesen Paradigmenwechsel die Formulierung der Methode deutlich vereinfacht. Hierdurch ist eine computergestützte Umsetzung in einem flexiblen Diskretisierungskernel möglich, der eine dynamische Veränderung der Netzstruktur und beliebige Netzirregularitäten erlaubt. Durch eine umfassende Testreihe von ein-, zwei- und dreidimensionalen singulären Benchmarks aus dem Gebiet der Wärmeleitung, der Elastizität, der konvektionsdominierten Transportprobleme und der Kontaktmechanik wird gezeigt, dass die Multi-Level *hp*-Methode die gleiche Approximationsgenauigkeit wie konventionelle *hp*-Ansätze erzielt. Des Weiteren wird gezeigt, dass es die vereinfachte Implementierung der Methode erlaubt, die Netzstruktur kontinuierlich während der Laufzeit der Simulation zu verändern. Auf diese Weise kann das Verfeinerungsgebiet dem Verlauf von bewegten Irregularitäten folgen. Die sich hieraus ergebenden Vorteile werden im Kontext von sich dynamisch ändernden Randbedingungen, kohäsiver Delamination, sich bewegenden Einschlüssen und Kontaktproblemen mit großen Deformationen demonstriert.

Abstract

The *hp*-version of the Finite Element Method has proven to yield an exponential convergence of the approximation error even for problems with singular solution characteristics. Despite the resulting possible advantages, *hp*-FEM has not become the standard computational method in today's engineering practice. This has to be partially attributed to the problem of hanging nodes, which yields a high complexity of the method's formulation and implementation.

By proposing the multi-level *hp*-concept, the thesis at hand aims at reducing this complexity, while yielding the same high approximation quality as conventional *hp*-approaches. To this end, the conventional *refine-by-replacement* idea is dropped in favour of a *refine-by-superposition* approach. The idea of the multi-level *hp*-method is to apply this second refinement technique by superposing a coarse base discretization with a fine, high-order overlay mesh in domains of interest. This alternative *hp*-formulation fully circumvents the problem of hanging nodes since the global continuity can be ensured by applying homogeneous Dirichlet boundary conditions on the boundary of the overlay mesh.

This work demonstrates that this change of refinement-paradigm significantly reduces the complexity of the approach, such that the formulation can be implemented in a flexible discretization kernel, which naturally allows for dynamically changing, arbitrary irregular meshes. By means of an extensive test-suite of one-, two-, and three-dimensional singular benchmarks from the field of heat conduction, elasticity, convection dominated transport problems, and contact mechanics, it is demonstrated that the multi-level *hp*-FEM yields the same high approximation quality as conventional *hp*-formulations. Furthermore, it is shown that the dynamic discretization capabilities of the method allow the refinement to stay local to moving sharp solution characteristics by continuously updating the discretization throughout the course of the simulation. The advantages resulting from this flexibility are demonstrated in the context of dynamically changing boundary conditions, cohesive delamination, moving inclusions, and computational contact mechanics.

Preface

This thesis was created during my employment at the Chair for Computation in Engineering at the Technische Universität München in the time from 2011 to 2016. The work was supported by the research project *Electro-thermo-mechanical modeling of Field Assisted Sintering Technology using high order finite elements validated by experiments* of the Deutsche Forschungsgemeinschaft (DFG) under grant RA 624/19. I thank all institutes and people with whom I collaborated and who supported me during that time.

In particular, I thank my doctoral supervisor Prof. Ernst Rank for his great support during my work and the numerous opportunities he offered me to develop both, my scientific know-how and my personal skills. The trust he put in me combined with the personal freedom he offered created a fruitful and enjoyable working climate. With his continuous support and his guidance, he substantially contributed to the success of this thesis.

I also thank Prof. Zohar Yosibash, who took part in my research starting with the co-supervision of my master's thesis. The DFG project that followed as well as regular research visits in Munich and meetings at conferences gave the chance for many interesting discussions that pushed my research in the right direction. I always enjoyed talking with Zohar and I am more than happy to have him as part of my PhD committee.

Furthermore, I thank Prof. Alessandro Reali, whom I got to know at the beginning of my time as a PhD research assistant. Right from day one, talking to Alessandro always was a pleasure and his input shed light on aspects that I had not seen before. For this reason, I am grateful that Alessandro is part of my PhD committee as well.

I also thank Prof. Fabian Duddeck for chairing the examination of this thesis.

Special thanks also go to Dr. Stefan Kollmannsberger, who was part of my research right from day one. He always guided my research in the right direction and gave me the freedom to develop and explore my ideas. With his congenial character, he provided a friendly atmosphere in which I always enjoyed working in. His continuous support substantially contributed to the success of this thesis.

I also thank Prof. Martin Ruess not only for his continuous input during my time as a research assistant but in particular for his support and the hospitality he offered me during my research visit at the TU Delft. I always enjoyed working with Martin and our collaboration in Delft pushed the numerical method discussed in this thesis to a new field of application.

I further thank Prof. Dominik Schillinger for his support throughout the past six years. Working with and talking to Dominik always was a pleasure, gave valuable input leading my research, and pushed me to improve my scientific skills.

Furthermore, I thank all colleagues with whom I had the pleasure to work with in the past years for their inputs and their nice company. In particular, I express my thanks to (in alphabetical order) Davide D'Angella, André Borrmann, Mohamed Elhaddad, Matthias Flurl,

Jérôme Frisch, John Jomo, Angelika Keidl, Jovana Kneževic, László Kudela, Phillip Kopp, Nina Korshunova, Ralf-Peter Mundani, Ali Özcan, Iason Papaioannou, Nevena Perovic, Martin Schlaffer, Christian Sorger, and Vasco Varduhn. I further thank Hanne Cornils for her friendly character and her support with all kind of administrative questions. Special thanks go to Hagen Wille and Tino Bog. I enjoyed all the scientific and non-scientific discussions we had and the activities and experiences we shared during work, on travel and outside of the office. Their great humour made me enjoy coming into office every day.

Furthermore, I thank my family and friends. In particular, I thank my parents for their patient support especially in my school time and in finding the right direction. Only their continuous help allowed my studies and the success of this work. I also thank Jill and Pat for welcoming me in their family during my year in Leeds and beyond that time. This first experience abroad was the starting point for my international studies and research.

Last, but definitely not least, I thank my wife Sandra. For over ten years, we now go through thick and thin together. Only your unconditional and invaluable help, support, and love during this time as well as your understanding for my "time management" allowed me to complete my academic studies. Without you, this thesis would not have been possible. I love you.

Nils Zander
Nuremberg, April 2017

Contents

Notation and glossary

This work follows the standard notation of functional analysis used in e.g. [Brenner and Scott, 2008; Ciarlet, 2002; Reddy, 1997]. In the context of continuum mechanics, a slightly altered version of the notation suggested by Holzapfel [2000] is used: scalar values are expressed in normal font (e.g. a, β, c), and tensors are denoted in bold font (e.g. $\boldsymbol{u}, \boldsymbol{\sigma}, \boldsymbol{C}$). However, in contrast to [Holzapfel, 2000], no distinction between the order of the tensor is made. Individual components/coordinates of tensors are denoted using the index notation, e.g. $u_i, \sigma_{ij}, C_{ijkl}$. Assuming a Cartesian basis, the basis vectors are omitted in this notation, and both, co- and contravariant basis are labeled in with sub-scripts [Holzapfel, 2000, Section 1.6]. Furthermore, Einstein's summation convention is used, e.g. $t_i = \sigma_{ij} n_j$.

Throughout the work, the notation, abbreviations and symbols are described when first used. Abbreviations and symbols used across chapters are summarized in following.

Abbreviations

CAD	Computer Aided Design.
dist.	Distribution.
dof	Degree(s) of freedom.
elem.	Elements/number of elements.
elev.	Elevation.
FCM	Finite Cell Method.
FEM	Finite Element Method.
IGA	Isogeometric Analysis.
lin.	Linear.
m.-l. *hp*	multi-level *hp*.
NURBS	Non-uniform rational B-splines.
PDE	Partial Differential Equation.
ref.	Refinement.
SEM	Spectral Element Method.
SUPG	Streamline upwind Petrov-Galerkin.
uni.	Uniform.

Functional analysis

$a(\cdot, \cdot)$	Bi-linear form.
$\lceil \cdot \rceil, \lfloor \cdot \rfloor$	Rounding operators ceil and floor, respectively.
C^m	Space of functions with a continuous m-th derivative.
$f(\cdot)$	Linear form.
$\tilde{f}(\cdot)$	Linear form shifted by Dirichlet boundary condition.
H^m	Sobolev space of order m.
H_0^m	Homogeneous Sobolev space of order m.
L_2	Space of square integrable functions.
$\|\|v\|\|$	Norm of function v.
$\|\|v\|\|_E$	Energy norm of function v.
$\|\|v\|\|_{H^1}$	H^1 norm of function v.
$\|\|v\|\|_{L_2}$	L_2 norm of function v.
$\|\|v\|\|_{\mathcal{V}}$	Norm of function v with respect to inner product space $\mathcal{V}$.
$\mathbb{R}^n$	n-dimensional space of real numbers.
$\mathcal{S}$	Admissible function space complying to the Dirichlet boundary conditions.
$\mathcal{V}$	Admissible functions space complying to homogeneous Dirichlet boundary conditions.

Continuum mechanics

$\boldsymbol{b}$	Left Cauchy-Green tensor.
$\hat{\boldsymbol{b}}$	Prescribed body load in deformed configuration.
$\hat{\boldsymbol{B}}$	Prescribed body load in initial configuration.
$\mathbb{C}$	Elastic material tensor.
$\boldsymbol{C}$	Right Cauchy-Green tensor.
$\nabla \cdot \boldsymbol{u}$	Divergence of a tensor $\boldsymbol{u}$.
$\boldsymbol{u} \cdot \boldsymbol{v}, u_i v_i$	Dot/inner product of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$.
$\boldsymbol{\sigma} : \boldsymbol{\varepsilon}, \sigma_{ij}\varepsilon_{ij}$	Dot/inner product of two higher-order tensors $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$.
$\boldsymbol{e}$	Euler-Almansi strain tensor.
$\boldsymbol{E}$	Green-Lagrange strain tensor.
$\boldsymbol{\varepsilon}$	Engineering strain tensor.
$\boldsymbol{f_n}$	Flux over boundary.
$\boldsymbol{f}$	Flux tensor.
$\boldsymbol{F}$	Deformation gradient.
Γ	Domain boundary.

Γ_D	Dirichlet boundary in initial configuration.
γ_D	Dirichlet boundary in deformed configuration.
Γ_N	Neumann boundary in initial configuration.
γ_N	Neumann boundary in deformed configuration.
G_{ext}	External virtual work.
G_{int}	Internal virtual work.
J	Determinant of deformation gradient $\boldsymbol{F}$.
$\boldsymbol{\kappa}$	Conductivity tensor.
$\boldsymbol{n}$	Outward pointing unit normal vector in deformed configuration.
$\boldsymbol{N}$	Outward pointing unit normal vector in initial configuration.
Ω, Ω_0	Control volume or domain in initial configuration.
Ω_c	Control volume or domain in deformed configuration.
$\boldsymbol{P}$	First Piola stress tensor.
$\boldsymbol{\varphi}$	Deformation mapping.
$\hat{\boldsymbol{\varphi}}$	Prescribed deformation mapping at Dirichlet boundary.
$\boldsymbol{q}$	Cauchy heat flux vector.
$\hat{\boldsymbol{q}}$	Applied flux on Neumann boundary (e.g. heat flux or traction).
$\hat{\boldsymbol{s}}$	Generic source term.
$\hat{s}$	Heat source.
$\boldsymbol{S}$	Second Piola stress tensor.
$\boldsymbol{\sigma}$	Cauchy stress tensor.
$\boldsymbol{t}$	Cauchy traction vector.
$\hat{\boldsymbol{t}}$	Prescribed Cauchy traction vector on Neumann boundary.
$\boldsymbol{T}$	Piola traction vector.
$\hat{\boldsymbol{T}}$	Prescribed Piola traction vector on Neumann boundary.
$(\cdot)^\top$	Transpose of a vector or tensor.
W	Hyper-elastic energy potential.
$\boldsymbol{x}$	Spatial point in deformed configuration.
$\boldsymbol{X}$	Material point in initial configuration.

Finite element discretization

$\boldsymbol{A}$	Stiffness matrix.
A_i	Magnitude of singular solution expansion in polar coordinates.
α	Irregularity exponent.
I_i	i-th interval in a one-dimensional mesh.
β	Algebraic convergence rate.

c_i	Midpoint of i-th element.
e	Approximation error $u - u_h$.
$\boldsymbol{F}$	Right-hand-side or load vector.
$\nabla_{\xi^k}(\cdot)$	Gradient with respect to the reference coordinates of refinement level k.
h	Element size.
h_i	Size of i-th element.
$\boldsymbol{J}_k$	Jacobi matrix of refinement level k.
$\boldsymbol{H}^k$	Hessian matrix of refinement level k.
k	Refinement level or number of refinement levels.
K	Cell or element of a mesh $\mathcal{T}_h$.
$\overset{\circ}{K}$	Interior of a cell K.
L_i	Legendre polynomial of order i.
$\boldsymbol{LM}$	Location map.
N	Basis function of finite element space.
$\boldsymbol{N}$	Basis function matrix of finite element space.
p	Polynomial degree of element shape functions.
P_i	Integrated Legendre polynomial of order i.
P_v	Patch associated to node, edge, or face of a mesh $\mathcal{T}_h$.
p_i	Polynomial degree of element shape functions of i-th element.
$\boldsymbol{\Psi}_K$	Geometric mapping of reference element onto element K.
q	Geometric grading ratio.
$\boldsymbol{r}, (r, s, t)$	Spatial point in reference element.
r_i	Mesh design factor of i-th element.
s	Slope of order elevation in graded hp-mesh.
$\mathcal{S}_h$	Finite element sub-space of $\mathcal{S}$.
ξ^K	Shape function of element K.
ξ_j^K	j-th shape function of element K.
$\boldsymbol{\pi}^K$	Connectivity matrix of element K.
$\boldsymbol{\xi}^{\hat{K}}$	Set of element shape functions of element K.
$\hat{\xi}^{\hat{K}}$	Standard shape function of element K.
$\hat{\boldsymbol{\xi}}^{\hat{K}}$	Set of standard shape functions of element K.
Θ	Exponential convergence exponent.
$\boldsymbol{u}$	Analytical solution function.
$\boldsymbol{u}_h$	Numerical approximation of analytical solution $\boldsymbol{u}$.
$\boldsymbol{u}_b$	Base mesh solution.

$\boldsymbol{u}_o$	Overlay solution.
$\boldsymbol{\Upsilon}$	Geometric mapping from refined to coarse interface.
$\boldsymbol{v}$	Test function.
$\mathcal{V}_h$	Finite element sub-space of $\mathcal{V}$.

Error control

C_E	Relative error threshold for element refinement.
C_S	Smoothness threshold for h- and p-refinement.
η	Estimate of global approximation error.
η_K	Error indicator of element K.
r	Residual.
Θ	Effectivity index of error estimator.

Finite Cell Method

α	Indicator function.
Ω_{phy}	Physical domain.
Ω_{fict}	Fictitious domain.
$\Omega_{\cup}$	Union of physical and fictitious domain.

Cohesive crack growth

δ	Gap opening at limit strength of cohesive bond.
Γ_C	Potential crack path.
$\boldsymbol{t}(\llbracket \boldsymbol{u} \rrbracket)$	Cohesive traction in crack interface.
$\boldsymbol{t}_{\text{ult}}$	Limit strength of cohesive bond.
$\llbracket \boldsymbol{u} \rrbracket$	Gap opening.

Convection diffusion problems

$\boldsymbol{a}$	Convective velocity vector.
P_e	Element Péclet number.
τ_h	SUPG stabilization parameter for h-FEM.
τ	SUPG stabilization parameter for p-FEM.

Contact Mechanics

ϵ	Penalty parameter.
g	Normal contact gap (positive in penetration).

δg	Virtual gap function.
G_{c}	Virtual work of contact.
$\hat{K}^{\mathrm{ext}}$	Extended reference space of element K.
$\boldsymbol{\nu}$	Outward pointing unit normal on deformed master contact interface.
p_N	Contact pressure.
$\boldsymbol{X}$	Contact point on slave contact interface.
$\bar{\boldsymbol{Y}}$	Contact point on master contact interface.

Chapter 1

Introduction

1.1 Motivation

In the past decades, the accurate numerical simulation of complex physical phenomena has been key to expanding technological frontiers. In particular in the fields of mechanical, structural and civil engineering, the Finite Element Method (FEM) has become the method of choice in today's engineering practice. The core principle of this discretization approach is to partition the geometry under consideration into a finite number of elements and to represent the numerical approximation of the analytical solution by so-called shape functions defined on these elements. This element-wise approach allows for the use of complex, unstructured meshes, which gives the method a high geometric flexibility. Moreover, continuous research in this field has extended the applicability of the FEM to almost every field of engineering interest including solid and fluid mechanics, contact, friction, non-linear deformations and material models, fluid structure interaction, bio-mechanics or dimensionally-reduced structures such as beams, plates and shells.

Despite this great success, one constant request of engineering practice is to simulate *larger* numerical models with *higher* accuracy in a *shorter* time. The classical approach to increase the accuracy is the use of finer meshes with decreasing element sizes. Although this so-called h-refinement is prevalent and has proven to yield excellent results, the obtained numerical approximation converges to the analytical solution at a comparably slow rate. This does not pose a problem when applying h-refinement to obtain a first rough estimate or in applications with high modeling uncertainties. However, in situations requiring highly accurate solutions, the resulting high number of unknowns necessary for the simulation renders a pure h-refinement computationally unfeasible.

An alternative strategy to increase the result quality is to use shape functions of higher approximation order. Classical approaches of this kind are the p-version of the Finite Element Method (p-FEM) and the Spectral Element Method (SEM). In both cases, tailored high-order polynomials are used to approximate the analytical solution. The main advantage of these approaches is that—for smooth problems—the approximation error decays exponentially under an elevation of the approximation order. In this way, the same approximation accuracy can be reached at a significantly smaller number of unknowns compared to classical h-refinement.

In the past decade, the family of high-order methods has been extended by the concept of Isogeometric Analysis (IGA). Here, (high-order) B-splines or non-uniform rational B-splines

(a) Crack in steel structure [Mann, 2011]

(b) In-plane wall cracking due to Killari earthquake in 1993 [WHE, 2016]

Figure 1.1: Examples of failure initiated at geometric features.

(NURBS)—common in the world of Computer Aided Design (CAD)—are used to represent the numerical solution. Compared to the aforementioned high-order methods, the advantage of this approach is the higher inter-element continuity of up to C^{p-1}, with p being the order of the used splines. In contrast, p-FEM and SEM elements are only C^0-continuous. This higher global continuity renders the method applicable to higher-order variational problems, such as e.g. shells and higher-order phase fields. Furthermore, it provides a higher accuracy per degree of freedom compared to the previous methods, when applied to smooth problems.

Unfortunately, the presence of singularities or stress concentrations has a detrimental effect on the performance of the aforementioned high-order methods. In this case, the convergence decays to an algebraic characteristics that is only slightly better than the convergence of a pure h-refinement.

Such non-smooth characteristics are a common problem in the field of computational mechanics and can be causes by e.g. re-entrant corners, small geometric features, crack-tips, material interfaces, changing boundary conditions, or contact boundaries. Unfortunately, these points often initiate the failure of a mechanical structure as illustrated in Figure 1.1. Therefore, the numerical prediction is required to be as accurate as possible in these regions to prevent the possibly catastrophic outcome.

An alternative refinement strategy—tailored for these kind of problems—is the hp-version of the Finite Element Method. The essential idea of the approach is to obtain higher accuracy by combining the decrease of the element size with the use of high-order polynomials as shape functions. In this way, the finite element mesh can be refined in the vicinity of the singularities, whereas a coarse, high-order discretization can be used to approximate the analytical solution in domains where it is smooth. A detailed mathematical analysis of this methods reveals that the simultaneous change of h and p yields an exponential decrease of the approximation error even in the presence of singularities. This allows for a significantly higher accuracy for the same number of unknowns.

However, the hp-method has not become the standard in engineering simulations. This has to be at least partially attributed to the high implementational complexity of this refinement

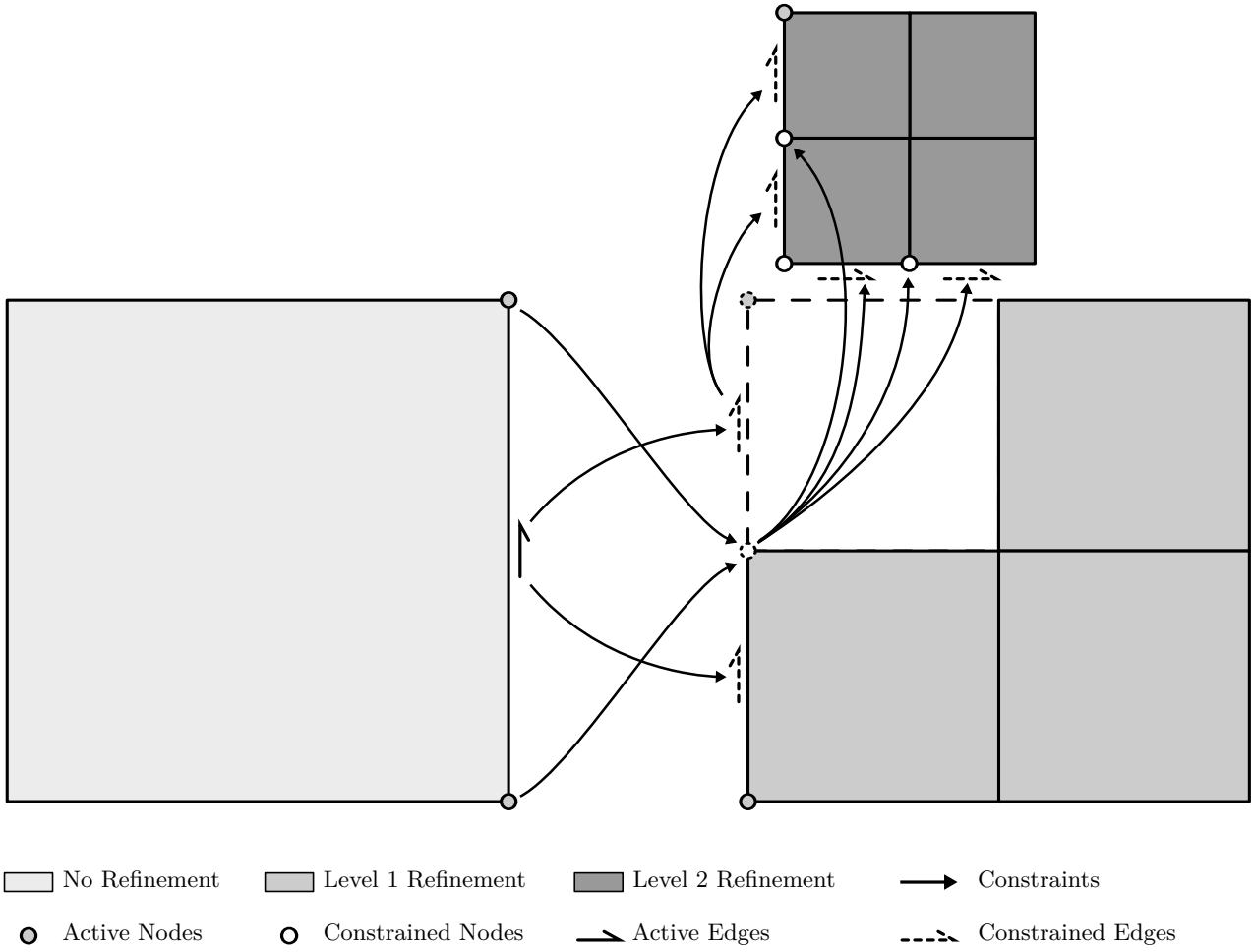


Figure 1.2: Multiply constrained modes occurring in classical *hp*-refinement strategies [Zander et al., 2015].

method. One cause of this complexity is the inherent idea of the approach to perform *h*-refinement by *replacing* coarse elements by a set of finer elements. As illustrated in Figure 1.2, the resulting fine-scale nodes have no corresponding counterpart in the neighboring coarse element. These so-called *hanging* nodes render the discretization *irregular*.

One possibility to circumvent this problem is to additionally sub-divide the adjacent coarse elements to form a regular transition between the refined and coarse parts of the mesh. Although this approach has proven to yield good results, the automatic creation of these transition elements is challenging, in particular in higher dimensions.

An alternative approach is to directly use the irregular discretization for the finite element computation. To this end, the hanging nodes have to be constrained appropriately such that the final approximation is C^0-continuous. Although the applicability of this constrained approximation approach has been shown in various contexts, it comes at a high algorithmic complexity. This applies in particular to cases where hanging nodes have to be constrained in terms of already constrained nodes. The resulting cascade of constraints has only been implemented in very few code frameworks, which are mostly used for research purposes. Fur-

thermore, in applications evolving e.g. moving loads, evolving cracks, propagating shocks, or sliding contact, the position of a stress concentration or singularity might change during the course of the simulation. Therefore, the mesh design has to be updated accordingly to accurately capture the new solution field. When following the prevalent refine-by-replacement approach, this dynamic change demands a sophisticated discretization kernel.

This rationale motivates the objectives presented in the next section.

1.2 Objectives

The aim of the *multi-level hp*-version of the Finite Element Method—described in this work—is to formulate a refinement scheme that a) alleviates the aforementioned algorithmic and implementational complexity of hanging nodes and b) yields the same approximation quality as conventional *hp*-methods.

Point a) requires that the rule-set ensuring the linear independence and the compatibility of the basis functions is as simple as possible such that it is implementable in a flexible data structure. Concerning point b), the primary goal is that the refinement scheme yields an exponential decay of the approximation error in the presence of singularities. Here, the convergence is not only to be understood with respect to the number of unknowns but also with respect to the time required to compute the solution. Furthermore, the refinement shall not be limited to simple benchmarks but also applicable for three-dimensional unstructured meshes. Moreover, the scheme should be compatible with classical error and smoothness estimation techniques to yield an automatic, error-driven *hp*-adaptive scheme. The final aspect is that the discretization kernel should allow for a dynamic change of the mesh structure during the simulation such that moving singularities can be resolved with high accuracy.

To meet these demands, the classic refine-by-replacement idea is abandoned. Instead, the multi-level *hp*-scheme refines the discretization by *superposing* the base mesh with a finer overlay mesh in domains of interest. By this change of paradigm, the inter-element continuity of the shape functions can be ensured easily by applying homogeneous Dirichlet boundary conditions on the boundary of the refinement zone. In this way, hanging nodes are avoided by definition, which reduces the implementational complexity of the scheme. This allows for a highly flexible discretization kernel featuring arbitrary irregular meshes, which can be refined and coarsened continuously throughout the simulation runtime.

In accordance with the TUM Promotionsordnung [TUM, 2010, §6, Abs. 1, Satz 2], the idea of the multi-level *hp*-method has been pre-published. The refinement scheme was first introduced for two-dimensional problems in [Zander et al., 2015] and extended to three-dimensional problems in [Zander et al., 2016]. In [Zander et al., 2017], the applicability of this method for the accurate resolution of cohesive fracture has been demonstrated. The combination of the multi-level *hp*-refinement scheme with error and smoothness estimators was presented in [D'Angella et al., 2016]. The main scientific research as well as the textual elaboration of the first three publications was performed by the author of this work. The fourth publication has been co-authored by the author of this work. In accordance with the Wiley-VCH Copyright Transfer Agreement [Wiley, 2016], the Springer Copyright Transfer Statement [Springer, 2016], and the Elsevier publishing agreement [Elsevier, 2016], the current work summarizes these scientific results and supplements them with a more detailed description of the context and the background. Furthermore, the analysis of the multi-level *hp*-refinement scheme is

extended by additional, more detailed numerical studies to assess the approximation quality of the suggested method in more depth. Throughout the work, the following footnotes are used to reference the original publications and to mark literal transposition. [a] [b] [c] [d] [e] [f] [g] [h]

1.3 Outline

This work is organized in two parts. The first part formulates the multi-level *hp*-version of the Finite Element Method. In the second part, the applicability of multi-level *hp*-method is analyzed in the context of different problem classes of engineering interest. The content of the individual chapters is summarized in the following.

Part I: Formulation of multi-level *hp*-FEM

In **Chapter 2**, the fundamentals of the Finite Element Method are outlined. The main purpose of the chapter is to introduce and define the terminology used throughout this work. To this end, the first part of the chapter describes the mathematical models considered in this work in their strong and variational/weak form. Furthermore, the requirements for the existence and uniqueness of an analytical solution are reviewed. The second part of the chapter describes the fundamentals of the Finite Element Method. To this end, the idea of the Galerkin method is outlined and the most important approximation properties are stated. On this basis, the construction of finite element function spaces is outlined. In this context, the respective terminology—such as "element", "mesh", or "shape function"—is defined. The chapter closes by outlining the proof that the finite element approximation converges to the exact solution. The requirements necessary for this convergence are interpreted in an engineering context.

In **Chapter 3**, the different versions of the Finite Element Method and their convergence properties are discussed. As a motivation, the chapter begins with a detailed convergence analysis for a one-dimensional, singular benchmark. For this example, the *a priori* estimation

[a] The following chapter/section/paragraph is based on [Zander et al., 2015]. The main scientific research as well as the textual elaboration of the publication was performed by the author of this work.

[b] The following chapter/section/paragraph is based on [Zander et al., 2016]. The main scientific research as well as the textual elaboration of the publication was performed by the author of this work.

[c] The following chapter/section/paragraph is based on [Zander et al., 2017]. The main scientific research as well as the textual elaboration of the publication was performed by the author of this work.

[d] The following chapter/section/paragraph is based on the Master's thesis [D'Angella, 2015] that was supervised by the author of this work. The resulting publication [D'Angella et al., 2016] was co-authored by the author of this work.

[e] In accordance with the Elsevier publishing agreement [Elsevier, 2016], the text of this chapter/section/paragraph is taken in an adjusted version from [Zander et al., 2014]. The main scientific research as well as the textual elaboration of the publication was performed by the author of this work.

[f] In accordance with the Springer Copyright Transfer Statement [Springer, 2016], the text of this chapter/section/paragraph is taken in an adjusted version from [Zander et al., 2015]. The main scientific research as well as the textual elaboration of the publication was performed by the author of this work.

[g] In accordance with the Elsevier publishing agreement [Elsevier, 2016], the text of this chapter/section/paragraph is taken in an adjusted version from [Zander et al., 2016]. The main scientific research as well as the textual elaboration of the publication was performed by the author of this work.

[h] In accordance with the Wiley-VCH Copyright Transfer Agreement [Wiley, 2016], the text of this chapter/section/paragraph is taken in an adjusted version from [Zander et al., 2017]. The main scientific research as well as the textual elaboration of the publication was performed by the author of this work.

of the approximation error is stated, and different mesh refinement types are compared. In the second part of the chapter, these results are cast into the different versions of the Finite Element Method. In particular, the *p*- and *hp*-version of the method are outlined in detail. In the final part, the approximation properties of these finite element formulations known from literature are summarized and compared.

In **Chapter 4**, the classical construction of *hp*-meshes is reviewed. The chapter focuses on the question of how to resolve the aforementioned mesh irregularities caused by the classical refine-by-replacement approach. To this end, the chapter begins with a detailed literature review of the different strategies developed in the past decades. In the second part, the prevalent idea of a constrained approximation is outlined in more detail. For this purpose, the derivation of the constraint coefficients and their assembly in the final stiffness matrix is discussed. The chapter closes by pointing out the common algorithmic limits of this approach.

Motivated by these challenges, the suggested multi-level construction of *hp*-meshes is discussed in **Chapter 5**. To this end, the chapter begins with formalizing the applied refine-by-superposition idea. In a second step, a literature survey provides an overview of the different contexts in which this approach has been applied successfully before. The final part of the chapter then describes the use of this idea to construct hierarchical *hp*-meshes wherein hanging nodes need no special consideration. In particular, a simple rule-set to ensure the linear independence and the compatibility of the final function space is derived.

In **Chapter 6**, the implementational aspects of the multi-level *hp*-method are addressed. To this end, the chapter begins with the presentation of a possible data structure in the context of an object-oriented finite element framework. The properties of the core classes are derived, and their interaction is discussed. On this basis, the second part of the chapter describes the hierarchical refinement procedure in detail. The essential steps of the algorithm are illustrated in pseudo code. In the third part of the chapter, the implementation of the rule-set for a linear-independent and compatible basis is discussed by a means of a full outline of the algorithm in terms of pseudo code. The chapter concludes with details on how the shape functions are evaluated in the hierarchical mesh structure, and how the final stiffness matrix can be formed.

With **Chapter 7**, the first part of this work closes by describing an error controlled multi-level *hp*-adaptive refinement approach. To this end, the chapter first outlines the essential idea and properties of an adaptive feedback technique for the construction of finite element meshes. From this idea, the need of an error and smoothness estimator is derived. For this reason, the first and second part of the chapter provide a literature review of the different techniques usable for estimating the error and the smoothness of a numerical approximation in a cheap and effective way. In the third and final part of this chapter, these two techniques are combined to formulate an error-controlled multi-level *hp*-adaptive refinement scheme. To this end, the chapter outlines how the selected estimation techniques have to be reinterpreted to estimate the approximation error and the smoothness in a hierarchical mesh structure.

Part II: Application of multi-level *hp*-FEM

The second part of this work begins with **Chapter 8**, where the approximation properties of the suggested multi-level *hp*-method are analyzed in the context of problems with singular or non-smooth solutions. The test suite includes the classical benchmarks, such as the L-shaped domain, the Fichera corner problem, the cracked panel problem, and the three-dimensional shock problem. For these applications, the convergence of the approximation error is analyzed

with respect to the number of unknowns and the computational time required for the simulation. In this context, different mesh design strategies and the error-controlled refinement algorithm are compared. In addition to the approximation error, other numerical properties of the multi-level *hp*-method—such as the condition number or the number of necessary solver-iterations—are studied and compared to references obtained by conventional *hp*-approaches. Besides these simple-shaped benchmarks, the refinement scheme is further tested in the context of a curved, three-dimensional structure to analyze the applicability of the approach for problems of engineering relevance.

Having assessed the properties of the multi-level *hp*-approach in the presence of a stationary singularity, **Chapter 9** analyzes the ability of the refinement scheme to dynamically adapt the discretization during the runtime of the simulation. To this end, the scheme is applied to resolve a propagating wave front, while keeping the refinement local to the sharp gradients. In a second benchmark, the potential of the method is tested in the context of a discontinuous, traveling traction, which induces a moving discontinuity in the stress field. The chapter closes by extending this benchmark to three dimensions.

Chapter 10 studies potential benefits of combining the suggested refinement scheme with the Finite Cell Method (FCM) to capture moving inclusions on a fixed, non-matching background mesh. To this end, the essential idea of the Finite Cell Method is outlined in the first part of this chapter. In the second part, the combination of the two methods is analyzed for a two- and a three-dimensional problem.

In **Chapter 11**, the performance of the refinement scheme is studied in the context of progressing, cohesive crack growth. For this purpose, the chapter begins with a brief introduction into the mechanical and numerical modeling of cohesive fracture and outlines the use of the multi-level *hp*-refinement method in this context. In the second part of this chapter, the approximation quality of the *hp*-refined discretization is analyzed for the well-known double cantilever beam benchmark. In a second example, the suggested refinement method is used to resolve the progressing crack front in a three-dimensional setting. In the last example, the multi-level *hp*-approach is combined with the Finite Cell Method to simulate the delamination of a complex-shaped structure without resolving the geometry by the mesh.

Chapter 12 performs a first step in extending the applications range of the suggested refinement method towards fluid mechanics. An essential challenge in this application regime is a convection-dominated transport. The chapter focuses on this aspect by applying the multi-level *hp*-scheme to convection-diffusion problems. To this end, the first part of the chapter briefly reviews the mathematical modeling of this application field. By means of one- and two-dimensional benchmarks, the second part analyzes the ability of the suggested refinement scheme to resolve sharp boundary layers, which are a common problem in convection-dominated scenarios. Furthermore, the compatibility of the multi-level *hp*-method with standard stabilization schemes, such as the streamline upwind Petrov-Galerkin scheme (SUPG), is studied.

In **Chapter 13**, the approximation quality of the multi-level *hp*-refinement concept is studied in the context of computational contact mechanics. From a high-order discretization point of view, the first challenge in this application field is the reduced regularity of the analytical solution at the boundary of the refinement zone. The second challenge is that—in the presence of non-linear deformations—the position of the contact zone might change in the course of the simulation. The aim of the chapter is to analyze whether the dynamic discretization capabilities of the multi-level *hp*-scheme can be used to resolve these two challenges. To this end,

the chapter first briefly reviews the essentials of non-linear solid and computational contact mechanics. In a second step, a refinement strategy is formulated to automatically refine the discretization in the vicinity of the moving contact points. On this basis, the third section of the chapter then systematically compares the approximation properties of the suggested methods by means of the Hertz benchmark and an example involving large deformations and moving contact.

This work closes with a concluding outlook in **Chapter 14**.

Part I

Formulation of multi-level hp-FEM

Chapter 2

Fundamentals of the Finite Element Method

The current chapter aims at outlining the fundamentals of the Finite Element Method (FEM). The first part of the chapter addresses the mathematical formulation of the physical phenomena in its differential and variational form. Further, the requirements for the existence and uniqueness of the solution to these variational boundary value problems are discussed. The second part of the chapter describes the essential idea of the finite element discretization and defines the respective terminology. The third part addresses the requirements for the finite element solution to converge to the analytical solution. This forms the foundation for the discussion of the different types of convergence, which follows in the next chapter.

2.1 Mathematical modeling of physical phenomena

In reality, physical phenomena are determined by the interaction of molecular, atomic or even sub-atomic particles. This *microscopic* viewpoint is well suited for modeling and understanding the fundamental principles of physical phenomena on a small scale. However, with a growing problem size, the number of particles that need to be considered quickly exceeds a reasonable range. Therefore, the mathematical description of problems on an engineering scale demands a different modeling ansatz that takes a *macroscopic* viewpoint.

A successful approach is the *continuum* concept, which is described in detail in e.g. [Holzapfel, 2000; Malvern, 1977; Marsden and Hughes, 1994]. This modeling technique is based on the assumption of a *continuous* distribution of matter throughout space. With this abstraction, the physical state of a medium can be characterized by certain *field quantities* such as density, velocity, and temperature.

2.1.1 Mathematical models in their strong and weak formulation

To describe the spatial distribution of theses quantities, the underlying physical phenomena have to be modeled mathematically. Commonly, these mathematical models are based on the assumption that the mass, the momentum, and/or the energy contained in a control volume Ω is conserved. Hence, the change in time of such a physical quantity $\boldsymbol{\Phi}$ has to be in balance with its flux $\boldsymbol{f_n}$ over the domain boundary $\Gamma = \partial\Omega$ and the production or source term $\hat{\boldsymbol{s}}$ inside

the control volume. Further, it is assumed that the flux $\boldsymbol{f_n}$ can be expressed as a function of a primary variable $\boldsymbol{u}(\boldsymbol{x})$ describing e.g. the displacement, the velocity or the temperature distribution within Ω. All four quantities are tensors of rank m.

Mathematically, this idea of *conservation* or *balance* is expressed in the following *integral* form

$$\dot{\boldsymbol{\Phi}}(t) = \int_{\Gamma} \boldsymbol{f_n}(\boldsymbol{u}) \, \mathrm{d}\Gamma + \int_{\Omega} \hat{\boldsymbol{s}} \, \mathrm{d}\Omega. \tag{2.1}$$

The second essential assumption in the concept of continuum mechanics is that there exists a unique tensor f of rank $m+1$, such that the flux over the domain boundary is given by

$$\boldsymbol{f_n} = \boldsymbol{f} \cdot \boldsymbol{n}. \tag{2.2}$$

Here, $\boldsymbol{n}$ denotes the outward pointing normal vector on the boundary Γ. This allows to re-write the conservation equation as

$$\dot{\boldsymbol{\Phi}}(t) = \int_{\Gamma} \boldsymbol{f}(\boldsymbol{u}) \cdot \boldsymbol{n} \, \mathrm{d}\Gamma + \int_{\Omega} \hat{\boldsymbol{s}} \, \mathrm{d}\Omega. \tag{2.3}$$

Unless otherwise noted, this work will focus on stationary scenarios. Accordingly, $\boldsymbol{\Phi}$ does not change over time, i.e. $\dot{\boldsymbol{\Phi}} = 0$. Therefore, the problem simplifies to

$$\int_{\Gamma} \boldsymbol{f}(\boldsymbol{u}) \cdot \boldsymbol{n} \, \mathrm{d}\Gamma + \int_{\Omega} \hat{\boldsymbol{s}} \, \mathrm{d}\Omega = 0. \tag{2.4}$$

Using the Gauss'-divergence theorem, the surface integral in the above expression can be transformed into a volume integral

$$\int_{\Gamma} \boldsymbol{f} \cdot \boldsymbol{n} \, \mathrm{d}\Gamma = \int_{\Omega} \nabla \cdot \boldsymbol{f} \, \mathrm{d}\Omega. \tag{2.5}$$

This allows to re-write (2.4) as

$$\int_{\Omega} \nabla \cdot \boldsymbol{f}(\boldsymbol{u}) \, \mathrm{d}\Omega + \int_{\Omega} \hat{\boldsymbol{s}} \, \mathrm{d}\Omega = 0 \quad \Leftrightarrow \quad \int_{\Omega} \left(\nabla \cdot \boldsymbol{f}(\boldsymbol{u}) + \hat{\boldsymbol{s}} \right) \mathrm{d}\Omega = 0. \tag{2.6}$$

Since the conservation has to be maintained for an arbitrary control volume Ω, the integrand in the above equation has to vanish for every point $\boldsymbol{x}$ in the domain Ω. Accordingly, a second possibility of expressing the mathematical model is the following *differential* form

$$\nabla \cdot \boldsymbol{f}(\boldsymbol{u}) + \hat{\boldsymbol{s}} = 0 \quad \forall \, \boldsymbol{x} \in \Omega. \tag{2.7}$$

This partial differential equation is supplemented by the following boundary conditions:

$$\boldsymbol{u} = \hat{\boldsymbol{u}} \quad \forall \, \boldsymbol{x} \in \Gamma_D \tag{2.8a}$$

$$\boldsymbol{f} \cdot \boldsymbol{n} = \hat{\boldsymbol{q}} \quad \forall \, \boldsymbol{x} \in \Gamma_N. \tag{2.8b}$$

Here, $\hat{\boldsymbol{u}}$ is the prescribed value of $\boldsymbol{u}$ on the Dirichlet part of the boundary Γ_D, and $\hat{\boldsymbol{q}}$ denotes the prescribed flux on the Neumann part of the domain boundary Γ_N, with $\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\Gamma_D \cap \Gamma_N = \emptyset$.

A third possibility of expressing the model mathematically is the *variational* form. It is obtained by multiplying (2.7) by a differentiable *test function* $\boldsymbol{v}$ that is zero on Γ_D. In a second step, the resulting equation is integrated over the domain Ω yielding

$$\int_\Omega \boldsymbol{v} \cdot (\nabla \cdot \boldsymbol{f}) \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{v} \cdot \hat{\boldsymbol{s}} \, \mathrm{d}\Omega = 0. \tag{2.9}$$

Since $\boldsymbol{v}$ is assumed to be differentiable, this expression can be further transformed by observing that

$$\nabla \cdot (\boldsymbol{v} \cdot \boldsymbol{f}) = \boldsymbol{v} \cdot (\nabla \cdot \boldsymbol{f}) + \nabla \boldsymbol{v} : \boldsymbol{f}. \tag{2.10}$$

In this way, the expression (2.9) can be written as

$$\int_\Omega \nabla \cdot (\boldsymbol{v} \cdot \boldsymbol{f}) \, \mathrm{d}\Omega - \int_\Omega \nabla \boldsymbol{v} : \boldsymbol{f} \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{v} \cdot \hat{\boldsymbol{s}} \, \mathrm{d}\Omega = 0. \tag{2.11}$$

Again using the Gauss'-divergence theorem, the left-most volume integral in the above equation can be transformed into an integral over the domain boundary

$$\int_\Omega \nabla \cdot (\boldsymbol{v} \cdot \boldsymbol{f}) \, \mathrm{d}\Omega = \int_\Gamma \boldsymbol{v} \cdot \boldsymbol{f} \cdot \boldsymbol{n} \, \mathrm{d}\Gamma \tag{2.12}$$

$$= \int_{\Gamma_D} \boldsymbol{v} \cdot \boldsymbol{f} \cdot \boldsymbol{n} \, \mathrm{d}\Gamma + \int_{\Gamma_N} \boldsymbol{v} \cdot \boldsymbol{f} \cdot \boldsymbol{n} \, \mathrm{d}\Gamma = \int_{\Gamma_N} \boldsymbol{v} \cdot \boldsymbol{f} \cdot \boldsymbol{n} \, \mathrm{d}\Gamma, \tag{2.13}$$

where the last transformation follows because the test function $\boldsymbol{v}$ was defined to be zero on Γ_D.

Following the boundary condition (2.8b), the product of the flux $\boldsymbol{f}$ and the normal vector $\boldsymbol{n}$ is prescribed on Γ_N as $\hat{\boldsymbol{q}}$. Hence, it follows

$$\int_\Omega \nabla \cdot (\boldsymbol{v} \cdot \boldsymbol{f}) \, \mathrm{d}\Omega = \int_{\Gamma_N} \boldsymbol{v} \cdot \hat{\boldsymbol{q}} \, \mathrm{d}\Gamma. \tag{2.14}$$

Inserting this result into expression (2.11) allows to give the final variational form of the mathematical model as

$$\int_\Omega \nabla \boldsymbol{v} : \boldsymbol{f} \, \mathrm{d}\Omega = \int_\Omega \boldsymbol{v} \cdot \hat{\boldsymbol{s}} \, \mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{v} \cdot \hat{\boldsymbol{q}} \, \mathrm{d}\Omega. \tag{2.15}$$

Commonly, this variational form is also denoted as the *weak* form, whereas the differential equation (2.7) is denoted as *strong* form. This terminology is derived from the continuity constraints posed on $\boldsymbol{f}$. In the case of (2.7), $\boldsymbol{f}$ has to be (strongly) differentiable in every point $\boldsymbol{x}$ of the domain Ω. This is not the case in the variational form (2.15). Here, only the product $\nabla \boldsymbol{v} : \boldsymbol{f}$ has to be integrable. This weaker requirement on the continuity gives this formulation its (second) name.

2.1.2 Weak form for heat conduction and linear elasticity

To complete the formulation of the problem, the flux $\boldsymbol{f}$ has to be expressed in terms of the primary unknown $\boldsymbol{u}$, which is chosen to describe the problem. Depending on the nature of the physical problem under consideration, $\boldsymbol{u}$ can denote the displacement, velocity, temperature, or the electric potential. This primary unknown is commonly related to the respective flux field $\boldsymbol{f}$ by constitutive models. As this work focuses on heat conduction and elasticity, the description is limited to these two cases.

Linear heat conduction

In the case of linear heat condition, the primary unknown can be identified as the temperature. Accordingly, u is a scalar field ($m = 1$) and thus the bold notation is dropped. The source term $\hat{s}$ represents a (scalar) heat source distributed in the domain Ω. Applying Stokes' heat flux theorem [Holzapfel, 2000, Page 162], the (scalar) heat flux q_n entering the control volume Ω is given by the product of the Cauchy heat flux vector $\boldsymbol{q}$ and the outward pointing normal vector $\boldsymbol{n}$, i.e.

$$q_n = -\boldsymbol{q} \cdot \boldsymbol{n}. \tag{2.16}$$

The negative sign is needed as the normal vector is outward pointing but q_n denotes the heat that enters the body [Holzapfel, 2000, Page 162]. Accordingly, the flux $\boldsymbol{f}$ in (2.15) is given by

$$\boldsymbol{f} = -\boldsymbol{q}. \tag{2.17}$$

Using Fourier's law of heat conduction, the heat flux can be expressed in terms of the temperature gradient by the thermal conductivity tensor $\boldsymbol{\kappa}$ of the respective material

$$\boldsymbol{q} = -\boldsymbol{\kappa} \cdot \nabla u. \tag{2.18}$$

Here, the negative sign is needed because the heat flows from regions with high temperatures to regions with low temperature value and thus against the direction of the gradient.

Inserting these relations in (2.15) gives the variational form modeling heat conduction

$$\int_{\Omega} \nabla v \cdot \boldsymbol{\kappa} \cdot \nabla u \, \mathrm{d}\Omega = \int_{\Omega} v \, \hat{s} \, \mathrm{d}\Omega + \int_{\Gamma_N} v \, \hat{q} \, \mathrm{d}\Omega. \tag{2.19}$$

Linear elasticity

In the case of linear elastic deformation of a continuum, the primary solution represents the displacement vector $\boldsymbol{u}$. The source of the deformation is the applied body load $\hat{\boldsymbol{b}}$ (e.g. gravitational acceleration) and the surface flux corresponds to the applied traction $\hat{\boldsymbol{t}}$. All three fields are tensors of rank one (i.e. vectors, $m = 1$). The flux $\boldsymbol{f}$ is a rank two tensor and corresponds to the Cauchy stress tensor $\boldsymbol{\sigma}$.

The model is completed using a constitutive relation expressing the stress $\boldsymbol{\sigma}$ as a function of the displacement $\boldsymbol{u}$. In the context of linear elastic deformation, a linear correlation between the stress and the (engineering) strain $\boldsymbol{\varepsilon}$ is assumed:

$$\boldsymbol{\sigma}(\boldsymbol{u}) = \mathbb{C} : \boldsymbol{\varepsilon}(\boldsymbol{u}) \quad \text{with } \boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\top}\right), \tag{2.20}$$

and $\mathbb{C}$ denoting the elastic material tensor.

The use of this relation in the weak form (2.15) requires a slight reformulation of the left-hand side integrand. To this end, one exploits the constraint that the stress tensor has to be symmetric to conserve the angular momentum [Holzapfel, 2000, Page 147], i.e. $\boldsymbol{\sigma} = \boldsymbol{\sigma}^\top$. This allows the left-hand side integrand in (2.15) to be written as

$$\boldsymbol{\nabla v} : \boldsymbol{\sigma} = \boldsymbol{\nabla v}^\top : \boldsymbol{\sigma}^\top = \boldsymbol{\nabla v}^\top : \boldsymbol{\sigma}. \tag{2.21}$$

With this equality, the integrand can be reformulated in terms of the strain tensor as follows

$$\boldsymbol{\nabla v} : \boldsymbol{\sigma} = \frac{1}{2}\left(\boldsymbol{\nabla v} : \boldsymbol{\sigma} + \boldsymbol{\nabla v}^\top : \boldsymbol{\sigma}\right) = \boldsymbol{\varepsilon}(\boldsymbol{v}) : \boldsymbol{\sigma} = \boldsymbol{\varepsilon}(\boldsymbol{v}) : \mathbb{C} : \boldsymbol{\varepsilon}(\boldsymbol{u}). \tag{2.22}$$

Using this result in (2.15), the final model reads

$$\int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{v}) : \mathbb{C} : \boldsymbol{\varepsilon}(\boldsymbol{u})\,\mathrm{d}\Omega = \int_\Omega \boldsymbol{v}\cdot\hat{\boldsymbol{b}}\,\mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{v}\cdot\hat{\boldsymbol{t}}\,\mathrm{d}\Omega. \tag{2.23}$$

Summary

In order for the variational forms to be evaluable, all integrals in (2.19) and (2.23) have to be finite. For the integrals on the right hand side, this can be ensured if $\boldsymbol{v}$, $\hat{s}$, $\hat{\boldsymbol{b}}$, $\hat{q}$, and $\hat{\boldsymbol{t}}$ are L_2-integrable. The left hand sides of the weak forms (2.19) and (2.23) involve the first derivatives of the solution $\boldsymbol{u}$ and the test function $\boldsymbol{v}$ and the material constants $\boldsymbol{k}$ and $\mathbb{C}$. Assuming that $\boldsymbol{\kappa}$ and $\mathbb{C}$ are finite, these integrals are finite if the derivatives of $\boldsymbol{u}$ and $\boldsymbol{v}$ are L_2-integrable in Ω. Accordingly, $\boldsymbol{u}$ and $\boldsymbol{v}$ have to be found in the Sobolev space of first order $H^1(\Omega)$ that complies to the Dirichlet boundary conditions

$$H^1_{\hat{u}}(\Omega) = \left\{v_i \in H^1(\Omega) : v_i = \hat{u}_i \quad \forall\, \boldsymbol{x} \in \Gamma_D\right\} \tag{2.24a}$$
$$H^1_0(\Omega) = \left\{v_i \in H^1(\Omega) : v_i = 0 \quad \forall\, \boldsymbol{x} \in \Gamma_D\right\}. \tag{2.24b}$$

The final mathematical model can then be formulated as follows

$$\text{Find } \boldsymbol{u} \in H^1_{\hat{u}}(\Omega) \text{ such that } a(\boldsymbol{v},\boldsymbol{u}) = f(\boldsymbol{v}) \quad \forall\, \boldsymbol{v} \in H^1_0(\Omega). \tag{2.25}$$

Here, $f(\boldsymbol{v})$ and $a(\boldsymbol{v},\boldsymbol{u})$ are a linear- and a bi-linear form. Depending on the problem under consideration, these read

$$a(v,u) = \int_\Omega \nabla v \cdot \boldsymbol{\kappa} \cdot \nabla u\,\mathrm{d}\Omega \qquad \text{and} \quad f(v) = \int_\Omega v\,\hat{s}\,\mathrm{d}\Omega + \int_{\Gamma_N} v\,\hat{q}\,\mathrm{d}\Omega \tag{2.26a}$$
$$a(\boldsymbol{v},\boldsymbol{u}) = \int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{v}) : \mathbb{C} : \boldsymbol{\varepsilon}(\boldsymbol{u})\,\mathrm{d}\Omega \quad \text{and} \quad f(\boldsymbol{v}) = \int_\Omega \boldsymbol{v}\cdot\hat{\boldsymbol{b}}\,\mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{v}\cdot\hat{\boldsymbol{t}}\,\mathrm{d}\Omega. \tag{2.26b}$$

2.1.3 Existence and uniqueness of the weak solution

Having derived the mathematical problem formulations in the previous section, the next essential question is whether a solution *exists* and whether this solution is *unique*. To answer this question, the operands of the variational problem are classified in terms of the following characteristics.

Definition 2.1.1: Continuous operator [Reddy, 1997, Page 142]
Let f be an operator $f : \mathcal{V} \to \mathcal{W}$ where $\mathcal{V}$ and $\mathcal{W}$ are normed spaces. Then f is continuous at *$v_0 \in \mathcal{V}$ if for $\epsilon > 0$ there is a positive number δ such that*

$$||f(v_0) - f(u)||_{\mathcal{W}} < \epsilon \quad \text{whenever } ||v_0 - u||_V < \delta. \tag{2.27}$$

If (2.27) holds for every $v \in \mathcal{V}$, f is denoted as continuous.

Definition 2.1.2: Bounded operator [Reddy, 1997, Page 146]
Let f be a linear operator $f : \mathcal{V} \to \mathcal{W}$ where $\mathcal{V}$ and $\mathcal{W}$ are normed spaces. Then f is bounded *if there exists a number $K > 0$ such that:*

$$||f(v)||_{\mathcal{W}} \leq K\,||v||_{\mathcal{V}} \quad \forall\, v \in \mathcal{V}. \tag{2.28}$$

The least upper bound taken over all members of $\mathcal{V}$ defines the norm $||f||$

$$||f|| = \sup_{v \in \mathcal{V}, v \neq 0} \frac{||f(v)||_{\mathcal{W}}}{||v||_{\mathcal{V}}}. \tag{2.29}$$

It follows that

$$||f(v)||_{\mathcal{W}} \leq ||f||\,||v||_{\mathcal{V}}. \tag{2.30}$$

The continuity and the boundedness of linear operators are closely connected, as expressed in the following lemma taken from e.g. [Reddy, 1997]

Lemma 2.1.1:
A linear operator $f : \mathcal{V} \to \mathcal{W}$ from a normed space $\mathcal{V}$ to a normed space $\mathcal{W}$ is continuous if and only if it is bounded.

In a similar way, the continuity of a bi-linear form can be expressed by considering the boundedness with respect to both its arguments

Definition 2.1.3: Continuous bi-linear form [Brenner and Scott, 2008; Reddy, 1997]
Let $a(\cdot,\cdot)$ be a bi-linear form $a : \mathcal{U} \times \mathcal{V} \to \mathbb{R}$ where $\mathcal{U}$ and $\mathcal{V}$ are normed linear spaces. Then $a(\cdot,\cdot)$ is called a continuous *or* bounded *bi-linear form if there exists a positive number M such that:*

$$|a(u,v)| \leq M\,||u||_{\mathcal{U}}\,||v||_{\mathcal{V}} \quad \forall\, u \in \mathcal{U},\, v \in \mathcal{V}. \tag{2.31}$$

Definition 2.1.4: Coercive/V-elliptic bi-linear form [Reddy, 1997, Page 165]
Let $a(\cdot,\cdot)$ be a bi-linear form $a : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ where $\mathcal{V}$ is an inner product space. Then $a(\cdot,\cdot)$ is called a coercive *or* V-elliptic *bi-linear form if there exists a number $\alpha > 0$ such that:*

$$a(v,v) \geq \alpha\,||v||_{\mathcal{V}}^2 \quad \forall\, v \in \mathcal{V}. \tag{2.32}$$

If the left and right hand side of the weak form meet these requirements, the existence and the uniqueness of the solution u is ensured by the following lemma

Lemma 2.1.2: Lax-Milgram Lemma [Ciarlet, 2002, Theorem 1.1.3]
Let $\mathcal{V}$ be a Hilbert space, and let $a : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ be a continuous, V-elliptic bi-linear form, and let $f : \mathcal{V} \to \mathbb{R}$ be a continuous linear form. Then the abstract variational problem:

$$\text{Find } u \in \mathcal{V} \text{ such that } a(v,u) = f(v) \quad \forall\, v \in \mathcal{V}. \tag{2.33}$$

has one and only one solution.

The properties of Hilbert spaces are reviewed in e.g. [Brenner and Scott, 2008, Section 2.2], and a detailed proof of the above lemma can be found in e.g. [Ciarlet, 2002, Pages 8–9], [Reddy, 1997, Pages 166–169] or [Brenner and Scott, 2008, Pages 60–63]. Therefore, these details are not stated here. It is more important to note that the requirements on continuity and coercivity of the Lax-Milgram Lemma are met by the forms given in (2.26) assuming that the constitutive terms $\boldsymbol{\kappa}$ and $\mathbb{C}$ are bounded away from zero and infinity. Therefore, a unique solution can be found. Moreover, since $a(\cdot,\cdot)$ is coercive and $f(\cdot)$ is bounded it follows that

$$\alpha \, ||u||_{\mathcal{V}}^2 \leq a(u,u) = f(u) \leq ||f|| \, ||u||_{\mathcal{V}} \tag{2.34}$$

$$\Rightarrow ||u||_{\mathcal{V}} \leq \frac{1}{\alpha} ||f||. \tag{2.35}$$

Hence, small changes in the input data $f(\cdot)$ lead to small changes on the solution u. Therefore, the solution continuously depends on $f(\cdot)$, which renders the continuous problem well posed [Ciarlet, 2002, Remark 1.1.3].

In Section 2.3, it is shown that these properties also carry over to the discrete finite element problem, and that—under some further assumptions—the numerical approximation convergences to the unique analytical solution. Before addressing these aspects, the following section outlines the essential ideas of the Finite Element Method.

2.2 Finite element discretization

Despite the rather simple structure of the above equations, a closed-form solution can only be derived for extremely simplified settings. Therefore, large efforts have been invested in the last decades to find approximations to the analytical solution by utilizing the growing computational power. To this end, different numerical discretization methods have been introduced, which are tailored for the respective problem under consideration. In the context of mechanical, structural and civil engineering, the Finite Element Method (FEM) emerged to a prevalent approach. The basic idea is to discretize the problem in its variational formulation using the Galerkin method in a way that maximizes the geometric flexibility.

The following description addresses the core aspects of this discretization strategy. The first part of this section outlines the essential ideas and properties of the Galerkin method. The second part formalizes the construction of the finite element spaces.

2.2.1 Galerkin method

The essential idea of the Galerkin method is to find an approximation to the analytical solution of the variational problem

$$\text{Find } \tilde{u} \in \mathcal{S} \text{ such that } a(v, \tilde{u}) = f(v) \quad \forall\, v \in \mathcal{V}. \tag{2.36}$$

As discussed during the derivation of the weak formulation, the function spaces $\mathcal{S}$ and $\mathcal{V}$ have to be *admissible*, in the sense that functions in the space $\mathcal{S}$ comply to the applied Dirichlet boundary condition $\hat{u}$ on Γ_D, and the functions in $\mathcal{V}$ are zero on Γ_D. Accordingly, the two spaces $\mathcal{S}$ and $\mathcal{V}$ cannot be equal if $\hat{u} \neq 0$. This un-symmetry is resolved by "shifting" the solution function $\tilde{u}$ by a suitable function $g \in \mathcal{S}$ such that $\tilde{u} = g + u$ with $u \in \mathcal{V}$ [Szabó and Babuška, 1991, Page 17]. This allows to write the above weak form as

$$\text{Find } u \in \mathcal{V} \text{ such that } a(v, u) = \tilde{f}(v) \quad \forall\, v \in \mathcal{V}, \tag{2.37a}$$

$$\text{with} \quad \tilde{f}(v) = f(v) - a(v, g). \tag{2.37b}$$

If $\hat{u} \equiv 0$, g is chosen as the zero-function such that $\tilde{u} = u$.

To find an approximation to the analytical solution u of the above problem, the Galerkin method defines a similar problem in a finite-dimensional sub-space $\mathcal{V}_h \subset \mathcal{V}$. This discrete, finite-dimensional problem is defined as

$$\text{Find } u_h \in \mathcal{V}_h \text{ such that } a(v_h, u_h) = \tilde{f}(v) \quad \forall\, v_h \in \mathcal{V}_h. \tag{2.38}$$

Given that the sub-space $\mathcal{V}_h$ and the forms $a(\cdot, \cdot)$ and $f(\cdot)$ satisfy the conditions of the Lax-Milgram Lemma 2.1.2, the existence and the uniqueness of the numerical approximation can be ensured [Ciarlet, 2002, Page 38]. Hence, also the discrete problem is well posed.

Since the sub-space $\mathcal{V}_h$ is finite-dimensional, it can be spanned by a finite number of basis functions: $\boldsymbol{N} = \{N_1, N_2, \ldots, N_n\}$, with $n = \dim(\mathcal{V}_h)$. Accordingly, every element of the space can be represented as a linear combination of the basis functions

$$\forall\, v \in \mathcal{V}_h, \quad v = \sum_{i=1}^{n} N_i \hat{v}_i = \boldsymbol{N}\hat{\boldsymbol{v}}, \tag{2.39}$$

with $\hat{v}_i$ being the respective coefficients of the linear combination. These coefficients are typically denoted as *degrees of freedom* (DOFs). Using this formalism and the Einstein summation convention, equation (2.38) can be transformed into the following discrete problem

$$\text{Find } \hat{\boldsymbol{u}} \in \mathbb{R}^n \text{ such that } a(N_i\hat{v}_i, N_j\hat{u}_j) = f(N_i\hat{v}_i) \quad \forall\, \hat{\boldsymbol{v}} \in \mathbb{R}^n. \tag{2.40}$$

Due to the linearity of $a(\cdot, \cdot)$ and $f(\cdot)$, the coefficient vectors can be taken out of the two forms

$$\hat{v}_i a(N_i, N_j)\hat{u}_j = \hat{v}_i f(N_i) \quad \Leftrightarrow \quad \hat{v}_i\big(a(N_i, N_j)\hat{u}_j - f(N_i)\big) = 0 \tag{2.41}$$

As the above expression has to be fulfilled for any $\hat{\boldsymbol{v}} \in \mathbb{R}^n$, the term inside of the bracket has to be zero

$$a(N_i, N_j)\hat{u}_j - f(N_i) = 0. \tag{2.42}$$

Hence, the solution coefficients $\hat{\boldsymbol{u}}$ to the discrete problem can be found by solving the following system of linear equations:

$$\boldsymbol{A}\hat{\boldsymbol{u}} = \boldsymbol{F} \quad \text{with} \quad A_{ij} = a(N_i, N_j) \quad \text{and} \quad F_i = f(N_i). \tag{2.43}$$

In the context of computational mechanics, the matrix $\boldsymbol{A}$ and the right-hand side vector $\boldsymbol{F}$ are typically denoted as *stiffness matrix* and *force vector*, respectively. For the matrix $\boldsymbol{A}$, different advantageous properties can be observed. First, $\boldsymbol{A}$ is invertible due to the (discrete) Lax-Milgram Lemma. Second, $\boldsymbol{A}$ is symmetric if $a(\cdot,\cdot)$ is symmetric. And third, $\boldsymbol{A}$ is positive definite if $a(\cdot,\cdot)$ is coercive.

It has to be noted that—by construction—the numerical solution u_h is zero on the Dirichlet boundary. To construct the approximation to the in-homogeneous solution $\tilde{u}$, the shifting by g has to be reversed. Accordingly, the final numerical approximation is given by $\tilde{u}_h = u_h + g$. The space containing $\tilde{u}_h$ is denoted as

$$\mathcal{S}_h = \{\tilde{v}_h \in \mathcal{S} : \exists v_h \in \mathcal{V}_h \text{ such that } \tilde{v}_h = v_h + g\}. \tag{2.44}$$

Clearly, the error of the approximations are equivalent as

$$e = u - u_h = (\tilde{u} - g) - (\tilde{u}_h - g) = \tilde{u} - \tilde{u}_h. \tag{2.45}$$

To analyze the properties of the error, the two forms (2.38) and (2.37a) are subtracted in $\mathcal{V}_h$ giving

$$a(v_h, u) - a(v_h, u_h) = a(v_h, u - u_h) = a(v_h, e) = 0 \quad \forall\, v_h \in \mathcal{V}_h. \tag{2.46}$$

Therefore, the approximation error e is *a-orthogonal* to any element v_h of the space $\mathcal{V}_h$. This key to the success of the Galerkin method is denoted as *Galerkin orthogonality*.

This result has a geometric interpretation when $a(\cdot,\cdot)$ is symmetric. In this case, $a(\cdot,\cdot)$ represents an inner product $(\cdot,\cdot)_a$, which allows the identification of the approximation u_h as the a-orthogonal projection of the analytical solution u onto $\mathcal{V}_h$ (see e.g. [Ciarlet, 2002, Remark 2.4.1] or [Reddy, 1997, Section 10.2]). Moreover, the symmetry of $a(\cdot,\cdot)$ implies the Cauchy-Schwarz inequality (see e.g. [Brenner and Scott, 2008, Remark 2.1.8])

$$a(v, w)^2 \leq a(v, v)a(w, w) \quad \forall\, v,\, w \in \mathcal{V}, \tag{2.47}$$

and the so-called *energy norm* can be defined [Brenner and Scott, 2008, Proposition 2.1.9]

$$||v||_E = \sqrt{a(v, v)} \quad \forall\, v \in \mathcal{V}. \tag{2.48}$$

Using the Galerkin-orthogonality and the symmetry of the bi-linear form, the error in the energy norm can be computed as [Szabó et al., 2004]

$$\begin{aligned}
||e||_E^2 &= ||\tilde{u} - \tilde{u}_h||_E^2 = ||u - u_h||_E^2 && \text{(2.49a)}\\
&= a(u - u_h, u - u_h) = a(u, u) - 2a(u, u_h) + a(u_h, u_h) && \text{(2.49b)}\\
&= a(u, u) - 2a(u - u_h + u_h, u_h) + a(u_h, u_h) && \text{(2.49c)}\\
&= a(u, u) - \underbrace{2a(u - u_h, u_h)}_{=0,\ \text{from (2.46) since } u_h \in \mathcal{V}_h} - 2a(u_h, u_h) + a(u_h, u_h) && \text{(2.49d)}\\
&= a(u, u) - a(u_h, u_h) = ||u||_E^2 - ||u_h||_E^2. && \text{(2.49e)}
\end{aligned}$$

Since $||e||_E^2 \geq 0$, the above result implies that

$$||u_h||_E^2 \leq ||u||_E^2. \tag{2.50}$$

Accordingly, the homogeneous approximation u_h underestimates the strain energy of the homogeneous solution u. It has to be noted that this result does **not** hold for the pair of in-homogeneous solutions $\tilde{u}$ and $\tilde{u}_h$ (if $g \not\equiv 0$): as $\tilde{u}_h \notin \mathcal{V}_h$, the Galerkin orthogonality cannot be applied to the second term in (2.49d). Therefore, the error cannot be computed as the difference of the norms of the two in-homogeneous solutions

$$||e||_E^2 = ||\tilde{u} - \tilde{u}_h||_E^2 = ||u - u_h||_E^2 = ||u||_E^2 - ||u_h||_E^2 \neq ||\tilde{u}||_E^2 - ||\tilde{u}_h||_E^2. \tag{2.51}$$

Further exploiting the symmetry of the problem, the following fundamental bound for the approximation error in the energy norm can be derived *cf.* [Hughes, 2000, Page 186]. To this end, a function $w_h \in \mathcal{V}_h$ is selected, and by means of the symmetry and the bi-linearity of $a(\cdot, \cdot)$ it follows that

$$a(e + w_h, e + w_h) = a(e, e) + \underbrace{2a(w_h, e)}_{=0, \text{ from (2.46) since } w_h \in \mathcal{V}_h} + a(w_h, w_h) \tag{2.52}$$

Since $a(\cdot, \cdot)$ is assumed to be coercive, $a(w_h, w_h) \geq 0$. This implies that

$$a(e, e) \leq a(e + w_h, e + w_h) \quad \forall w_h \in \mathcal{V}_h. \tag{2.53}$$

Let $v_h = u_h - w_h$ and note that $e + w_h = u - u_h + w_h = u - v_h$. Hence, it follows that

$$||e||_E^2 = a(e, e) = a(u - u_h, u - u_h) \leq a(u - v_h, u - v_h) \quad \forall\, v_h \in \mathcal{V}_h. \tag{2.54}$$

In other words, the error of the Galerkin solution u_h is smaller than the error of any other solution candidate v_h. Accordingly, u_h is the *best approximation* to u possible in $\mathcal{V}_h$ measured in the energy norm.

Equivalently, one can choose $\tilde{v}_h = \tilde{u}_h - w_h$ and observe that $e + w_h = \tilde{u} - \tilde{u}_h + w_h = \tilde{u} - \tilde{v}_h$ to yield

$$||e||_E^2 = a(e, e) = a(\tilde{u} - \tilde{u}_h, \tilde{u} - \tilde{u}_h) \leq a(\tilde{u} - \tilde{v}_h, \tilde{u} - \tilde{v}_h) \quad \forall\, \tilde{v}_h \in \mathcal{S}_h. \tag{2.55}$$

Hence, the best approximation property also applies to the in-homogeneous solution pair.

Having outlined these corner stones of the Galerkin method, the following section addresses the construction of the sub-space $\mathcal{V}_h$ by the Finite Element Method.

2.2.2 Finite element spaces

As expressed in equation (2.38), the key idea of the Galerkin approach is to find an approximation to the analytical solution by restricting the continuous problem (2.37a) to a finite-dimensional sub-space $\mathcal{V}_h \subset \mathcal{V}$. Accordingly, the essential aspect of any Galerkin-based discretization method is the systematic definition of the basis functions $\{N_1, N_2, \ldots, N_n\}$ that span the space $\mathcal{V}_h$. Thereby, one main aim is to maximize the geometric flexibility of the discretization. At the same time, the resulting stiffness matrix $\boldsymbol{A}$ should be kept as sparse

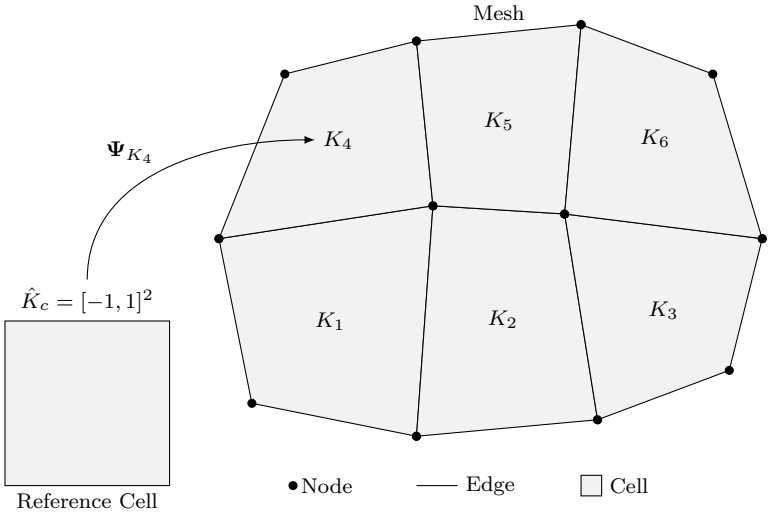


Figure 2.1: Entities of a finite element mesh.

and well-conditioned as possible. To meet these demands, the basis idea of the Finite Element Method is to partition the domain Ω into a finite number of elements. On each of these elements, standard shape functions are defined. By assembling the shape functions of adjacent elements, the basis functions N_i can be constructed in a systematic and easy-to-implement way. This element-wise definition of the basis functions allows for a high complexity of the geometric model under consideration, which renders the finite element approach as the method of choice in today's engineering practice. Following [Braess, 2007; Ciarlet, 2002; Demkowicz, 2007; Schwab, 1998; Šolín, 2004], the essential ideas of this procedure and the respective terminology are outlined in the following sub-sections.

2.2.2.1 The finite element mesh

The first step in constructing the finite element space $\mathcal{V}_h$ is the discretization of the domain Ω. This process is denoted as mesh generation or meshing. Although the concept of a mesh is very intuitive, the mathematical formulation and the generic implementation is highly complex and demands for a detailed description of solid modeling. For this reason, this section only outlines the fundamental concepts. A mathematically more rigorous presentation of the this matter is given in e.g. [Mäntylä, 1988; Shewchuk, 1999; Thompson et al., 1998].

The essential idea is to subdivide the domain Ω into a finite number of cells that are defined in the following way

Definition 2.2.1: Cell K [Ciarlet, 2002, Page 38]
A cell *K is a closed subset of Ω with a Lipschitz-continuous boundary and a non-empty interior denoted by $\overset{\circ}{K}$.*

The most common cell types for the finite elements are triangles, quadrilaterals, tetrahedrons, prisms, and hexahedrons [Šolín, 2004; Zienkiewicz et al., 2005a]. This work only considers quadrilateral and hexahedral cells.

The corner points of these cells are denoted as *nodes*, and the connection between two nodes

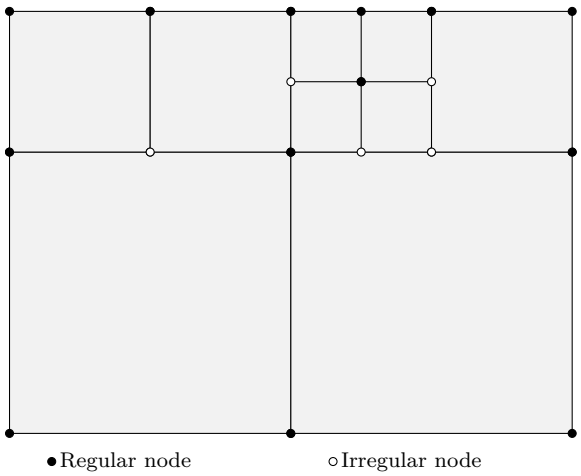


Figure 2.2: Example of an irregular mesh with hanging nodes.

are called *edges*. The regions enclosed by a closed wire of edges are denoted as *faces*.

To allow for a systematic implementation of the Finite Element Method, it is assumed that the shape of each cell K can be described by a *standard* or *reference cell* $\hat{K}$ that is mapped onto K by a bijective *geometric mapping* $\mathbf{\Psi} : \hat{K} \to K$. This idea is illustrated in Figure 2.1.

The collections of cells is denoted as

Definition 2.2.2: Mesh $\mathcal{T}_h$ [Ciarlet, 2002, Page 38]
A mesh $\mathcal{T}_h$ *is a partition of* Ω *into a finite number of disjoint cells* K_i *such that:*

a) The original domain is recovered by the cells, i.e. $\bar{\Omega} = \bigcup_{K_i \in \mathcal{T}_h} K_i$.

b) The interior of two distinct cells do not overlap, i.e. $\overset{\circ}{K}_i \cap \overset{\circ}{K}_j = \emptyset \quad \forall\, K_i, K_j \in \mathcal{T}_h,\, i \neq j$.

To establish an adjacency relation between the cells of the mesh, the cells are grouped into patches in the following way:

Definition 2.2.3: Patch
Let v *be a node, edge, or face of the mesh* $\mathcal{T}_h$. *Then the set of all cells* $K \in \mathcal{T}_h$ *that are adjacent to* v *form the* patch P_v, *that is*

$$P_v = \{K \in \mathcal{T}_h : v \subseteq K\}. \tag{2.56}$$

In the context of locally changing cell sizes, a distinction between a regular and an irregular discretization has to be made. As stated by Felippa [2013], the essential question is whether a cell "gets along with its neighbors". An example where this is not the case is depicted in Figure 2.2. The following definitions render this aspect more formally.

Definition 2.2.4: Regular and irregular/hanging nodes, edges, and faces [Demkowicz et al., 1989, Page 83]
A node is called regular *if it constitutes a node for each of the adjacent cell. Otherwise the node is denoted as* irregular *or* hanging. *The analog applies to edges and faces.*

Definition 2.2.5: Regular/irregular mesh [Schwab, 1998, Definition 4.5]
A mesh $\mathcal{T}_h$ is denoted as regular *if it contains no hanging nodes, edges, or faces. That is: the intersection of two cells $K_i \cap K_j, i \neq j$, is either empty or a node or an* entire *edge or face of K_i and K_j. If a mesh is not regular, it is denoted as* irregular.

Definition 2.2.6: Index of irregularity [Demkowicz et al., 1989, Page 83]
The index of irregularity *is the maximum number of irregular nodes per cell side.*

For example, the irregularity index of the mesh depicted in Figure 2.2 is two.

2.2.2.2 Elements and shape functions

As mentioned in the introduction of this section, the essential idea of the Finite Element Method is to define the basis functions of the space $\mathcal{V}_h$ in an element-wise manner. For this purpose, a set of functions is associated to every cell of the mesh. To allow for a systematic implementation of the method, these functions are defined on the corresponding reference cell $\hat{K}$.

Definition 2.2.7: Standard element shape functions
The standard element shape functions $\hat{\boldsymbol{\xi}}^{\hat{K}} = \{\hat{\xi}_1^{\hat{K}}, \hat{\xi}_2^{\hat{K}}, \ldots, \hat{\xi}_{n_P}^{\hat{K}}\}$ *are a set of linear independent polynomials of order p_k defined on $\hat{K}$.*

By combining these standard functions with the geometric mapping of the respective cell, the element shape functions can be defined on the actual cell geometry as follows:

Definition 2.2.8: Element shape functions
The element shape functions $\boldsymbol{\xi}^K = \left\{\xi_1^K, \xi_2^K, \ldots, \xi_{n_P}^K\right\}$ *are a set of linear independent functions defined on K with $\xi_i^K = \hat{\xi}_i^{\hat{K}} \circ \boldsymbol{\Psi}_K^{-1}$. The* space of element shape functions $\mathcal{P}$ *is defined by the span of these basis functions, that is*

$$\mathcal{P} = span\left\{\xi_1^K, \xi_2^K, \ldots, \xi_{n_P}^K\right\}. \tag{2.57}$$

Using these shape functions, the finite element can be defined formally as follows:

Definition 2.2.9: Finite element [Ciarlet, 2002, Page 78]
A finite element *in $\mathbb{R}^n$ is a pair $(K, \mathcal{P})$ where:*

- *K is a cell as in Definition 2.2.1.*
- *$\mathcal{P}$ is the element space according to Definition 2.2.8.*

To simplify the notation, the elements of the mesh are identified by their cell K.

2.2.2.3 The finite element space and its assembly

Using the aforementioned definition of the mesh $\mathcal{T}_h$ and the individual elements, the finite element space can be defined in an element-wise manner:

Definition 2.2.10: Finite element space $\mathcal{V}_h(\mathcal{T}_h)$
On a mesh $\mathcal{T}_h$, the finite element space *is given by*

$$\mathcal{V}_h(\mathcal{T}_h) = \{v \in \mathcal{V} : v|_K \in \mathcal{P}(K), \quad \forall K \in \mathcal{T}_h\}. \tag{2.58}$$

As shown in (2.39), the essential step for constructing the system of linear equations is the selection of a suitable set of basis functions $\{N_i\}$ that spans the space $\mathcal{V}_h$. Here, the support of these basis functions should be as small as possible to optimize the numerical properties of the method. To this end, the classical Finite Element Method associates each basis function N_i to a node, edge, or face v of the mesh $\mathcal{V}_h$ and defines its support as the patch P_v corresponding to v. Hence, the respective shape function is non-zero only on the cells adjacent to the associated node, edge or face and zero on all other cells of the mesh.

The starting point for this systematic construction of the basis functions N_i is the observation that—by Definition 2.2.10—any function $v \in \mathcal{V}_h$ can be represented as a linear combination of element shape functions. Therefore, for every $K \in \mathcal{T}_h$ there exists a set of scalar values $\{a_1, a_2, \ldots, a_{n_p^K}\}$ such that

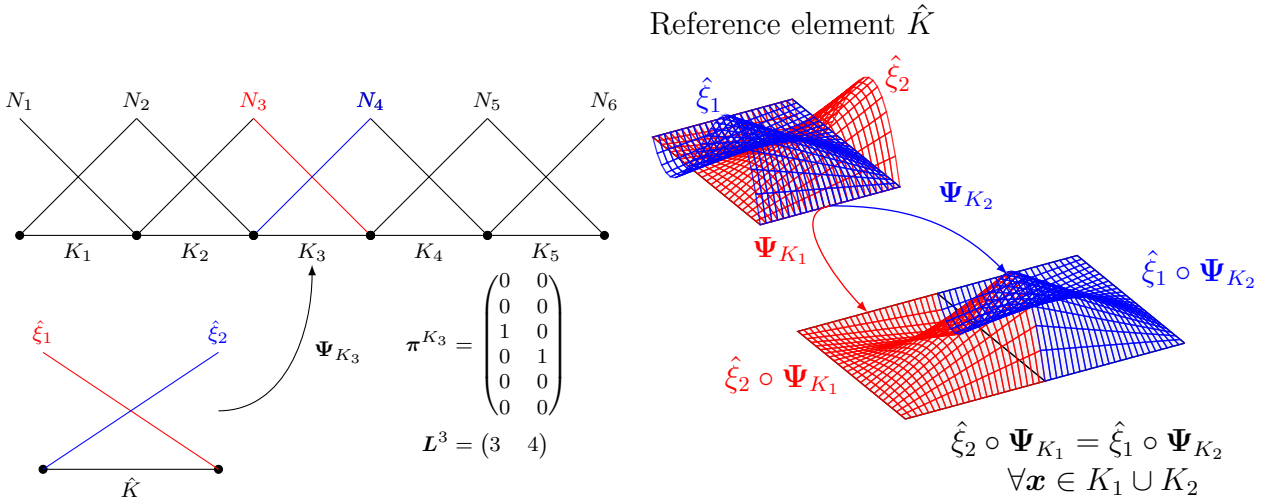
$$v|_K = \sum_{j=1}^{n_p^K} a_j \xi_j^K. \tag{2.59}$$

It is therefore natural to also define the basis functions of the space $\mathcal{V}_h$ as a linear combination of the element shape functions. Thus, a basis function N_i is given by

$$N_i|_K = \sum_j \pi_{ij}^K \xi_j^K \quad \forall\, K \in \mathcal{T}_h. \tag{2.60}$$

In this way, the basis functions can be fully defined by specifying the factors π_{ij}^K. A one-dimensional example of this approach is illustrated in Figure 2.3a. Following [Ainsworth and Oden, 1997], this matrix $\boldsymbol{\pi}^K$ is denoted as the *connectivity matrix* or *connectivity mapping* of the element K.

However, the values π_{ij}^K cannot be chosen arbitrarily as the basis functions N_i have to meet the continuity requirements of the space $\mathcal{V}$. For example, if $\mathcal{V} = H^1$, the basis functions have



(a) Illustration of the basis function definition using the connectivity matrix $\boldsymbol{\pi}$

(b) Pairwise definition of element shape functions for simple construction of C^0-continuous basis functions

Figure 2.3: Illustration of the basis function definition using the connectivity matrix $\boldsymbol{\pi}$.

to be in $H^1(K)$ for all $K \in \mathcal{T}_h$ and C^0-continuous in-between elements (see e.g. [Ciarlet, 2002, Theorem 2.1.1] or [Šolín, 2004, Lemma 1.1]). Since the element shape functions $\boldsymbol{\xi}^K$ are defined as polynomials, they naturally comply to the first requirement. However, the inter-element continuity of the basis functions has to be respected when defining the values π_{ij}.

In the framework of the standard Finite Element Method, these constraints are met by a careful selection of the element shape functions ξ_j. These are designed such that a continuous basis function N_i can be constructed by using only *one* shape function ξ_j^K per element K, that is

$$\forall K \in P_v, \text{there exists one and only one } \xi_j^K \in \boldsymbol{\xi}^K \text{ such that } N_i|_K = \xi_j^K. \tag{2.61}$$

Here, P_v denotes the patch associated to N_i. To meet the C^0-inter-element continuity constraints, the element shape functions are defined in pairs that have the same trace on opposing sides of the element (see Figure 2.3b). The construction of C^1-continuous basis functions is considerably more challenging. An overview about such element formulations is given in e.g. [Braess, 2007, Chapter 2.5].

The one-to-one correspondence between the global basis functions N_i and the element shape function ξ_j simplifies the structure of the matrix $\boldsymbol{\pi}$. As illustrated in Figure 2.3a, $\boldsymbol{\pi}$ contains only the values one and zero and only one non-zero entry per row and column [Oden, 1969].

This Boolean nature of the connectivity allows to transform the matrix into an equivalent, one-dimensional mapping array [Bathe, 2007; Hughes, 2000; Karniadakis and Sherwin, 2013]. This so-called location map $\boldsymbol{L}^K$ directly associates the index of the local shape function ξ_j^K of an element K to the index of the corresponding global basis function, that is

$$N_{L_j^K}|_K = \xi_j^K \tag{2.62}$$

With the help of the location map, the final stiffness matrix $\boldsymbol{A}$ in (2.43) can be computed in an element-wise procedure. To this end, an element stiffness matrix $\boldsymbol{A}^K$ is computed for every element K as

$$A_{ij}^K = a(\xi_i^K, \xi_j^K)_K. \tag{2.63}$$

Here, $a(\xi_i^K, \xi_j^K)_K$ represents the evaluation of the bi-linear form on the element K. These element contributions are then *assembled* into the final stiffness matrix $\boldsymbol{A}$ by summing the respective entries of each element matrix $\boldsymbol{A}^K$ as illustrated in Algorithm 1.

This standard assembly routine is at the heart of most FEM-implementations (see e.g. [Bathe, 2007; Hughes, 2000; Karniadakis and Sherwin, 2013]). However, it has to be noted that this algorithm is based on the one-to-one correspondence between global basis functions N_i and element shape function ξ_j^K expressed in (2.61). This direct relation is given in the case of a regular discretization, but it cannot be assumed when the mesh contains hanging nodes. Accordingly, the use of an irregular discretization demands for a generalized assembly procedure. For this reason, this aspect is re-considered when discussing classical *hp*-approaches in Chapter 4.

Algorithm 1 Standard assembly routine of the Finite Element Method

```
1  // first, initialize global stiffness matrix with zero ...
2  A = 0
3  // Then, run over each element of the mesh ...
4  for each element K ∈ T_h
5  {
6    // ... and each row ...
7    for each ξ_i ∈ ξ^K of element K
8    {
9      // ... and each column of the element matrix A^K ...
10     for each ξ_j ∈ ξ^K of element K
11     {
12       // ... and add the element stiffness entry to the
13       // respective global stiffness entry
14       A[L^K[i]][L^K[j]] += A^K[i][j]
15     }
16   }
17 }
```

2.3 Convergence requirements of the Finite Element Method

As discussed in the previous section, applying the Finite Element Method to a symmetric variational boundary value problem provides the best approximation to the analytical solution $v \in \mathcal{V}$ contained in the sub-space $\mathcal{V}_h \subset \mathcal{V}$. Therefore, an important question is *if* and *how* the numerical approximation approaches the exact solution as the FE-space is increased. The first aspect of this question is addressed in this section by outlining the essential requirements for *convergence*. The different types of convergence are discussed in Chapter 3.

To render this important aspect more formal, a *family of sub-spaces* $\{\mathcal{V}_h\}$ is considered. Here, h denotes the defining parameter of the family and has limit zero [Ciarlet, 2002, Page 104]. For every sub-space, the discrete problem (2.38) can be solved, giving rise to a *family* of discrete solutions $\{u_h\}$. This allows the following formal definition of convergence:

Definition 2.3.1: Convergence [Ciarlet, 2002, Page 104] or [Reddy, 1997, Section 10.2]
A family of discrete solutions $\{u_h\}$ *is said to be* convergent *to the exact solution* u *if*

$$\lim_{h \to 0} ||u - u_h||_{\mathcal{V}} = 0. \tag{2.64}$$

It is essential to note that the above definition of convergence is made with respect to the $\mathcal{V}$-norm rather than the energy norm. Therefore, a matching equivalent to the aforementioned best approximation property has to be formulated. This can be done by observing that the following similar error estimate in the $\mathcal{V}$-norm can be derived when assuming that the bi-linear

form is coercive and continuous [Reddy, 1997, Section 10.2]:

$$\begin{aligned}
\forall\, v_h \in \mathcal{V}_h, \quad \alpha||u-u_h||_{\mathcal{V}}^2 &\leq a(u-u_h, u-u_h) && \text{(2.65a)}\\
&= a(u-u_h+v_h-v_h, u-u_h) && \text{(2.65b)}\\
&= a(u-v_h, u-u_h) + \underbrace{a(v_h-u_h, u-u_h)}_{=0,\text{ from (2.46) since } v_h-u_h\in\mathcal{V}_h} && \text{(2.65c)}\\
&= a(u-v_h, u-u_h) && \text{(2.65d)}\\
&\leq M||u-v_h||_{\mathcal{V}}||u-u_h||_{\mathcal{V}}. && \text{(2.65e)}
\end{aligned}$$

This result is rendered in the famous lemma of Céa:

Lemma 2.3.1: Céa's Lemma [Reddy, 1997, Theorem 10.1]
Let $\mathcal{V}_h$ be a closed sub-space of a Hilbert space $\mathcal{V}$, and let $a : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ be a continuous, coercive bi-linear form, with the respective coefficients M and α. Then there exists a constant C, independent of h, such that

$$||u-u_h||_{\mathcal{V}} \leq C \inf_{v_h\in\mathcal{V}_h} ||u-u_h||_{\mathcal{V}}, \quad \textit{with } C = \frac{M}{\alpha}. \tag{2.66}$$

Consequently, a sufficient condition for the Galerkin approximation to converge to the analytical solution is that there exists a family of discrete space $\{\mathcal{V}_h\}$ such that

$$\lim_{h\to 0} \inf_{v_h\in\mathcal{V}_h} ||u-u_h||_{\mathcal{V}} = 0. \tag{2.67}$$

This fundamental result has several implications on the discrete problem. First, the bi-linear $a(\cdot,\cdot)$ has to be *continuous* and *coercive*. Second, the sub-spaces require some approximation power such that the smallest error decays to zero. This is typically denoted as *completeness*. Third, the FE-spaces $\mathcal{V}_h$ have to be sub-spaces of $\mathcal{V}$. This is typically denoted as being *conforming*.

The formal proof that the Finite Element Method complies to these requirements and in fact yields a converging solution is given in e.g. [Brenner and Scott, 2008, Theorem 11.2.28], [Brenner and Scott, 2008, Section 5.3], [Reddy, 1997, Example 9.3.6], [Johnson, 2009, Example 2.3]. Instead of repeating these results at this point, this work aims at interpreting the above implications in terms of design rules for the mesh construction. To this end, it is assumed that $a(\cdot,\cdot)$ has the following form

$$a(\boldsymbol{u},\boldsymbol{v}) = \int_\Omega \frac{\partial^m \boldsymbol{v}}{\partial \boldsymbol{x}^m} : \mathbb{C} : \frac{\partial^m \boldsymbol{u}}{\partial \boldsymbol{x}^m}\, \mathrm{d}\Omega. \tag{2.68}$$

Here, the *variational index* m denotes the spatial derivative of the highest order. For the aforementioned Laplace-based problem considered in (2.26), m is equal to one. For other problems—such as e.g. in the case of beam or shell models—m can be larger.

The following outline is based on [Hughes, 2000, Section 3.1], [Felippa, 2013, Section 19] and [Ciarlet, 2002, Section 2.4]. Thereby, it has to be pointed out that the given requirements are sufficient but not necessary conditions for convergence.

Implications of the continuity of $a(\cdot, \cdot)$

In the context of structural engineering, the continuity of $a(\cdot, \cdot)$ implies that any finite displacement $\boldsymbol{u}$ introduces a finite energy into the elastic continuum. Clearly, this demands that the material parameters (e.g. Young's modulus) are finite. Furthermore, the m^{th}-derivative of $\boldsymbol{u}$ has to be square-integrable. Accordingly, the test and solution functions u and v have to be chosen from the Sobolev space of order m. Therefore, $\mathcal{V}(\Omega) = H^m(\Omega)$ when considering problems of type (2.68).

Implications of the coercivity of $a(\cdot, \cdot)$

Again considering the context of structural engineering, the demand for the coercivity of $a(\cdot, \cdot)$ implies that any non-zero deformation $\boldsymbol{u}$ has to result in a non-zero energy. First, this requires that the material parameter (e.g. Young's modulus) is greater than zero. Second, zero-energy modes have to be prevented.

In structural engineering, this applies to rigid body motions (translations and rotations). These have to be constrained by respective Dirichlet boundary conditions. To this end, the test and solution functions u and v are chosen from the admissible spaces defined in (2.24).

The numerical quadrature used to approximate the weak form numerically can also affect the coercivity. For example reduced integration schemes might result in spurious hourglass-shaped modes of zero-energy that have to be controlled (see e.g. [Hughes, 2000, Section 4.6]).

Implications of the completeness of the FE-space

In order for the numerical approximation to converge, the discrete spaces must have enough approximation power such that $\inf_{v_h \in \mathcal{V}_h} ||u - u_h||_{\mathcal{V}}$ actually decreases as h approaches zero. This can be ensured by using *m-complete* polynomial element shape functions, in the sense that "element shape functions must represent exactly all polynomial terms of order $\leq m$ in the Cartesian coordinates" [Felippa, 2013, Section 19]. Following [Hughes, 2000, Section 3.1], this is plausible as the element shape functions have to approximate the m^{th}-derivative (e.g the stress/strain state). Accordingly, this highest derivative of the shape functions should be at least an element-wise constant, non-zero function, such that the approximation error decreases as the element size decreases ($h \to 0$).

Implications of the conformity of the discrete space

As mentioned before, Céa's Lemma assumes that the discrete solution u_h is found in a space $\mathcal{V}_h$ that is a *sub*-space of $\mathcal{V}$, i.e. $\mathcal{V}_h(\Omega) \subset \mathcal{V}(\Omega)$. Although this seems to be a natural requirement from mathematical viewpoint, it has several practical implications when designing the finite element mesh.

Firstly, the used finite element mesh has to be conforming to the geometry. This requires that the elements of the mesh recover the original domain Ω, that is $\bigcup_{K_i \in \mathcal{T}_h} = \bar{\Omega}$. While this is no problem in the context of polygonal-shaped domains, this aspect has to be considered when solving problems on domains with curved boundaries.

Secondly, the basis functions used for the finite element approximation have to be contained in $\mathcal{V}$. Therefore, they have to fulfill the regularity constraints of the space. Here, the global

continuity of the basis functions is particularly important: the finite elements have to be *compatible*. For the general problem (2.68) with $\mathcal{V} = H^m(\Omega)$, this demands for a C^m-continuity of the shape functions within each cell K and a C^{m-1}-continuity between adjacent elements [Hughes, 2000, Section 3.1]. In the context of structural engineering ($m = 1$), this requirement implies that no gaps should open between elements when the mesh deforms due to an external load. When considering problems with a higher variational index or other physical phenomena—such as e.g. Maxwell's equation—the compatibility demands for e.g. an $H(\text{div})$- or $H(\text{curl})$-conforming discretization (see e.g. [Šolín, 2004]).

As discussed in the previous section, the standard Finite Element Method accounts for these constraints by a careful design of the element shape functions. However, this approach is only applicable when using regular discretizations. In the case of irregular meshes, ensuring the required inter-element continuity is an essential challenge. Therefore, this aspect is outlined in more detail when discussing the classical construction of *hp*-discretizations in Chapter 4. However, before addressing these details, the different versions of the Finite Element Method and their convergence properties are described in the next chapter.

Chapter 3

Finite element extensions and their convergence properties

As outlined in the previous section, the convergence of the finite element solution is guaranteed given that the bi-linear form and the finite element spaces meet the requirements of Céa's Lemma 2.3.1. The characteristic of this convergence—and with it the decay rate of the error—depends on the way that the finite element spaces are extended. Over the past decades, different strategies have been developed for maximizing the approximation quality for the specific problems under consideration. This chapter focuses on the three most prominent approaches: the *h*-, the *p*- and the *hp*-version of the Finite Element Method. The details of these refinement methods are described in Section 3.2, and their convergence properties are summarized in Section 3.3. To motivate the different types of refinement and examine their approximation error, the chapter begins with a one-dimensional benchmark example in the following section.

3.1 Motivating singular example

Céa's Lemma 2.3.1 suggests that the approximation quality of the finite element solution essentially depends on the smallest distance (metric) between the discrete function space $\mathcal{V}_h$ and the analytical solution u. Accordingly, the approximation error is influenced by the discrete function space *and* by the character of the analytical solution. To characterize the solution, the *smoothness* of the function is of essential importance. Following [Gui and Babuška, 1986a,b,c; Szabó and Babuška, 1991], the classical example of a one-dimensional, non-smooth function—which shall be the subject of the present section—is

$$u(x) = -x^{\alpha} + \alpha x, \quad \text{with } x \in \Omega = [0, 1], \tag{3.1}$$

where α is chosen as a non-integer value. If α is chosen smaller than one, the first derivative of u tends to infinity as x approaches zero. As this function-characteristic is common for singularities caused by re-entrant corners, the above function serves as a model problem for high-dimensional applications. For this reason, the results presented in the following also carry over to problems in two and three dimensions.

When choosing $\alpha > 1/2$, u and its derivative are square integrable, and thus u is contained in $H^1(\Omega)$. This renders the solution well suited for testing the performance of different FEM-refinement strategies. To this end, the following partial differential equation is considered:

$$-u'' = -\alpha(\alpha-1)x^{\alpha-2} \quad \forall\, x \in (0,1) \tag{3.2a}$$

$$\text{with} \quad u = 0 \quad \text{at } x = 0 \tag{3.2b}$$

$$\text{and} \quad u' = 0 \quad \text{at } x = 1. \tag{3.2c}$$

The solution of this boundary value problem is given by (3.1).

To find a finite element approximation to the analytical solution, the differential form is transferred into the variational formulation:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } a(v,u) = f(v) \quad \forall\, v \in H_0^1(\Omega), \tag{3.3}$$

with the linear and bi-linear form being

$$a(v,u) = \int_0^1 v'\, u' dx \quad \text{and} \quad f(v) = \int_0^1 -v\, \alpha(\alpha-1)x^{\alpha-2} dx, \tag{3.4}$$

and

$$H_0^1(\Omega) = \left\{ v \in H^1(\Omega) : v|_{x=0} = 0 \right\}. \tag{3.5}$$

For the numerical approximation of the solution, the above problem is restricted to a finite dimensional sub-space $\mathcal{V}_h \subset H_0^1$. This space is based on a finite element mesh with $m+1$ nodes, whose geometric positions x_i, with $i = 0, 2, \ldots, m$, are such that

$$0 = x_0 < x_1 < x_2 < \ldots < x_m = 0. \tag{3.6}$$

Between the nodes, m cells/intervals $I_i = (x_{i-1}, x_i)$ are introduced. On every interval, an element is defined using polynomials of degree p_i as element shape functions. At the nodes, the two non-zero modes of the adjacent elements are joined into one global function to ensure the C^0-continuity of the global basis functions. The approximation properties of this discretization for the aforementioned problem were analyzed in detail in the fundamental works of Gui and Babuška [1986a,b,c]. Closely following these descriptions, the essential findings are recapitulated here.

In their work, Gui and Babuška considered two different measures for every element. The first measure is the element size

$$h_i = x_i - x_{i-1}. \tag{3.7}$$

The second measure accounts for the distance of the element to the singularity. As the singularity is positioned in the origin ($x = 0$), this is done via the midpoint of the element

$$c_i = \frac{x_i + x_{i-1}}{2}. \tag{3.8}$$

This distance of the element to the singularity is taken relative to the element size in the following way

$$\eta_i = 2\frac{c_i}{h_i} = \frac{x_i + x_{i-1}}{x_i - x_{i-1}}. \tag{3.9}$$

This ratio is used to define the mesh design factor as

$$r_i = \frac{1}{\eta_i + \sqrt{\eta_i^2 - 1}} = \frac{\sqrt{x_i} - \sqrt{x_{i-1}}}{\sqrt{x_i} + \sqrt{x_{i-1}}}. \tag{3.10}$$

In their analysis, Gui and Babuška show that the error contribution of the element at the singularity decays differently than the contributions of the other elements. In [Gui and Babuška, 1986b, Theorem 1.1], they estimate the error of the first element as

$$||e||_{E(I_1)} \approx \frac{h_1^{\alpha-1/2}}{p_1^{2\alpha-1}}, \tag{3.11}$$

where "$\approx$ means that the ratio of the left and the right hand side is bounded above and below by equivalency constants which merely depend on α" [Gui and Babuška, 1986b]. For all other elements away from the singularity, the following estimate for the error is derived:

$$||e||_{E(I_i)} \approx \frac{h_i^{\alpha-1/2}}{\sqrt{1-r_i^2}} \frac{r_i^{p_i+1-\alpha}}{p_i^{\alpha}} \left(\frac{1}{p_i^{\alpha-1/2}} + (1-r_i^2)^{\alpha-1/2} \right) \text{ if } 0 < r_i^2 < 1 - \frac{1}{p_i} \tag{3.12a}$$

$$||e||_{E(I_i)} \approx h_i^{\alpha-1/2} \frac{r_i^{p_i+1-\alpha}}{p_i^{\alpha-1/2}} \left(\frac{1}{p_i^{\alpha-1/2}} + (1-r_i^2)^{\alpha-1/2} \right) \quad \text{if } 1 - \frac{1}{p_i} \leq r_i^2 \leq 1. \tag{3.12b}$$

These results show that the approximation error depends on the element sizes h_i, the polynomial orders p_i, and the mesh design reflected in r_i. From this observation, different refinement strategies can be derived, which are analyzed in the following.

The classical approach to control the error is to decrease the element sizes h_i uniformly while fixing the polynomial degree of all elements to a common value $p_i = p$. In this case, the number of unknowns N scales linearly with the number of elements, i.e. $N = \mathcal{O}(m) = \mathcal{O}(1/h)$.

According to estimate (3.1), the error of the first element decreases algebraically with a rate of $\alpha - 1/2$ when extending the function space in this way. Similarly, the error contributions of the other elements away from the singularity also decay algebraically. A more detailed analysis given in [Gui and Babuška, 1986b, Theorem 2.1] shows that the sum of all error contributions converge at a rate of p. It follows that the convergence rate of the total error is governed by either $\alpha - 1/2$ or p depending on which element dominates the overall decay. Accordingly, the convergence of a uniform h-refinement with a fixed p has the following algebraic form

$$||e||_E \leq C(\alpha, p) N^{-\beta} \quad \text{with } \beta = \min\left(p, \alpha - \frac{1}{2}\right), \tag{3.13}$$

where $C(\alpha, p)$ is positive and independent of h.

This theoretical result is confirmed by a numerical study performed with $\alpha = 0.65$, whose results are depicted in Figure 3.1. The approximation error decays algebraically with a rate

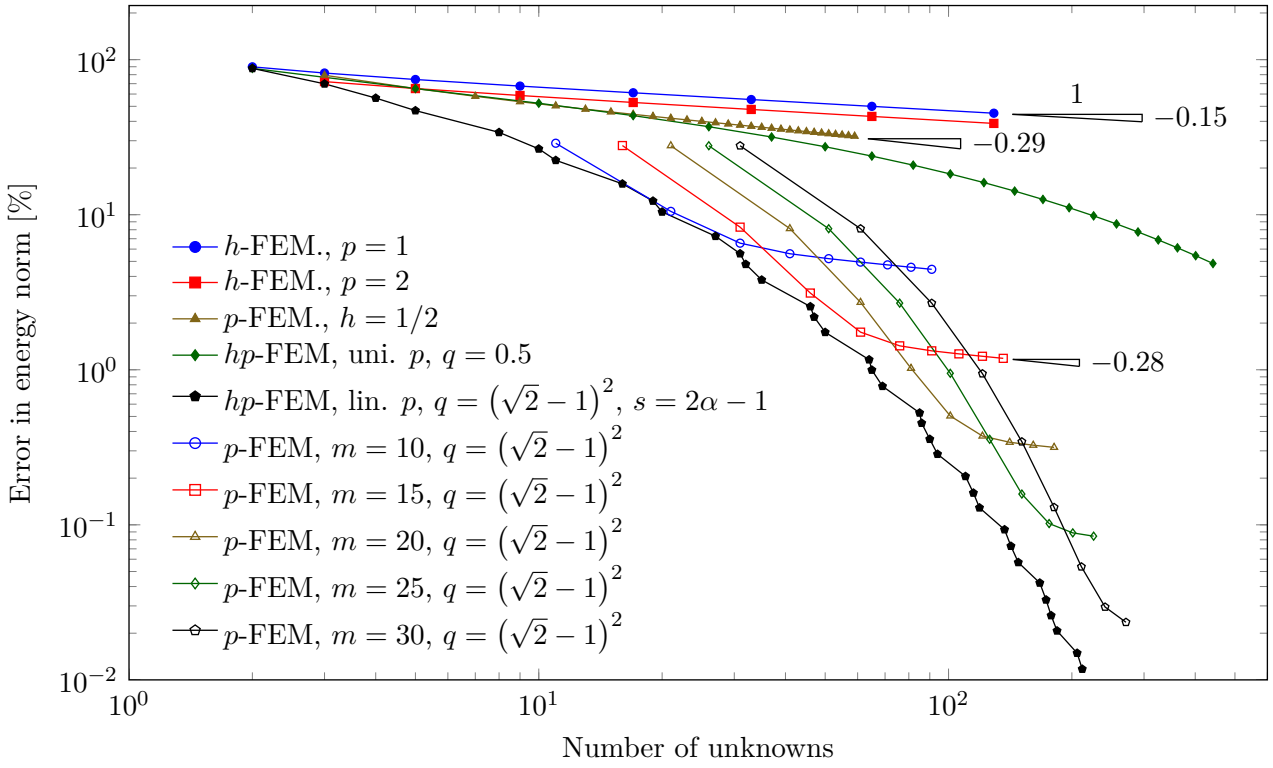


Figure 3.1: Comparison of the convergence of different FEM-versions for problem (3.2) with $\alpha = 0.65$. Used abbreviations: h: uniform element size, p: polynomial order of shape functions, hp: simultaneous change of h and p, uni.: uniform, lin.: linear, m: number of elements, q: ratio of geometric progression, s: ratio between m and p.

of 0.15, which corresponds to the above estimate for the selected value of α. The convergence characteristic does not change when increasing the polynomial degree of the used element shape functions. A further analysis presented in [Gui and Babuška, 1986b, Section 3.3] demonstrates that the convergence of a uniform h-refinement can never be better than algebraic even if the values of p_i are chosen non-uniformly.

An alternative to a uniform h-refinement is the uniform elevation of the polynomial degrees $p_i = p$, while keeping the element sizes unchanged. In this case, the number of unknowns N scale linearly with the ansatz order, i.e. $N = \mathcal{O}(p)$.

For this extension type, the estimates (3.12) predict that the error contributions of the elements away from the singularity decrease exponentially with respect to the base r. However, the error contribution of the element at the singularity only decreases algebraically, with a rate of $2\alpha - 1$. As this slower decrease of the error eventually dominates the overall convergence, the asymptotic decay of the total error is algebraic of the form

$$||e||_E \leq C(\alpha, p) N^{-\beta} \quad \text{with } \beta = 2\alpha - 1. \tag{3.14}$$

The numerical results presented in Figure 3.1 confirm these theoretical estimates. This first comparison demonstrates that a uniform order elevation yields twice the convergence rate of a uniform h-refinement.

Despite this remarkable advantage, the convergence of the global error is still limited to an algebraic decay due to the error contribution of the first element. To yield an exponential decrease of the global error, the mesh-family has to be designed such that also the error contribution of the first element decays exponentially. Accordingly, an exponential convergence demands for a *simultaneous* change of h_i and p_i.

One conceivable choice is to use two elements and to *relocate* the middle node such that the size—and with it the error contribution—of the first element decreases exponentially. With a simultaneous increase of the polynomial order p_2, also the error contribution of the second element decreases exponentially. However, as the position of the middle node x_1 approaches zero, the base of the exponential convergence r_2 convergences to one (see (3.10)). This renders the asymptotic convergence characteristic again algebraic if the approximation order is increased linearly.

This analysis demonstrates that for an exponential convergence of the overall error the mesh design has to a) decrease h_1 exponentially, b) increase the polynomial orders p_i, and c) keep r_i as small as possible. One—but not the only—possibility respecting these three demands is the use of a *geometrically graded mesh*. Here, the position of the nodes is given by a geometric progression

$$x_i = q^{m-i} \quad \text{with } i = 0, 1, 2, \ldots, m \quad \text{and } 0 < q < 1. \tag{3.15}$$

With this mesh design, the ratios r_i are equal for all elements

$$r_i = \frac{1 - \sqrt{q}}{1 + \sqrt{q}} \equiv r. \tag{3.16}$$

Accordingly, the error contributions of the elements away from the singularity converge exponentially with respect to the same base r when increasing their order p_i. At the same time, also the size of the first element—and with it its error contribution—decreases exponentially, while the number of elements m only increases linearly. When simultaneously increasing the polynomial degree p uniformly on all elements, the number of unknowns scales quadratically with p

$$N = \mathcal{O}(mp) = \mathcal{O}(p^2). \tag{3.17}$$

Accordingly, the global approximation error decays exponentially with respect to the square root of the number of unknowns N. A further analysis of the convergence presented in [Gui and Babuška, 1986b, Theorem 3.4] gives the following estimate of the approximation error

$$||e||_E \leq C(\alpha, q) \frac{1}{\sqrt{N^\sigma}} e^{-\kappa\sqrt{N}} \quad \text{with } \sigma = \min(\alpha, 2\alpha - 1), \tag{3.18}$$

with κ being positive and independent of N. This estimate is confirmed by the numerical study presented in Figure 3.1. The depicted results demonstrate the dramatic increase in accuracy gained by the *hp*-approach compared to a pure *h*- or *p*-refinement.

The performance of the refinement scheme can be further increased when—in addition to the element sizes h_i—also the polynomial order p_i are not chosen uniformly but increased linearly away from the singularity. For the present case, that is

$$p_i = \lceil 1 + s(i-1) \rceil \quad \text{with } i = 1, 2, 3, \ldots, m. \tag{3.19}$$

A more detailed analysis performed by Gui and Babuška shows that the convergence is optimal when using $s = 2\alpha - 1$ in combination with a grading of $q = \left(\sqrt{2} - 1\right)^2$ [Gui and Babuška, 1986b, Theorem 3.2]. The respective results presented Figure 3.1 demonstrate the superiority of theses optimal values.

The final refinement strategy to be address here concerns a uniform p-elevation on a mesh that is pre-graded (but fixed) towards the point of singularity. In this case, two different ranges of the convergence arise. In the first range, the contributions of the (larger) elements away from the singularity dominate the total error. As these contributions decay exponentially under an elevation of the ansatz order, also the overall convergence is exponential. However, the error of the first element only decreases algebraically. Hence, it eventually dominates the total error. At this point, the rate of convergence decreases, and the convergence becomes algebraic in the asymptotic limit. Accordingly, this second range is denoted as asymptotic range, whereas the first part is the pre-asymptotic range.

These two ranges are clearly visible in the S-like characteristic of the numerical results presented in Figure 3.1. Since the error contribution of the first element depends on the size h_1, the turning point separating the two ranges can be controlled by increasing the number of elements m. This is of essential importance, as it allows the desired accuracy to be reached with high convergence rates by choosing m accordingly.

The theoretical estimates and the numerical results presented in this section demonstrate the high gains of accuracy that can be achieved when combining a simultaneous change of h and p in the hp-version of the Finite Element Method. The following two sections will demonstrate that these advantage also carry over to higher-dimensional problems.

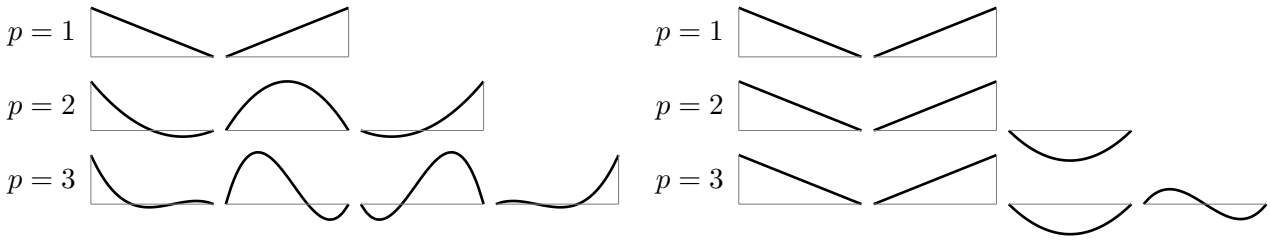
3.2 Finite element extensions

The previously discussed example motivates the use of a simultaneous refinement of the element size h and the polynomial order p for the approximation of singular solutions in the context of a one-dimensional problem. In this section, this hp-version of the Finite Element Method is formalized for higher dimensional applications. To this end, the section begins with a brief overview of the p-version of the Finite Element Method in the following section. Section 3.2.2 then outlines the extension to hp-FEM.

3.2.1 The p-version of the Finite Element Method

The essential idea of high-order finite elements is to improve the approximation quality by increasing the order p of the polynomial shape functions. Classical descriptions of the Finite Element Method are based on the use of Lagrange polynomials [Hughes, 2000; Zienkiewicz et al., 2005a]. These shape functions have the advantage that at so-called nodal points, only one shape function is non-zero as illustrated in Figure 3.2a. Therefore, this set of shape functions is interpolating at these points and in that sense naturally extend the nodal-based idea of linear finite elements. Unfortunately, the condition number of the stiffness matrix increases drastically with the polynomial order when using a Lagrange basis. This renders Lagrange polynomials beyond order three not well suited for simulations.

For this reason, the idea of the p-version of the Finite Element Method is to construct a set of polynomial element shape functions $\left\{\hat{\xi}_i\right\}$ such that the numerical properties of the stiffness



(a) 1D Lagrange shape functions (b) Integrated Legendre shape functions

Figure 3.2: Comparison of one-dimensional shape functions following [Düster et al., 2001].

matrix are as good as possible. In the optimal case, this change of shape functions would transform the stiffness matrix into the identity matrix. Following e.g. [Szabó and Babuška, 1991; Szabó et al., 2004], this procedure is outlined in the following.

For an one-dimensional problem, the stiffness matrix of an element K is computed as

$$A_{ij}^K = \int_K \frac{\partial \xi_i}{\partial x} E \frac{\partial \xi_j}{\partial x} \, \mathrm{d}x, \tag{3.20}$$

where E is a constitutive parameter that is assumed to be constant over K. To avoid off-diagonal terms, the optimal shape functions have to be orthogonal in the above sense. Functions with a comparable property are the Legendre polynomials

$$L_0(r) = 1 \tag{3.21a}$$
$$L_1(r) = r \tag{3.21b}$$
$$L_i(r) = \frac{1}{n} \left[(2n-1) \; r \, L_{n-1}(r) - (n-1) \; L_{n-2}(r) \right] \quad i = 2, 3, \ldots, p, \tag{3.21c}$$

which are defined in the interval $r \in [-1, +1]$. These shape functions are orthogonal in the sense that

$$\int_{-1}^{1} L_i L_j \, \mathrm{d}r = \begin{cases} \frac{2}{2i+1}, & \text{if } i = j \\ 0, & \text{else.} \end{cases} \tag{3.22}$$

To yield an optimized element stiffness matrix, this property has to be transferred to the function's first derivative. For this purpose, the p-version of the Finite Element Method is based on the integrated Legendre polynomials defined as

$$P_i(r) = \sqrt{\frac{2i-1}{2}} \int_{-1}^{r} L_{i-1}(t) \, dt = \frac{1}{\sqrt{4i-2}} \left(L_i(r) - L_{i-2}(r) \right) \quad i = 2, 3, \ldots . \tag{3.23}$$

The one-dimensional shape functions are then constructed by combining the classical linear

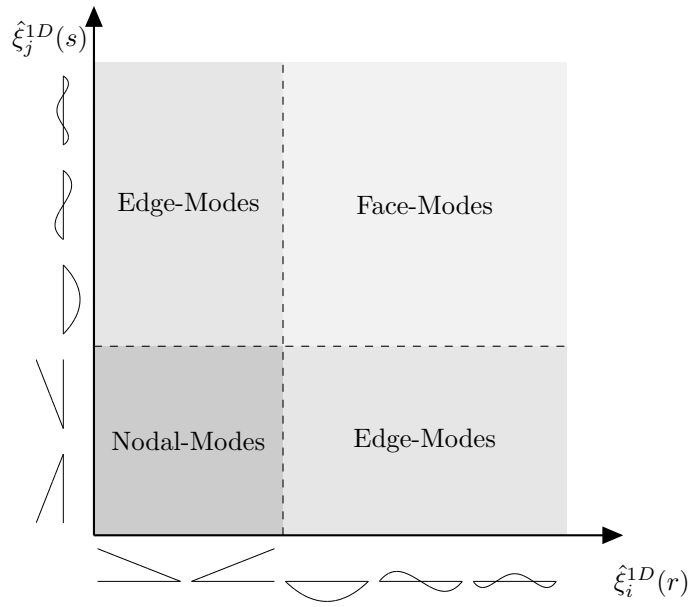


Figure 3.3: Tensor-product structure of the two-dimensional Legendre-based shape functions [Zander et al., 2014].

modes with the higher-order integrated Legendre polynomials

$$\hat{\xi}_1(r) = \frac{1}{2}(1+r) \tag{3.24a}$$

$$\hat{\xi}_2(r) = \frac{1}{2}(1-r) \tag{3.24b}$$

$$\hat{\xi}_i(r) = P_{i-1}(r) \quad i = 2, 3, \ldots, p+1. \tag{3.24c}$$

All but the linear modes are orthogonal to each other. Accordingly, the high-order stiffness matrix is almost diagonal with only two off-diagonal entries. This structure significantly simplifies the solution process and allows the use of high-order shape functions beyond degree three.

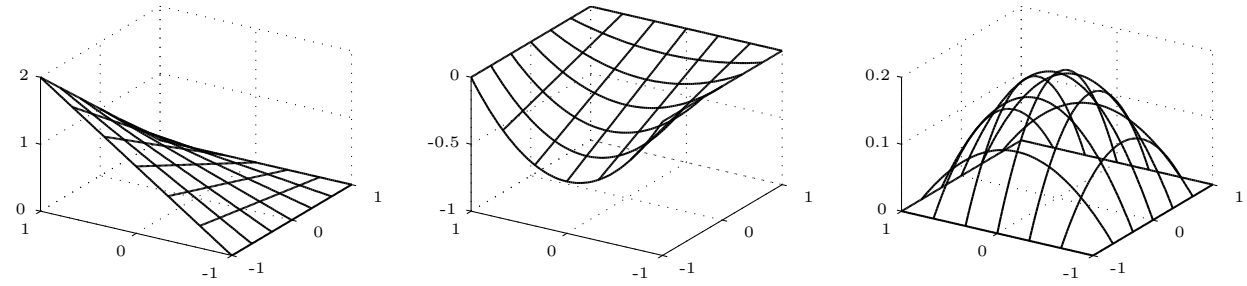
A second important difference to the Lagrange shape functions is that the integrated Legendre polynomials are hierarchical in the sense that the order of the basis is increased by only adding one additional function. All other functions remain unchanged. This is in contrast to the Lagrange basis, where every higher order basis is composed of a completely different set of shape functions. This difference is illustrated in Figure 3.2.

The two- or three-dimensional basis functions are obtained by a tensor product of the one-dimensional shape functions

$$\hat{\xi}_{i,j}^{2D}(r,s) = \hat{\xi}_i^{1D}(r)\,\hat{\xi}_j^{1D}(s) \tag{3.25a}$$

$$\text{and} \quad \hat{\xi}_{i,j,k}^{3D}(r,s,t) = \hat{\xi}_{i,j}^{2D}(r,s)\,\hat{\xi}_k^{1D}(t)\,. \tag{3.25b}$$

As depicted in Figure 3.3, the elements of the two-dimensional tensor-product space can be sorted into the following three groups:



(a) Nodal-mode (b) Edge-mode (c) Face-mode

Figure 3.4: Two-dimensional mode types following [Szabó and Babuška, 1991].

Nodal-mode The combination of two one-dimensional nodal-modes gives rise to the well known bi-linear modes (see Figure 3.4a). Just as in 1D, these can be associated directly to one node giving the group its name.

Edge-mode The combination of a one-dimensional internal-mode and a one-dimensional nodal-mode results in a function that is non-zero at only one edge (see Figure 3.4b). Therefore, the mode can be directly associated to that edge and, thus, identified as a two-dimensional edge-mode.

Face-mode The combination of two one-dimensional internal-modes yields a function that is zero on all four nodes and edges (see Figure 3.4c). The mode can, therefore, be directly associated to the face and thus, identified as a bubble- or face-mode.

In the case of three-dimensional shape functions, also internal- or volume-modes have to be considered[f].

This mode-concept is of great advantage for the definition of C^0-continuous global basis functions $\{N_i\}$ that span the finite element space $\mathcal{V}_h$. As discussed in Section 2.2.2, each global basis function N_i is associated to either a node, an edge, a face, or—in a three-dimensional case—to a solid. A globally continuous nodal basis function can be defined as the set of the respective nodal shape functions of the elements adjacent to the node. In the same way, the basis functions associated to an edge are defined as the assembly of the respective edge-modes of the same degree of the elements adjacent to the respective edge. The global shape functions associated to a face directly correspond to the face modes of the respective element.

3.2.2 The *hp*-version of the Finite Element Method

As demonstrated for the one-dimensional benchmark setting in Section 3.1, a pure elevation of the approximation order is not sufficient to yield an exponential decay of the error in the case of singularities. Instead, also the element sizes and the mesh design have to be adapted.

In the context of the one-dimensional example discussed in Section 3.1, it was demonstrated that a geometrically graded mesh achieves exponential convergence also in the presence of singularities. Besides the remarkable gain in accuracy, the choice of a geometrically graded mesh has the advantage that the family of meshes can be generated by recursively *sub-dividing* the element(s) at the singularity in a ratio of $q : (1-q)$. As the geometry of all other elements remains unchanged, no re-generation of the complete mesh is necessary. In two- and three-

dimensional problems, this is of great advantage, which is why this type of *hp*-refinement is predominantly used.

One common cause of singularities in applications of higher dimensions are re-entrant corners in the geometry. This case is denoted as point- or vertex-singularity. The geometric meshes designed for this kind of problems show an inherent layer structure [Babuška and Guo, 1996]: the zero-th layer $\mathcal{L}_0$ consists of all elements whose domains include the singular vertex. The first layer $\mathcal{L}_1$ includes all elements adjacent to the zero-th layer and so on. Thus, a higher-dimensional geometric mesh family is formally defined as

Definition 3.2.1: Geometric mesh family [Schwab, 1998, Definition 4.30]
A geometric mesh family $\mathcal{T}_q^k$ *with* $k+1$ layers *and a* grading factor $0 < q < 1$ *on the domain* Ω *with a singular vertex* $v \in \bar{\Omega}$ *is defined recursively as follows: if* $k = 0$, $\mathcal{T}_q^0 = \{\Omega\}$. *Given* $\mathcal{T}_q^k$, *generate* $\mathcal{T}_q^{k+1}$ *by sub-dividing the elements whose closure includes* v *with a ratio* $q : (1-q)$.

For an exponential convergence of the approximation error, the graded mesh structure has to be combined with high-order shape functions. To control the condition number of the resulting stiffness matrix, the high-order shape functions are based on the tensor product of the integrated Legendre shape functions discussed in Section 3.2.1. As demonstrated in Section 3.1, the decay rate of the approximation error can be significantly improved by linearly increasing the approximation order away from the singularity. These advantages also carry over to higher-dimensional problems. For this reason, the polynomial degree of the respective element is chosen according to its layer. For every layer j away from the point of singularity, the polynomial degree is increased linearly with a slope s

$$p_j = \lceil 1 + sj \rceil, \quad \text{with } j = 0, 1, 2, \ldots, k. \tag{3.26}$$

This combination of a simultaneous change of h and p reaches an exponential decay of the approximation error also in presence of a singularity. The details of the convergence properties are outlined the following section.

3.3 Summary of convergence properties

Closely following [Szabó et al., 2004; Yosibash, 2012], this section addresses the convergence properties of the aforementioned *h*-, *p*-, and *hp*-refinement methods. In addition to the chosen refinement strategy, the convergence characteristic also depends on the nature of the solution that is to be approximated. For this reason, the overview begins with a classification of the solution. Afterwards, the types of convergence are formally defined. In the final part of this section, the convergence properties of the different finite element extensions are summarized.

3.3.1 Problem classification

For the approximation quality of the finite element solution, the smoothness of the analytical solution is of essential importance. For this reason, the problems are classified depending on whether their solution is analytic. A function is analytic if it can be represented locally by a convergent power series [Krantz and Parks, 2002, Definition 1.1.3]. Using this criterion, Szabó *et al.* consider the following three different types of problems [Szabó et al., 2004]:

Category A: The exact solution u is analytic at all points in the domain, including its boundary.

Category B: The exact solution u is analytic at all points in the domain, including its boundary, except for a finite number of locations. These are denoted as singular points or singular edges.

Category C: The exact solution is neither in category A nor in B.

As noted in [Szabó et al., 2004], the solution in the vicinity of corner singularities in two-dimensional problems commonly has an asymptotic expansion of the following form:

$$\boldsymbol{u}(r,\theta) = \sum_{i=1}^{\infty} A_i \, r^{\alpha_i} \, \boldsymbol{F}(\theta), \tag{3.27}$$

where (r,θ) denote the polar coordinates, A_i and α_i are real numbers and $\boldsymbol{F}(\theta)$ is an analytic function. An analog expansion can be derived for three-dimensional problems. The comparison to the one-dimensional singular solution (3.1) reveals that the above expansion is very similar to the one-dimensional benchmark solution. For this reason, most observations made in Section 3.1 carry over to higher dimensions. Depending on the exponents $\{\alpha_i\}$, the solution itself or its derivatives might tend to infinity as r approaches zero. Therefore, $\alpha := \min_i \alpha_i$ characterizes the *regularity* of the solution.

3.3.2 Convergence properties

As discussed in the previous section, the strategies of how to increase the approximation accuracy of three versions can be summarized as follows:

***h*-FEM:** The approximation quality is increased by uniformly decreasing the size h of all elements, while keeping the order of the polynomial shape functions fixed, typically at $p = 1$ or $p = 2$.

***p*-FEM:** The approximation quality is increased by uniformly elevating the order p of the polynomial shape functions, while keeping the size of the elements fixed. For problems of category B, it is assumed that the singular points or edges are resolved by the mesh.

***hp*-FEM:** The approximation quality is increased by simultaneously decreasing the element size h and elevating the order p of the shape functions. For problems of category B, the mesh structure is graded towards the singular points or edges according to Definition 3.2.1, and the polynomial order is increased linearly over the layers according to (3.26).

As discussed in the context of the one-dimensional benchmark, there are two different ways in which the global approximation error can decay. If the asymptotic decrease can be bounded from above in the following way:

$$||u - u_h||_E \leq \frac{k}{N^{\beta}}, \tag{3.28}$$

the convergence is said to be *algebraic* with rate β. Here, k and β are positive constants dependent on u but independent of N. On the other hand, the convergence is considered as

Category	h-FEM	p-FEM	hp-FEM
A	algebraic $\beta = p/2$	exponential $\theta \geq 1/2$	exponential $\theta \geq 1/2$
B	algebraic $\beta = \frac{1}{2}\min(\alpha, p)$	algebraic $\beta = \alpha$	exponential $\theta \geq 1/3$
C	algebraic $\beta > 0$	algebraic $\beta > 0$	Remark 3.3.1

Table 3.1: FEM convergence for two-dimensional problems following [Szabó et al., 2004; Yosibash, 2012].

Category	h-FEM	p-FEM	hp-FEM
A	algebraic $\beta = p/3$	exponential $\theta \geq 1/3$	exponential $\theta \geq 1/3$
B	Remark 3.3.2		exponential see Table 3.3
C	algebraic $\beta > 0$	algebraic $\beta > 0$	Remark 3.3.1

Table 3.2: FEM convergence for three-dimensional problems following [Szabó et al., 2004; Yosibash, 2012].

exponential if the global approximation error can be bounded from above in the following way:

$$||u - u_h||_E \leq \frac{k}{e^{\gamma N^\theta}}. \tag{3.29}$$

Similar to the previous case, k, θ are γ are positive constants, which depend on u but not on N.

In the past decades, *a priori* estimates of the approximation error of the different versions of the Finite Element Method applied to problems of the aforementioned categories were derived in e.g. [Babuška and Guo, 1988, 1996; Babuška and Suri, 1994; Guo and Babuška, 1986b; Schwab, 1998]. Following [Szabó et al., 2004; Yosibash, 2012], these results are presented in the Tables 3.1 and 3.2.

The summary demonstrates that, for problems of category A, both, the p- and the hp-version of the Finite Element Method yield an exponential convergence. If instead the approximation quality is increased by decreasing the element size h, the error only algebraically.

If the problem is of category B, the p-extension yields only an algebraic decrease of the error. However, the rate β is at least twice as large as the rate obtained by a uniform h-refinement. These findings are in agreement with those made in the context of the one dimensional benchmark discussed in Section 3.1. To yield an exponential decay of the approximation error for problems of this category, the hp-version of the Finite Element Method has to be used. In the

Singularity Type:	Regular	Vertex	Edge	Edge-Vertex
θ	1/3	1/4	1/4	1/5

Table 3.3: Exponent of *hp*-convergence for different types of singularities of three-dimensional problems following [Babuška and Guo, 1996; Yosibash, 2012].

case of two-dimensional applications, the convergence is (at least) exponential with respect to the third-root of the number of unknowns (as $\theta \geq 1/3$). For three-dimensional problems, the exponent θ depends on the type of singularity. Following [Babuška and Guo, 1996; Yosibash, 2012], the values of the most common types of singularities are summarized in Table 3.3.

This comparison of the convergence properties demonstrates the superiority of the *hp*-version of the Finite Element Method in case of solutions with singularities. However, the construction of *hp*-graded meshes in higher dimensions bears some challenges. These aspects are addressed in the two following chapters.

Remark 3.3.1. "When u has a recognizable structure, then it is possible to achieve faster than algebraic rates of convergence with hp-adaptive methods" [Szabó et al., 2004].

Remark 3.3.2. "In three dimensions, u cannot be characterized by a single parameter. Nevertheless, the rate of p-convergence is at least twice the rate of h-convergence." [Szabó et al., 2004].

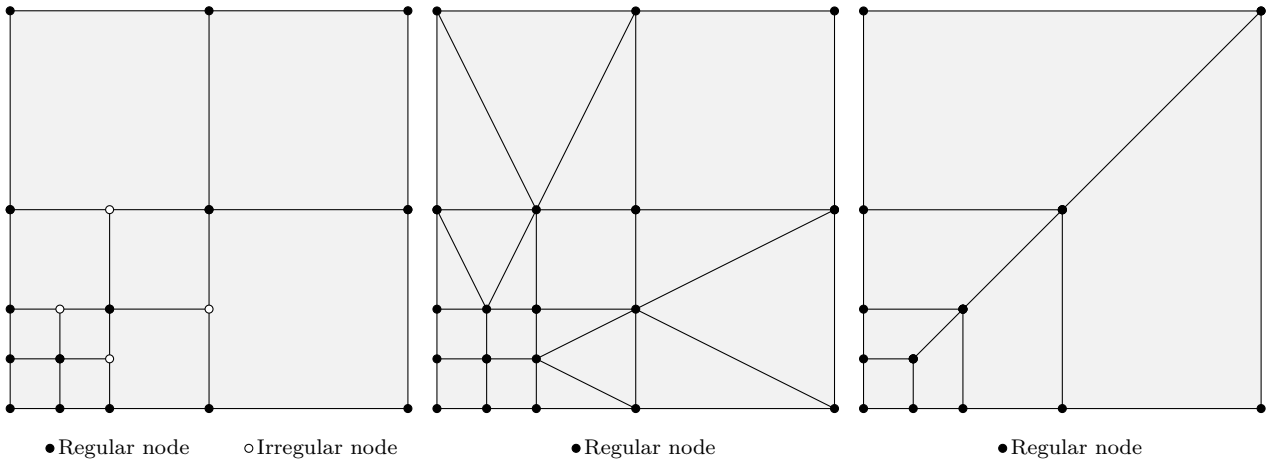
Chapter 4

Classical construction of *hp*-meshes: refinement by replacement

The previous chapter outlined that the *hp*-version of the Finite Element Method yields an exponential decay of the approximation error even for non-analytical solutions. The cornerstone of the method is the simultaneous elevation of the shape function order p combined with a graded decrease of the element size h towards the point(s) of singularity. Typically, a geometric grading of the elements is chosen as these mesh families can be constructed by recursively sub-dividing the elements at the singularity in a ratio of $q : (1 - q)$. In this way, the changes to the mesh remain local as the *hp*-meshes can be constructed by simply *replacing* the elements at the singularity by a new layer of finer elements as illustrated in Figure 4.1a. This approach is of great advantage as no global re-meshing is necessary, and it has proven to yield excellent results in different applications in e.g. [Cantwell et al., 2015, 2014; Korous and Šolín, 2012; Payette, 2012; Schnepp and Weiland, 2012; Šolín and Korous, 2012b].

The major challenge of this "refine-by-replacement" approach is that the discretization becomes non-matching as the shape functions of the fine-scale elements have no corresponding counterpart in the coarse, unrefined neighbors (see Figure 4.1a). The mesh is therefore *irregular* as discussed in Section 2.2.2. Accordingly, the standard assembly procedure described in Algorithm 1 cannot be used to ensure the inter-element continuity. For this reason, irregular meshes cannot be used directly for the construction of a finite element approximation. Instead, special measures have to be taken to restore the inter-element continuity of the shape functions.

This chapter outlines the most prevalent approaches for the construction of *hp*-discretizations. Here, two different types of approaches are considered. The first type restores the regularity of the discretization by introducing additional elements that form the transition between the refined and the coarse parts of the mesh. Different approaches of this type are outlined in Section 4.1. The second type of methods—prevalent in *hp*-applications—uses an irregular discretization to find the finite element approximation. To this end, the assembly procedure is generalized to appropriately constrain the shape functions associated to hanging nodes. This constrained approximation concept is outline in Section 4.2.



(a) Irregular mesh (b) Regular hybrid mesh (c) Regular quadrilateral mesh

Figure 4.1: Regular and irregular *hp*-discretization.

4.1 Transition elements

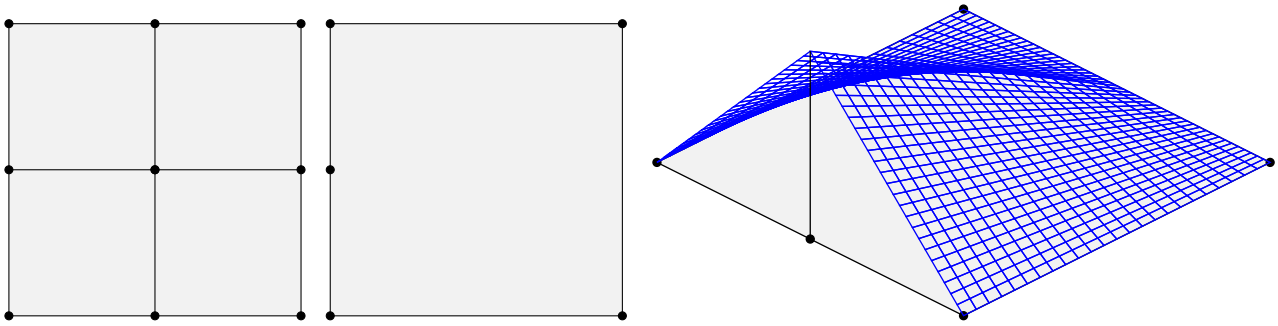
The simplest strategy to deal with irregular meshes is their conversion into a regular discretization. To this end, not only the elements at the singularity are sub-dived but also their neighbors. In this way, a regular transition between the refined and coarse part of the mesh can be formed as illustrated in Figure 4.1b.

As outlined in [Mitchell, 1989; Verfürth, 1996], classical approaches for the local refinement of triangular meshes are the longest-edge-bisection [Rivara, 1984a,b,c], newest-vertex-bisection [Kossaczký, 1994; Sewell, 1972], and the regular/red-green refinement [Bank and Sherman, 1980, 1981]. All three methods ensure the regularity of the mesh by performing a "clean-up" step after the actual refinement that sub-divides irregular elements accordingly.

In the case of quadrilateral meshes, the regularity of the mesh can be restored by means of triangular transition elements as illustrated in Figure 4.1b. Under certain restrictions on the refinement, the transition can also be formed by only quadrilateral elements as discussed in e.g. [Babuška and Rank, 1986, 1987; Rank and Babuška, 1987; Schneiders and Debye, 1995]. As illustrated in Figure 4.1c, certain template solutions are derived that are then used for the respective case under consideration. This approach has been extended to three-dimensional problems in [Schneiders, 2000; Schneiders and Debye, 1995].

The idea of restoring the mesh regularity via transition elements has been applied successfully in the context of *hp*-FEM in e.g. [Eibner and Melenk, 2006; Schwab, 1998; Schwab and Suri, 1999]. However, when using transition elements, the refinement spreads out over lager parts of the domain. Thus, the refinement loses its local character, which introduces unnecessary degrees of freedom as shown in e.g. [Šolín et al., 2010a]. Moreover, special care has to be taken to prevent distorted elements as these would decrease the approximation quality [Babuška and Aziz, 1976; Mitchell, 1989]. Furthermore, the automatic creation of transition elements is challenging in particular for three-dimensional problems.

An alternative idea to account for hanging nodes is to use variable-node elements for the transition between the refined and the coarse parts of the mesh. The essential idea of this



(a) Matching interface by variable node in transition element.

(b) Additional shape function of variable node.

Figure 4.2: Transition by variable node element following [Gupta, 1978].

approach—first introduced by Gupta in [Gupta, 1978]—is to define transition elements that have an additional node on one of their sides. This additional node can then form the transition to the fine elements. This idea is depicted in Figure 4.2. The approach has been extended successfully in e.g. [Cho and Im, 2006; Choi and Lee, 1996; Kim et al., 2008; Lim et al., 2012] also to three-dimensional problems and meshes with a higher irregularity index.

A similar idea was followed in the context of the eXtended Finite Element Method (XFEM) in e.g. [Cheng and Fries, 2012; Fries et al., 2011]. Their approach was to extend the function space of the transition element by an additional hat function that recovers the continuity of the basis functions.

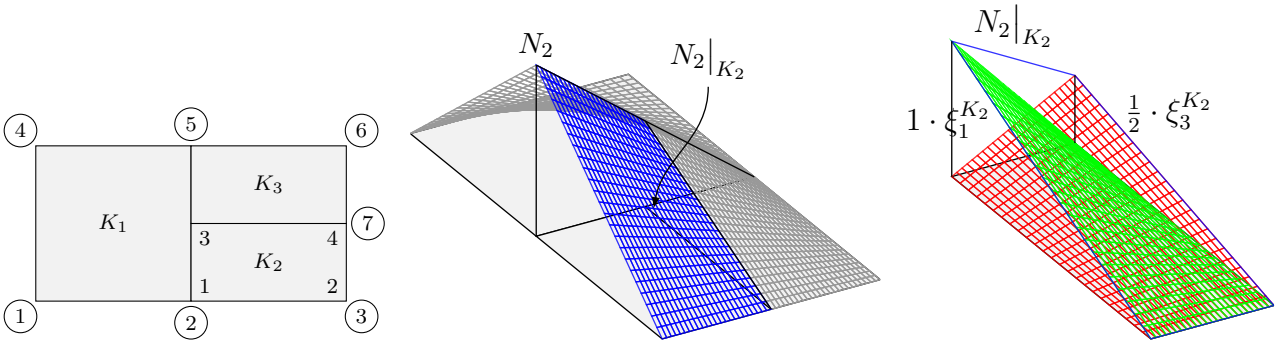
Both approaches resolve the irregularity of the discretization in an elegant way and have shown to yield an optimal convergence under a uniform refinement of the element size. However, to the author's knowledge, the formulation of these methods is so far restricted to the use of low-order shape functions and has not been used to construct *hp*-discretizations.

4.2 Constrained approximation

The prevalent approach in *hp*-finite elements is based on a *constrained approximation* of the analytical solution on irregular meshes [Ainsworth and Oden, 1997; Ainsworth and Senior, 1999; Demkowicz, 2007; Demkowicz et al., 1989; Frauenfelder, 2004; Kus, 2011; Rheinboldt and Mesztenyi, 1980; Schröder, 2011; Šolín, 2004; Šolín et al., 2008, 2010a]. As addressed in the introduction of this chapter, the major challenge of an irregular discretization is that the shape functions of neighboring refined and coarse elements are non-matching. This renders the construction of globally continuous basis functions significantly more complex compared to the case of regular meshes. In particular, the standard assembly procedure outlined in Section 2.2.2 has to be generalized as it is based on the assumption of a direct correspondence between *one* global basis function N_i and *one* element shape function ξ_j, in the sense

$$N_i|_K = \xi_j^K. \tag{4.1}$$

In the case of an irregular discretization, this simple nature of the one-to-one connectivity is lost. This becomes immediately evident when considering the simple scenario depicted



(a) Global and local degree of freedom numbering.

(b) Definition of global basis function N_2.

(c) Element-wise reconstruction.

Figure 4.3: Connection between global basis functions and local element shape functions in case of an irregular discretization.

in Figure 4.3. Here, the basis function N_2 has no directly corresponding local shape function in the second (and third) element for which condition (4.1) holds. Instead, this global basis function has to be re-constructed by combining the first and third element shape functions of the second element with a weight of 1 and 1/2, respectively. Accordingly, the basis function N_2 can only be given by a linear combination of multiple element shape functions. Using the connectivity matrix $\boldsymbol{\pi}$ introduced in Section 2.2.2, this reads

$$N_i|_K = \sum_j \pi_{ij}^K \xi_j^K \tag{4.2}$$

For the case depicted in Figure 4.3, the second row of the connectivity matrix $\boldsymbol{\pi}^{K_2}$ is given by

$$\left(\pi_{2j}^{K_2}\right)_{j=1,\dots,4} = \left[1, \quad 0, \quad \tfrac{1}{2}, \quad 0\right]. \tag{4.3}$$

Clearly, this connectivity is no longer one-to-one. Moreover, the values of π_{ij}^K are not necessarily one or zero, but any element in $\mathbb{R}$ is possible.

Accordingly, the *constrained approximation* is a two-step approach: first, the respective constraints π_{ij}^K that ensure the continuity have to be derived, and second, these constraints have to be incorporated in the assembly process. These two steps are outlined in the following two sections.

4.2.1 Derivation of constraint coefficients

The constraint coefficients ensuring the C^0-continuity of the basis functions can be computed in different ways as shown in e.g. [Ainsworth and Oden, 1997; Ainsworth and Senior, 1999; Byfut and Schröder, 2012; Demkowicz, 2007; Demkowicz et al., 1989; Frauenfelder, 2004; Kus, 2011; Kus et al., 2014; Schröder, 2011; Šolín, 2004; Šolín et al., 2008, 2010a]. The following outline is based on the principle setting depicted in Figure 4.4, in which a coarse element K_c is neighboring a refined element K_r. The two elements share an irregular interface (edge) γ.

The common starting point of most of the aforementioned approaches is expression (4.2) formulating the re-construction of a global shape function N_i as a linear combination of element

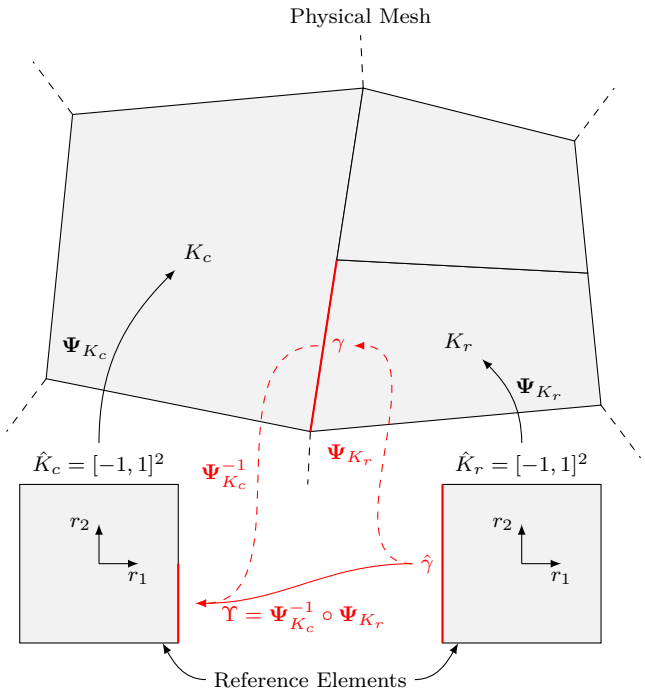


Figure 4.4: Depiction of geometric mappings used to determine the constraint coefficients of an irregular mesh.

shape functions ξ_j defined on the element K. As the equality holds on the complete element K, it has to hold—in particular—on the interface γ. Accordingly, the values of N_i along γ can be expressed in terms of the element shape functions defined on the refined element K_r in the following way

$$N_i(\boldsymbol{x}) = \sum_{j \in J} \pi_{ij}^{K} \xi_j^{K_r}(\boldsymbol{x}) \quad \forall\, \boldsymbol{x} \in \gamma. \tag{4.4}$$

Here, J denotes the set of indices of the element shape functions that are non-zero on the interface γ.

Observing that the coarse element K_c is not refined, the restriction of the global shape function N_i to K_c directly corresponds to *one* local shape function $\xi_l^{K_c}$ of this element, that is

$$N_i|_{K_c} = \xi_{l(i)}^{K_c}. \tag{4.5}$$

Since $\gamma \subset K_c$, this relation can be used to replace the left hand side expression in (4.4). This allows one to reformulate the constraint equation solely in terms of *element* shape functions defined on the two elements K_c and K_r as follows

$$\xi_{l(i)}^{K_c}(\boldsymbol{x}) = \sum_{j \in J} \pi_{ij}^{K_r} \xi_j^{K_r}(\boldsymbol{x}) \quad \forall\, \boldsymbol{x} \in \gamma. \tag{4.6}$$

According to Definition 2.2.8, the element shape functions ξ can be expressed in terms of standard element shape functions $\hat{\xi}$ defined in the reference space $\hat{K}$ using the geometric

mappings $\boldsymbol{\psi}_{K_c}$ and $\boldsymbol{\psi}_{K_r}$ of the two elements

$$\xi_j = \hat{\xi}_j \circ \boldsymbol{\Psi}_K^{-1}. \tag{4.7}$$

As a result, (4.6) can be expressed in terms of *standard* element shape functions

$$\hat{\xi}_{l(i)}^{K_c} \circ \boldsymbol{\Psi}_{K_c}^{-1}(\boldsymbol{x}) = \sum_{j \in J} \pi_{ij}^{K_r} \hat{\xi}_j^{K_r} \circ \boldsymbol{\Psi}_{K_r}^{-1}(\boldsymbol{x}). \quad \forall\, \boldsymbol{x} \in \gamma. \tag{4.8}$$

Combining the geometric mappings of the two elements as follows

$$\boldsymbol{\Upsilon} = \boldsymbol{\Psi}_{K_c}^{-1} \circ \boldsymbol{\Psi}_{K_r} \tag{4.9}$$

allows the formulation of the constraint equation in the reference space of the refined element

$$\hat{\xi}_{l(i)}^{K_c} \circ \boldsymbol{\Upsilon}(\boldsymbol{r}) = \sum_{j \in J} \pi_{ij}^{K_r} \hat{\xi}_j^{K_r}(\boldsymbol{r}) \quad \forall\, \boldsymbol{r} \in \hat{\gamma}. \tag{4.10}$$

This final form shows that the constraint coefficients depend on two aspects: a) the position of the two elements relative to each other (expressed in the mapping $\boldsymbol{\Upsilon}$) and b) the standard element shape functions $\hat{\xi}$ along the interface. With this information, the constraint coefficients $\pi_{ij}^{K_r}$ can be computed by collocating the above equation at a sufficient number of points on the interface $\hat{\gamma}$ and solving the resulting system of linear equations for $\pi_{ij}^{K_r}$.

In practice, however, the refinement is typically restricted to halving the long element side. Accordingly, the complexity of the geometric mapping can be reduced to two cases since the refined edge is either on the left or on the right half of the large edge. In this case, only the following two geometric mappings are possible

$$\Upsilon_L(r) = \frac{1}{2}(1-r) \quad \text{and} \quad \Upsilon_R(r) = \frac{1}{2}(1+r). \tag{4.11}$$

Further assuming that the shape functions are the same for all elements, the values of $\boldsymbol{\pi}$ can be pre-computed for both mappings in the setup phase of the finite element computation. This allows for an efficient re-use during the actual computation.

Moreover, if the set of basis function $\{\hat{\xi}_j\}$ is hierarchical under the elevation of the polynomial order p, this hierarchic nature carries over to the matrices $\boldsymbol{\pi}$. This allows to only compute the matrices for the highest polynomial degree used. During the assembly process the respective sub-matrices can be extracted. This property renders the constrained approximation approach particularly suitable for the commonly used integrated-Legendre shape functions. Furthermore, the constraint coefficients can be computed explicitly for this type of shape functions as shown in e.g. [Schröder, 2008]. Following [Ainsworth and Oden, 1997], the constraint coefficient matrices for a polynomial degree $p = 5$ are for example given by[1]:

$$\boldsymbol{\pi}^L = \begin{bmatrix} 1 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & -1/2 & 1/4 & 0 & 0 \\ 0 & 0 & -3/8 & 1/8 & 0 \\ 0 & 1/8 & 3/16 & -5/16 & 1/16 \end{bmatrix} \quad \boldsymbol{\pi}^R = \begin{bmatrix} 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 & 0 \\ -1/2 & 0 & 1/4 & 0 & 0 \\ 0 & 0 & 3/8 & 1/8 & 0 \\ 1/8 & 0 & 3/16 & 5/16 & 1/16 \end{bmatrix}. \tag{4.12}$$

[1]The constraint coefficients are given for the standard integrated Legendre polynomials and not for their normalized version discussed in Section 3.2.1.

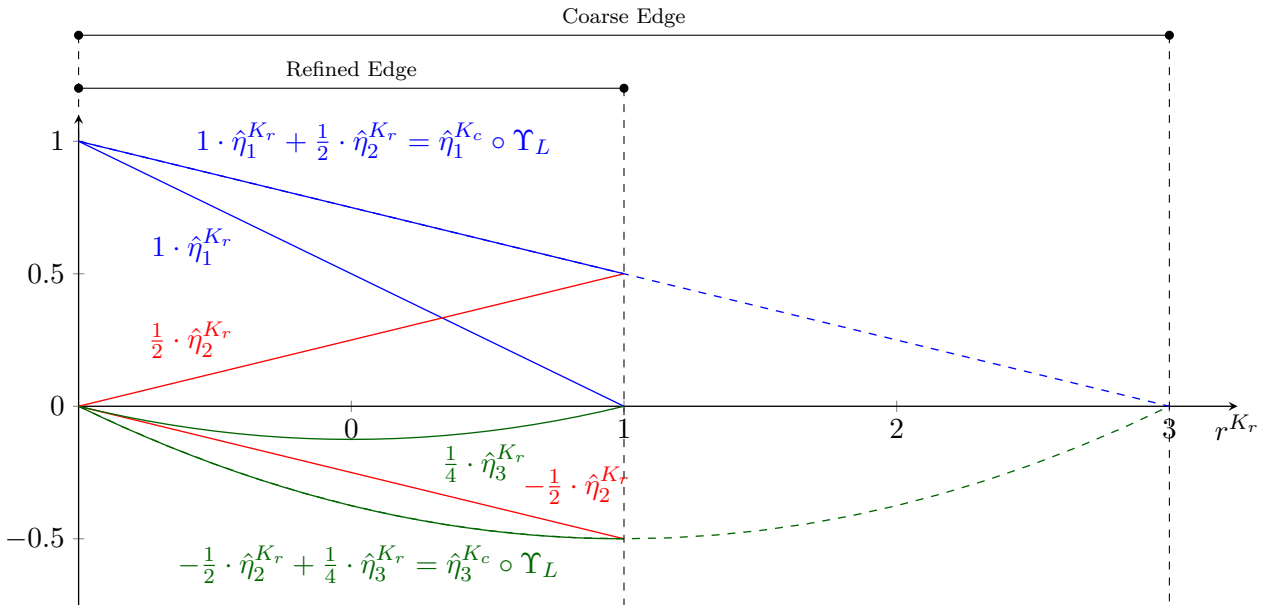


Figure 4.5: Re-construction example of coarse shape functions by refined shape functions.

Accordingly, the first shape function along the edge of the coarse element can be re-constructed by combining the first and second shape function along the edge of the refined element in the following way:

$$\hat{\eta}_1^{K_c} \circ \Upsilon_L = 1 \cdot \hat{\eta}_1^{K_r} + \frac{1}{2} \cdot \hat{\eta}_2^{K_r}. \tag{4.13}$$

Similarly, the coarse quadratic mode can be re-constructed by combining the second fine-scale linear mode and the fine-scale quadratic mode as:

$$\hat{\eta}_3^{K_c} \circ \Upsilon_L = -\frac{1}{2} \cdot \hat{\eta}_2^{K_r} + \frac{1}{4} \cdot \hat{\eta}_3^{K_r}. \tag{4.14}$$

These two re-construction examples are illustrated in Figure 4.5.

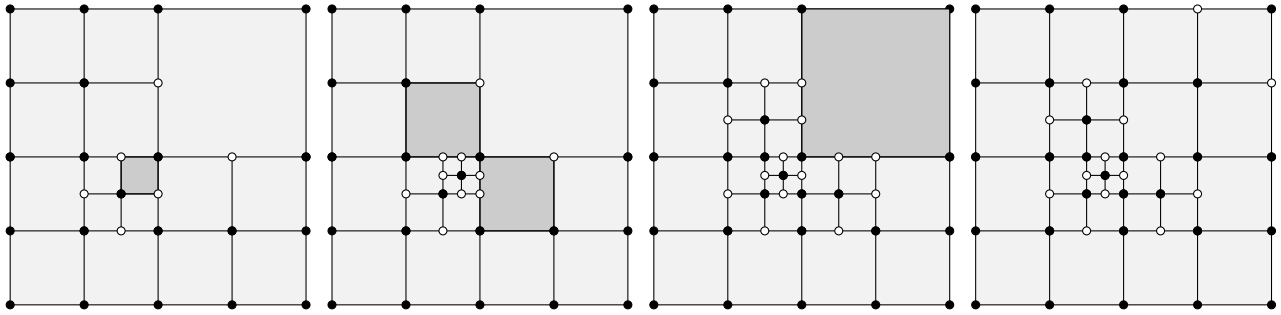
4.2.2 Generic assembly procedure

Having derived the constraint coefficients, these can be incorporated in the final system of equations in a *generic* assembly procedure. To formalize this procedure, let the weak form of the considered problem read as in (2.37a)

$$\text{Find } u \in \mathcal{V}_h(\Omega) \text{ such that } a(v,u) = f(v) \quad \forall v \in \mathcal{V}_h(\Omega), \tag{4.15}$$

with $\mathcal{V}_h$ denoting the space spanned by the global shape functions N_i. Using the Einstein summation convention and the linear combination (4.2), the linear functional on the right hand side can then be written in the following form

$$f(N_i) = \sum_{K \in \mathcal{T}_h} f(\pi_{ij}^K \xi_j^K) = \sum_{K \in \mathcal{T}_h} \pi_{ij}^K f(\xi_j^K) \tag{4.16}$$



(a) Initial mesh: the marked element is to be refined.

(b) Clean-up 1: the direct neighbours have to be refined.

(c) Clean-up 2: the neighbours of the neighbours have to be refined.

(d) Finial one-irregular mesh.

Figure 4.6: Subsequent refinement of mesh to maintain a one-irregular discretization following [Rachowicz et al., 1989].

Thus, the global load vector $\boldsymbol{F}$ is assembled from the local element force vectors $\boldsymbol{F}^K$ by the matrix vector multiplication

$$\boldsymbol{F} = \sum_{K \in \mathcal{T}_h} \boldsymbol{\pi}^K \boldsymbol{F}^K, \tag{4.17}$$

with $F_i = f(N_i)$ and $F_j^K = f(\xi_j^K)$. A similar transformation is applied to the bi-linear form:

$$a(N_i, N_j) = \sum_{K \in \mathcal{T}_h} a(\pi_{ik}^K \xi_k^K, \pi_{jl}^K \xi_l^K) = \sum_{K \in \mathcal{T}_h} \pi_{ik}^K a(\xi_k^K, \xi_l^K) \pi_{jl}^K \tag{4.18}$$

Therefore, the global system matrix $\boldsymbol{A}$ is assembled from the local element matrices $\boldsymbol{A}^K$ by the following matrix multiplications:

$$\boldsymbol{A} = \sum_{K \in \mathcal{T}_h} \boldsymbol{\pi}^K \boldsymbol{A}^K \left(\boldsymbol{\pi}^K\right)^\top, \tag{4.19}$$

with $A_{ij} = a(N_i, N_j)$ and $A_{kl}^K = a(\xi_k^K, \xi_l^K)$.

4.2.3 Algorithmic limits

The described approach of a constrained approximation has been applied successfully for generating a continuous *hp*-discretization on irregular mesh in e.g. [Ainsworth and Oden, 1997; Ainsworth and Senior, 1999; Bangerth et al., 2007; Demkowicz, 2007; Demkowicz et al., 1999, 1998, 1989; Frauenfelder, 2004; Frauenfelder and Lage, 2002; Paszyński and Demkowicz, 2006; Rachowicz and Demkowicz, 2000a, 2002; Rachowicz et al., 1989; Šolín, 2004; Šolín et al., 2008, 2010a]. However, the implementation of the procedure sketched above comes with a high algorithmic complexity [Šolín et al., 2008].

This holds in particular if the constraint coefficients have to be derived in terms of already constrained nodes. In this case, the global basis function N_i cannot be expressed in terms of *one* element shape function as in (4.5). Accordingly, the simplification of left hand side of the

constraint equation is not possible in the way described above. Instead, the constraints have to be expressed over different levels, see e.g. [Ainsworth and Oden, 1997].

For this reason, most *hp*-frameworks do not allow for these multiply hanging nodes. Instead, only 1-irregular meshes are considered to reduce the implementational complexity [Bangerth et al., 2007; Demkowicz, 2007; Demkowicz et al., 1999, 1998, 1989; Paszyński and Demkowicz, 2006; Rachowicz and Demkowicz, 2000a, 2002; Rachowicz et al., 1989; Šolín, 2004]. To this end, also neighboring elements are refined if multiply hanging nodes emerge. However, when following this approach the refinement partially looses its local character as illustrated in Figure 4.6. In the depicted pathological case, twelve additional elements have to be introduced to maintain the one-irregularity rule. Numerical studies presented in [Schröder, 2011; Šolín et al., 2010a] demonstrate that these additional refinements introduce unnecessary degrees of freedom and therefore decrease the convergence rates.

For this reason, different formulations have been derived that allow for arbitrary hanging nodes in e.g. [Ainsworth and Oden, 1997; Ainsworth and Senior, 1999; Frauenfelder, 2004; Frauenfelder and Lage, 2002; Kus, 2011; Schröder, 2011; Šolín et al., 2008, 2010a]. However, the initial refine-by-replacement idea remained unchanged. Therefore, these approaches demand for a sophisticated discretization kernel.

An alternative to the complex refine-by-replacement strategies is to refine by *superposition*. The use of this idea for the construction of geometrically graded *hp*-meshes is addressed in the next chapter.

Chapter 5

Multi-Level construction of *hp*-meshes: refinement by superposition[a,b,c]

As outlined in the previous chapter, the classical idea of an *hp*-refinement is to increase the approximation accuracy locally by *replacing* coarse elements by a set of finer elements. An alternative approach—followed in this work—is to refine a global *base* mesh solution $\boldsymbol{u}_b$ by locally *superposing* a fine-scale *overlay* solution $\boldsymbol{u}_o$. The final approximation $\boldsymbol{u}$ is then given by the sum of the two components

$$\boldsymbol{u} = \boldsymbol{u}_b + \boldsymbol{u}_o. \tag{5.1}$$

This hierarchical refinement strategy is illustrated in Figure 5.1.

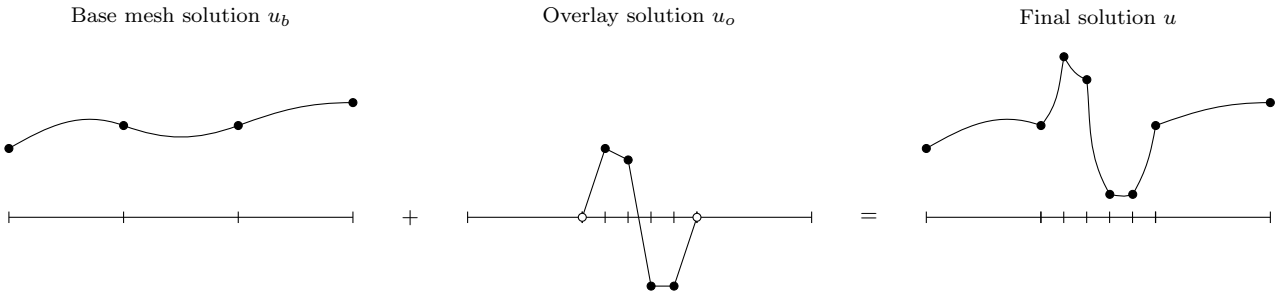


Figure 5.1: Conceptual idea of the refinement by superposition following [Rank, 1992].

The key point of this superposition approach is that the overlay solution $\boldsymbol{u}_o$ is constrained by homogeneous Dirichlet boundary conditions. Accordingly, $\boldsymbol{u}_o$ is zero on the transition of the refinement zone, which ensures the C^0-continuity of the final approximation. This simple way of ensuring the compatibility of the basis function offers an essential advantage compared to the aforementioned refine-by-replacement strategy, as no complex constraining of hanging nodes is necessary.

The current work proposes to use this alternative refinement concept to construct hierarchical *hp*-meshes that are free of hanging nodes but offer the same approximation accuracy as conventional discretizations. Before outlining the details of this approach in Section 5.2, the chapter begins with a brief overview of other FEM-based approaches that have applied this refinement concept before.

5.1 A brief history of superposition-based refinement methods

Global-local Finite Elements

To the author's knowledge, the idea of a refinement-by-superposition dates back to the pioneering work of Mote [1971]. To reduce the computational cost of numerical simulations, Mote suggests to combine the Ritz- and the conventional version of the Finite Element Method in a superposition approach. The advantage offered by the Ritz method is that it allows for the use of special solution functions that are known to capture the global deformation of the structure. In the vicinity of local features such as cracks or holes, this global solution is enriched by superposing a conventional finite element overlay mesh. Setting the cornerstone of the refinement-by-superposition idea, the final numerical solution is given by the sum of the base and overlay solution, and the global continuity is ensured by applying homogeneous Dirichlet boundary conditions on the overlay mesh. In his original work, Mote demonstrates the applicability of this approach in the context of beams and plane stress problems. In [Noor, 1986], the approach is compared to other global-local methods.

Hierarchical finite elements

In [Bank et al., 1988; Mitchell, 1988, 1991, 1992; Yserentant, 1986a,b], the refine-by-superposition concept is applied to perform an h-refinement of triangular finite element meshes. To this end, the standard nodal basis functions are replaced by a set of hierarchical basis functions. Accordingly, a k-times refined finite element space $\mathcal{V}_k$ is given by the direct sum of a base space $\mathcal{V}_0$ corresponding to an initial coarse base mesh and k overlayed spaces $\mathcal{O}_1, \ldots, \mathcal{O}_k$, i.e.

$$\mathcal{V}_k = \mathcal{V}_{k-1} \oplus \mathcal{O}_k, \quad \forall\, k > 1. \tag{5.2}$$

The spaces $\mathcal{O}_k$ are spanned by the shape functions of finer elements.

Following this construction approach, not only the sequence of the functions spaces $\{\mathcal{V}_k\}$ but also their basis functions are hierarchical. As shown in [Yserentant, 1986a,b] for triangular elements, this hierarchical structure has a beneficial effect on the condition number of the final equation system. Further, it allows for an efficient solution by multi-grid methods as shown in [Bank et al., 1988; Mitchell, 1988, 1991, 1992; Yserentant, 1986a,b]. In [Zienkiewicz and Craig, 1986], this idea is applied for p-refinement using hierarchical high-order shape functions.

Adaptive local overlapping grid method

In the adaptive local overlapping grid method—suggested in [Moore and Flaherty, 1992]—the governing equation is first solved on a coarse base mesh with bi-linear shape functions. Based on this solution, an error indicator is used to identify regions with a bad approximation quality. The discretization in these parts of the domain is h-refined by superposing finer overlay meshes. The refined regions are grouped to form rectangular "megagrids" on which the governing equations is solved again.

In [Moore and Flaherty, 1992], this spatial discretization is combined with a backward Euler time stepping scheme to simulate parabolic problems. Further, the discretization is adapted in every time step to capture propagating solution features such as flame fronts at high accuracy.

Spectral overlay method

The aim of the spectral overlay method introduced in [Belytschko et al., 1990] is to increase the approximation accuracy for solutions with high gradients. To this end, a standard finite element mesh is overlayed by a special, high-order overlay mesh—a so-called spectral overlay mesh—that resolves these local features. As in the previous approaches, the global continuity of the final approximation is ensured by applying homogeneous Dirichlet boundary conditions on the spectral overlay mesh.

In the original work of Belytschko et al. [1990], this idea is used to resolve local shear bands occurring in bifurcation problems. To this end, *a priori* knowledge of the local strain field is exploited by incorporating the results of a local bifurcation analysis in the definition of the overlay functions. In [Belytschko and Lu, 1992], this idea is extended to curvilinear elements. In [Gu and Belytschko, 1994], the approach is applied to determine stress singularities at bi-material wedges.

The *s*-version of the Finite Element Method

The superposition-version (*s*-version) of the Finite Element Method—introduced in [Fish, 1992b; Kim et al., 1991]—aims at generalizing the idea of the spectral overlay method to provide a general purpose refinement tool. To this end, elements with steep gradients in the solution are overlayed by a patch of high-order elements. In this way, the overlay mesh can accurately resolve fine-scale solution features. Possible linear dependencies between the shape functions of the two mesh levels are captured by eliminating equations with zero pivots occurring during the factorization process of the direct solver. Further, a complex integration mesh is used that resolves all intersecting element boundaries in both mesh levels.

The results presented in [Fish, 1992b] show that the use of a high-order overlay mesh improves the convergence also in the presence of singularities. In [Fish and Markolefas, 1993], this approach is coupled with an error estimation scheme. In [Fish, 1992a], the method is applied to crack problems using special singular overlay elements at the crack tip. The method is further extended in [Fish et al., 1994] using multiple levels of overlay meshes. In [Fish and Guttal, 1996], the *s*-version is used to model laminated shells. To this end, the global model is discretized using Equivalent Single Layer shell elements. In certain regions of interest, the approximation accuracy is increased by superposing a stack of three-dimensional elements.

Partition of Unity Methods

Another class of approaches following the refine-by-replacement idea are Partition of Unity Methods (PUM, [Babuška and Melenk, 1997; Melenk and Babuška, 1996]). The most successful approaches of this kind are the Generalized Finite Element Method (GFEM, [Strouboulis et al., 2000a,b, 2001]) and the eXtended Finite Element Method (XFEM, [Moës et al., 1999]). A comprehensive review of these methods is given in e.g. [Belytschko et al., 2009; Fries and Belytschko, 2010].

As phrased by Babuška and Melenk [1997], the objective of these discretization approaches is to permit the "inclusion of *a priori* knowledge about the differential equation in the ansatz spaces". Common applications where such *a priori* knowledge is available include immersed interfaces, evolving discontinuities, or crack tips. To incorporate this knowledge, the standard finite element function space is *extended* or *enriched* by special solutions that are tailored for the respective situation under consideration. In analogy to the other aforementioned refine-by-superposition methods, the final numerical approximation is given by the sum of the base mesh solution and an overlay function of choice $\psi(\boldsymbol{x})$

$$u(\boldsymbol{x}) = \sum_i N_i(\boldsymbol{x})\hat{u}_i + \sum_i N_i(\boldsymbol{x})\psi(\boldsymbol{x})\hat{a}_i. \tag{5.3}$$

Here, the coefficients $\hat{u}_i$ and $\hat{a}_i$ denote the degrees of freedoms of the base- and the overlay/enrichment solution, respectively. The above enrichment assumes that the standard nodal shape functions N_i build a partition of unity (PU)

$$\sum_i N_i(\boldsymbol{x}) = 1 \quad \forall\, \boldsymbol{x} \in \Omega. \tag{5.4}$$

This ensures that the target function ψ is contained in the finite element solution space.

The product of $\psi(\boldsymbol{x})$ and N_i can be seen as a new kind of local shape function whose support is restricted to the support of N_i. This allows for an easy implementation of the method as it only requires a change of shape functions while the degree of freedom handling remains unchanged.

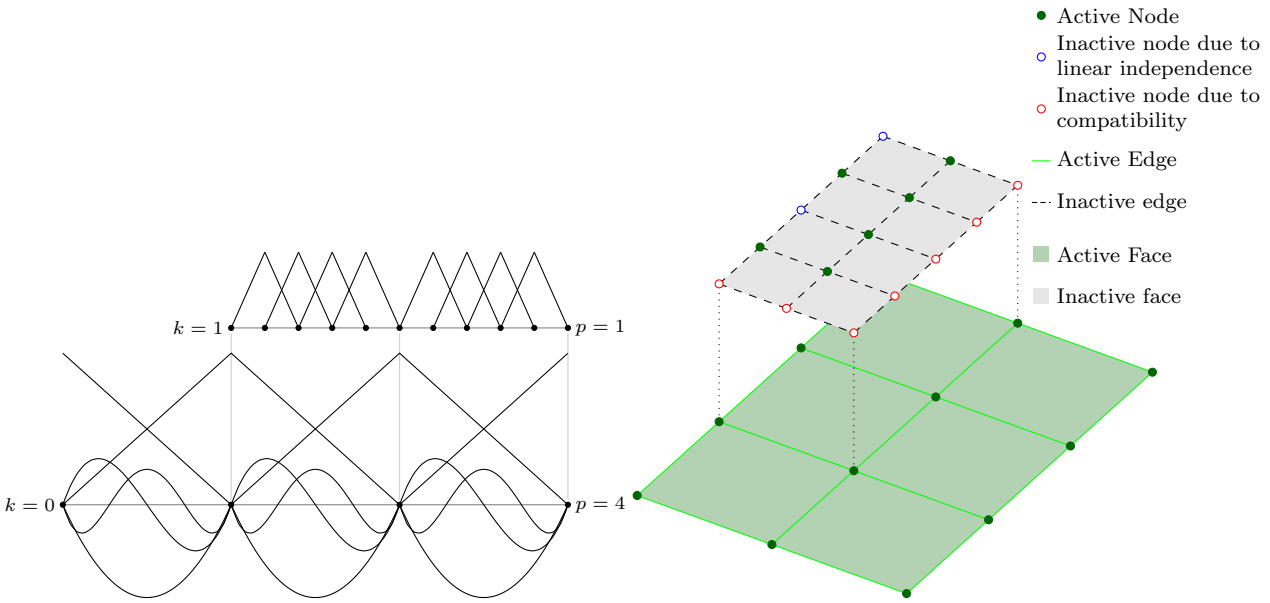
This enrichment concept has been a great success and the applicability was proven for various problems, including e.g. crack growth [Belytschko and Black, 1999; Loehnert et al., 2011; Moës and Belytschko, 2002; Moës et al., 1999], local refinement [Fries, 2008], embedded discontinuities [Cheng and Fries, 2009], solidification [Chessa et al., 2002], fluid structure interaction [Gerstenberger and Wall, 2008], and many more. The major challenges of this method include the correct integration of the possible complex enrichment function ψ. Further, possible (near) linear dependencies between base and the enrichment solution may result in a high condition number of the FE-equation system.

The *hp-d* version of the Finite Element Method

The final refine-by-superposition method to be mentioned in this review is the *hp-d* version of the Finite Element Method, which was introduced in [Rank, 1992]. The essential idea of this approach is that the analytical solution has a smooth character on a global scale, while non-smooth solution features are confined in small, local sub-regions. To resolve the smooth, large-scale part of the solution, the domain Ω is first discretized with a coarse *high-order* base-mesh. This base-mesh is superposed locally with a finer, low-order overlay mesh to also capture non-smooth solution features. This idea is illustrated in Figure 5.2.

As for the aforementioned refine-by-superposition approaches, the global continuity is ensured by applying homogeneous Dirichlet boundary conditions on the overlay solution. In the present case, this is reflected by deactivating all nodes on the boundary of the overlay mesh. In Figure 5.2b, these nodes are marked in red color.

The second important aspect to be considered is that the overlay mesh must not introduce a linear dependence between the shape functions of the two refinement levels. This is ensured by



(a) 1D-Case (b) 2D-Case

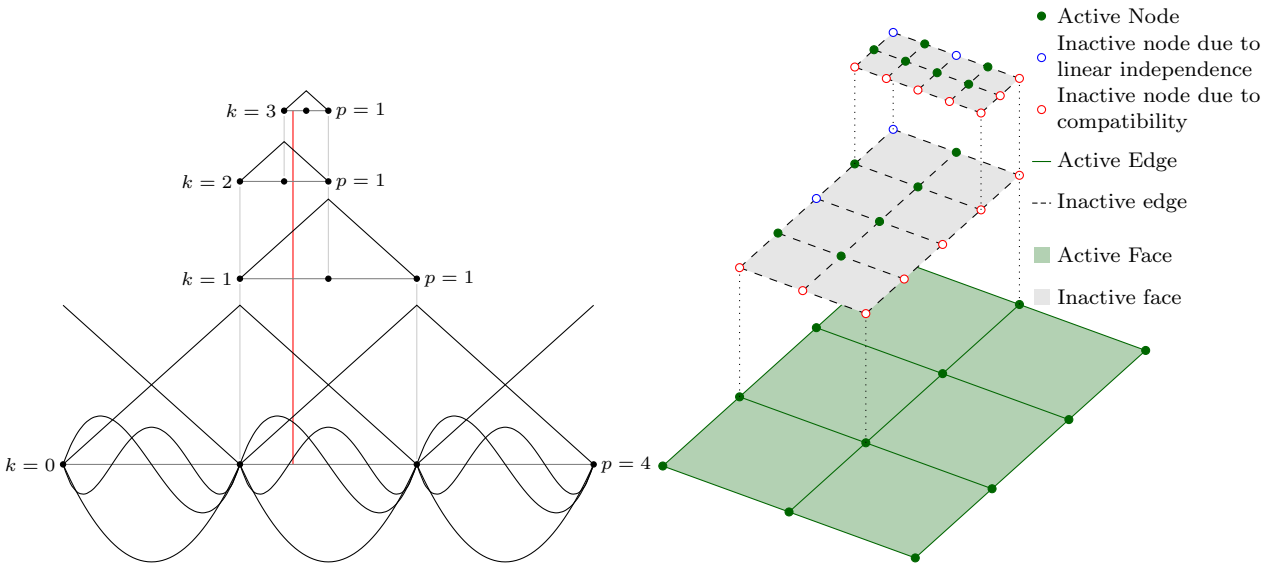
Figure 5.2: The *hp-d*-approach introduced in [Rank, 1992; Rank and Krause, 1997]: the coarse high-order base mesh captures the large-scale solution, while fine-scale characteristics are captured using the finer *h*-mesh that overlays the base mesh in the respective domain of interest.

first restricting the refinement to a structured superposition in which one fine-scale element can only overlay *one* base element. In this setting of structured superposition, linear dependencies can be removed by deactivating those overlay nodes that are direct descendants of base mesh nodes. In Figure 5.2b, these are marked in blue color.

It is worth noticing that, in principle, the linear dependence could also be ensured by activating the overlay nodes instead of the base mesh nodes. In this way, the support of the shape functions would be reduced. However, the chosen strategy of deactivating the overlay nodes offers the advantage that the base mesh remains unchanged. In this way, the basis functions of the finite element space are fully hierarchical as only new shape functions are added.

This non-intrusive nature of the method can be exploited by solving the coupled problem in a partitioned approach. To this end, the two refinement levels are discretized by two different finite element solvers. The inter-level coupling is handled as an additional volume load. In [Rank, 1992], this approach is used to simulate reaction-diffusion processes occurring in semi-conductors. In [Krause and Rank, 2003; Rank and Krause, 1997], the idea is used to linear elastic problems. The high flexibility of the method is used in [Düster et al., 2007a; Ledentsov, 2010; Ledentsov et al., 2010] to "locally enhance dimensionally reduced models" by superposing volumetric models in domains of interest.

In [Schillinger et al., 2012b], the *hp-d*-idea is extended to aggregate the fine scale shape functions in the direct vicinity of local features. To this end, the base element incorporating this feature is refined by superposing finer elements in the next refinement level. Performing



(a) 1D-case, the red line marks the position of a singularity.

(b) 2D-case

Figure 5.3: Conceptional idea of the hierarchical *hp-d*-method following e.g. [Schillinger et al., 2012b].

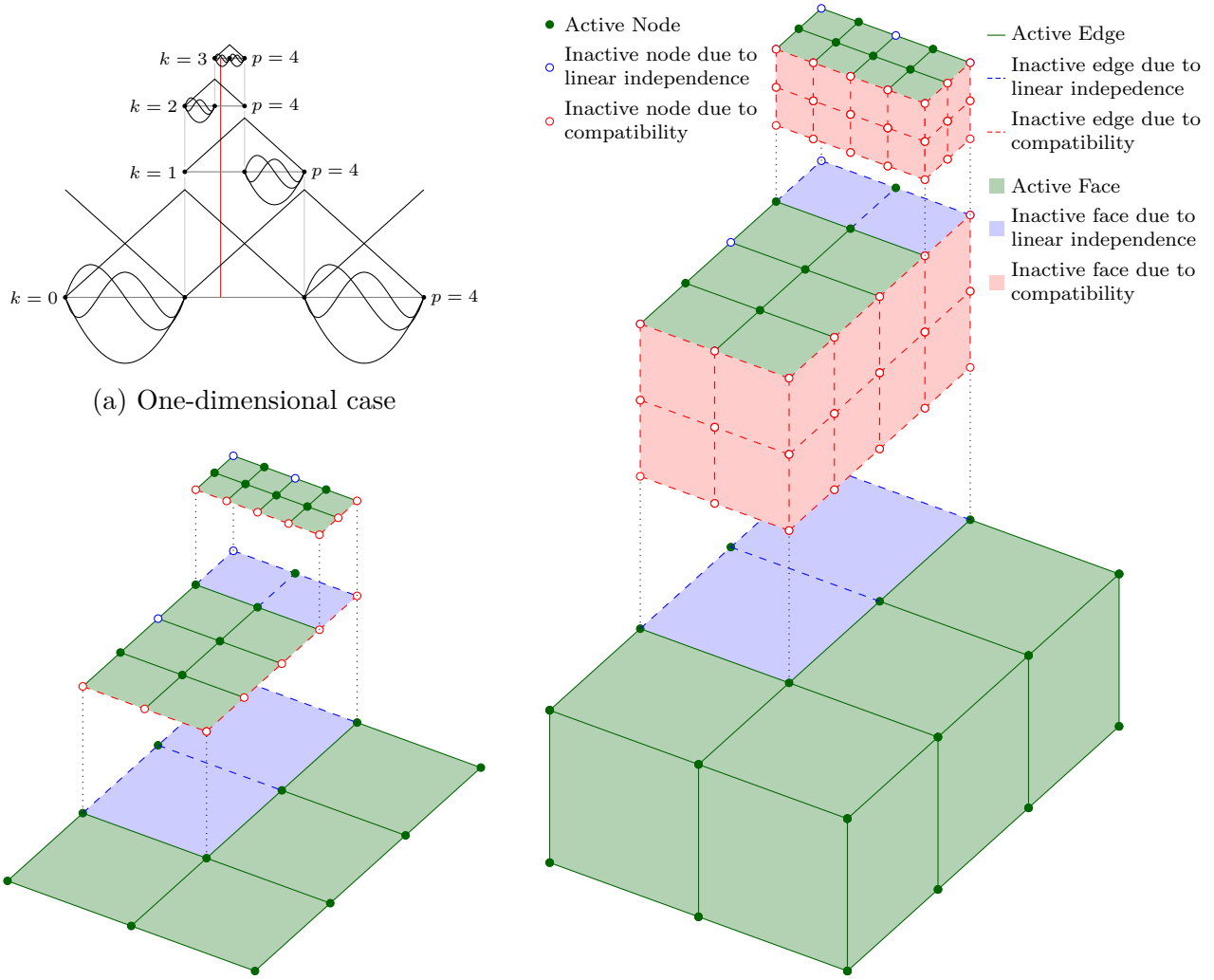
this process recursively gives rise to a refinement *tree*, in which the base element represents the root, and the sub-elements are the children. The last level of sub-elements is denoted as leaf-elements. Following the classical *hp-d*-idea, the global continuity is ensured by deactivating the nodes on the boundary of the overlay mesh, and a linear dependence is avoided by deactivating direct descendants of coarse nodes. This idea is illustrated in Figure 5.3.

Schillinger et al. [2012b] show that this hierarchical extension yields excellent results in the context of geometrically non-linear structural mechanics and fictitious domain problems. In [Schillinger, 2012; Schillinger et al., 2012a; Schillinger and Rank, 2011], this idea is used in the context of isogeometric analysis (IGA) to adaptively refine B-spline and NURBS elements. In [Schillinger et al., 2013], it is shown that this approach also yields excellent results in the framework of collocation methods[f].

Although the results presented in [Düster et al., 2007a; Krause and Rank, 2003; Ledentsov, 2010; Ledentsov et al., 2010; Rank, 1992; Rank and Krause, 1997; Schillinger et al., 2012b] show significant improvements of the approximation accuracy, the overlay only employs linear shape functions. Thus, the approximation quality is limited to an algebraic convergence in case of non-analytical solutions. The multi-level *hp*-method extends the *hp-d*-idea to yield exponential convergence also in the presence of singularities. The details of this method are described in the upcoming section.

5.2 The multi-level *hp*-method

As noted in the introduction to this section, the multi-level *hp*-version of the Finite Element Method aims at avoiding the problem of hanging nodes common in irregular *hp*-meshes,



(a) One-dimensional case

(b) Two-dimensional case

(c) Three-dimensional case

Figure 5.4: Conceptual idea of the multi-level *hp*-method [Zander et al., 2016].

while achieving the same approximation accuracy as conventional *hp*-formulations. To this end, the classical refine-by-replacement paradigm is replaced by the aforementioned refine-by-superposition approach.

As illustrated in Figure 5.4, the multi-level *hp*-method follows the idea of the *hp-d*-approach: the approximation quality is increased in the vicinity of a singularity by superposing the respective base element by finer overlay elements in the next refinement level. This process is repeated recursively giving rise to a refinement tree, in which the root is given by the base element, and the children being the sub-elements. The elements on the last refinement level represent the leaves of the refinement tree.

To construct an *hp*-discretization, the multi-level *hp*-approach uses *high-order* overlay meshes for the refinement. Exploiting the clear hierarchical structure in the overlay elements and the integrated Legendre shape functions, the following simple rule set is sufficient to ensure the compatibility and the linear dependence of the *hp*-basis.

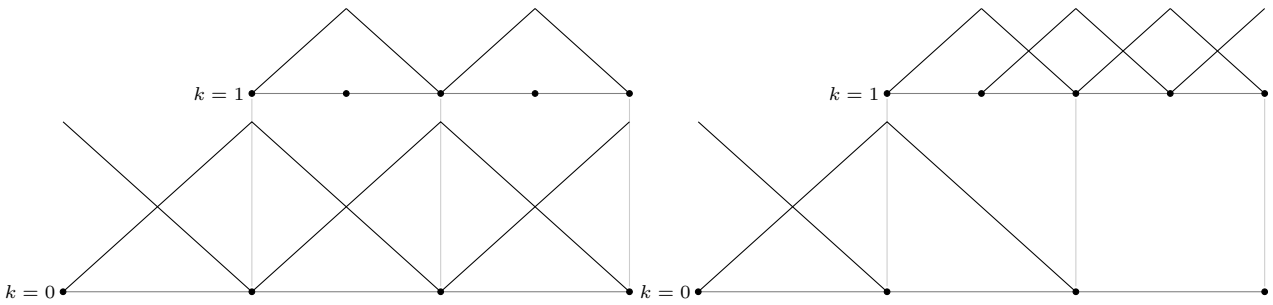
Compatibility

The global C^0-continuity of the shape functions—and with it the compatibility—is ensured by applying homogeneous Dirichlet boundary conditions on all overlay meshes. In the context of high-order overlay elements, this requires to remove all overlay shape functions from the FE-function space that are non-zero on the boundary of the refinement zone. As discussed in Section 3.2.1, the integrated Legendre shape functions can be classified as nodal, edge, face and volume modes. Accordingly, each shape function can be associated directly to one topological component. Using this inherent structure, the compatibility of the basis can be ensured by "deactivating" all nodes, edges, and in 3D faces on the boundaries of the overlay meshes. In this way, all "hanging" nodes are removed from the discretization, and the overlay solution is zero at the transition from the refined to the coarse domain. This ensures the global continuity of the solution. In Figure 5.4, these components are marked in red color. The illustrations demonstrates that this rule allows for arbitrary irregular meshes as the overlay discretizations can be arranged such that the respective mesh boundaries coincide.

Linear independence

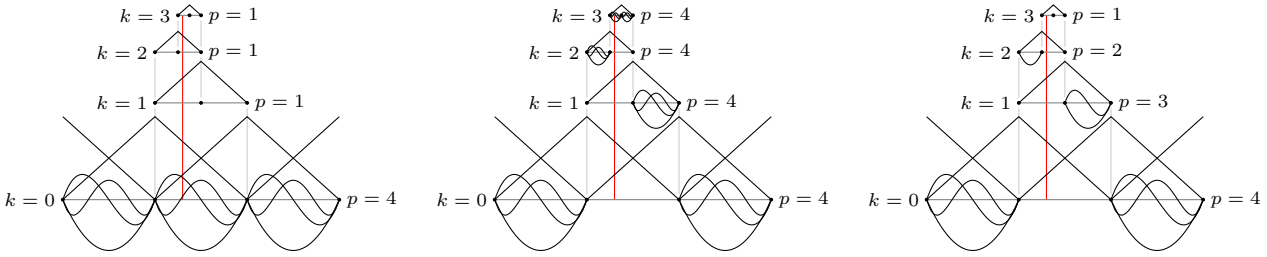
The second aspect to be considered are possible linear dependencies between shape functions on different refinement levels. Due to the use of high-order overlay meshes, this concerns the linear *and* the high-order shape functions.

To prevent linear dependencies between the nodal shape functions, only one of two nodes in a direct descendant relation (parent- and sub-node) may contribute to the final FE-space. Accordingly, either the sub-node of a parent or the parent of a sub-node has to be deactivated. The two approaches are compared in Figure 5.5. In both cases, the resulting function spaces are the same. When deactivating the sub-node of a parent, the linear modes are distributed over the different refinement levels. This allows for a smoother transition of basis functions in the case of mesh-coarsening during the runtime of the simulation. On the other hand, deactivating the parent of a sub-node reduce the support of the nodal basis functions and with it the numerical coupling between the levels. For historic reasons, the former approach is chosen in this work and linear dependencies between nodal modes are prevented by deactivating the overlay nodes that are direct descendants of coarse-scale nodes.



(a) Sub-node deactivation (b) Parent-node deactivation

Figure 5.5: Prevention of linear dependencies of nodal modes: from two nodes in a direct descendant relation (parent- and sub-node), one has to be deactivated.



(a) Linear overlay, *hp-d*-approach (b) Uniform multi-level *hp*-approach (c) Non-uniform multi-level *hp*-approach

Figure 5.6: Comparison of different approaches for hierarchical, high-order refinement.

To prevent linear dependencies between high-order modes, the hierarchical nature of the integrated Legendre shape functions is exploited. The linear independence is ensured by allowing each high-order mode to be present only once in each branch of the refinement tree. In Figure 5.4, the high-order modes are placed on the leaves of the refinement tree. On the coarser elements, only the linear shape functions remain active to ensure the global continuity. The modal concept of the integrated Legendre shape functions allows for a simple implementation of this rule as the redundant high-order shape functions can be removed easily by deactivating edges, faces, and solids that have active children. In Figure 5.4, these are marked in blue color. An alternative approach is to place the high-order shape functions on the base element. In this configuration, the multi-level *hp*-approach is identical to the *hp-d*-refinement strategy as illustrated in Figure 5.6a. The polynomial degree can also be adjusted non-uniformly within the mesh by setting the polynomial degree of the leaf-elements accordingly. This idea is depicted in Figure 5.6c.

This comparison shows that the idea to refine via superposition allows the formulation of a *hp*-discretization scheme that features the same flexibility as conventional approaches but avoids the problems of hanging nodes. Accordingly, the compatibility and linear independence of the basis functions can be ensured by a simple rule-set, which is directly applicable for one-, two- and three-dimensional refinement of arbitrary irregularity.

Although the described *hp*-formulation is easy to implement, the simplicity might imply a potential performance bottleneck: as the approximation accuracy is increased by superposing different elements, the number of shape functions with overlapping support increases. This results in a denser system of equations with a higher bandwidth compared to a conventional *hp*-scheme. Furthermore, the superposition leads to more shape functions being non-zero on each integration point. This might have negative influences on the time required for integrating the element matrices and for solving the system of equations.[f]

To quantify this aspect, the computational performance of the suggested hierarchical refinement approach is analyzed for several two- and three-dimensional benchmarks in Section 8. In this context it is noteworthy that the number of non-zero shape functions at each integration point is the same for both, the *hp-d-* and the multi-level *hp*-approach. Hence, the work required for computing the stiffness matrix for one integration domain is equivalent for both

methods. Therefore, it is to be expected that the latter refinement strategy performs better not only in terms of the number of unknowns but also when comparing the accuracy increase with respect to the run-time.

Chapter 6

Implementational aspects of multi-level *hp*-refinement[a,b,c]

As discussed in Section 5.2, the main challenge in the multi-level *hp*-approach is to ensure linear independence and compatibility of the basis functions. This is achieved by deactivating the "correct" topological components. The following chapter describes the implementation of the respective rule-set introduced in Section 5.2 in the context of an abstract object-oriented FEM-code. Thereby, only quadrilateral topologies are considered to limit the extend of the discussion. However, the handling of other topologies—such as triangles, hexahedrals, tetrahedral or prisms—follows in analogy. The outline is based on the description given in [Zander et al., 2016, 2015, 2017].

6.1 Data structure[h]

Within this work, the connectivity of the finite element mesh is formulated in terms of three types of topological components: the nodes, the edges, and the faces. In the context of an object oriented framework, each of these component types is represented by one corresponding class. As depicted in Figure 6.1, an edge consists of a list of nodes (typically two), and a faces consists of a list of nodes and a list of edges. To represent the hierarchical refinement tree mentioned in Section 5.2, each topological component additionally stores a reference to its sub-components that are created in the refinement process explained in Section 6.2. In the case of a node, this is one sub-node, whereas edges and faces store a list of sub-edges and sub-faces, respectively.

To treat the topological components in a uniform way, the three classes inherit from a common abstract base class `AbsTopologicalComponent`. In this way, large parts of the implementation can be re-used by placing it in the base class. In particular, this applies to the degree of freedom handling. As outlined in the previous section, the shape functions and corresponding degrees of freedom are identified as nodal, edge, or face modes. For this reason, each topological component stores a polynomial degree that specifies the order of the associated shape functions. Furthermore, the topological components are responsible for handling the degrees of freedom. To this end, the components store a list of degrees of freedom. Given that an `isActive` flag is set, this list is filled during the setup phase of the program taking

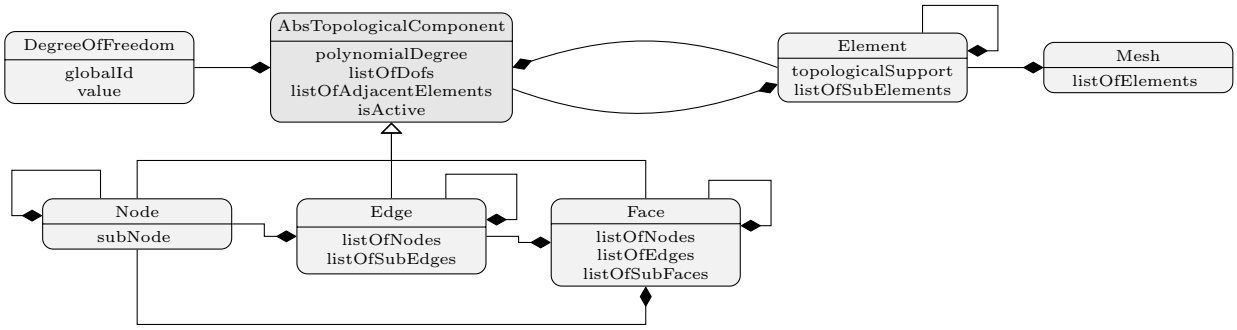


Figure 6.1: UML representation of the multi-level *hp*-data structure [Zander et al., 2017].

into account the associated polynomial degree.

Using the described topological setup, a finite element can be defined by specifying one topological component as its *support*. In the one-dimensional case, the elements are defined on edges, whereas in two dimensions, a face serves as the support. As depicted in Figure 6.1, each element stores a reference to the respective topological support, and vice versa each topological component holds a list of all adjacent elements. Using the topological support, the element can build up the location map and construct the element shape functions depending on the polynomial degree specified by the topological support. Additionally, each element holds a list of sub-elements, which overlay the respective element in the next refinement level. This allows the refinement tree to be build up recursively starting from the coarse base elements.

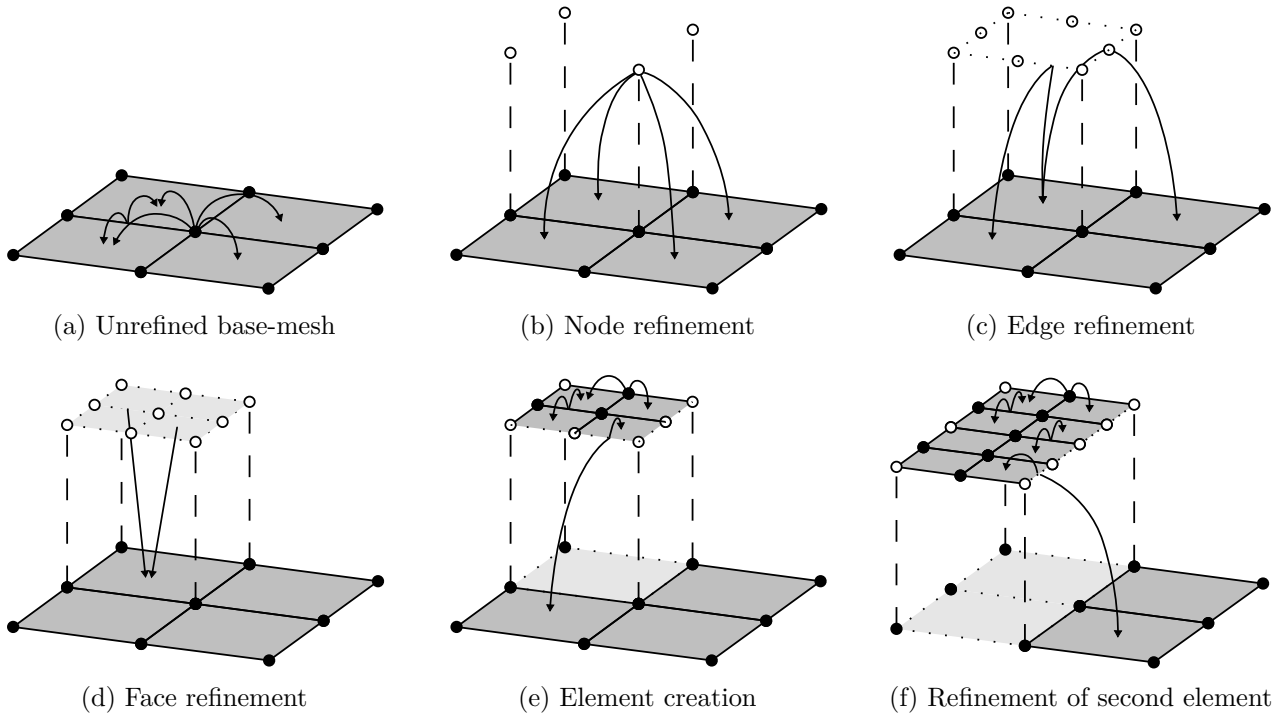
The list of all base elements is stored in the mesh class. The main responsibility of the mesh class is to evaluate the element shape functions at the quadrature points. In this way, the element stiffness matrices can be integrated and then assembled into the final system matrix.

6.2 Refinement procedure[g,h]

Based on the data-structure introduced in the previous section, the refinement of the discretization requires to subdivide the respective topological components such that new elements can then be defined on this sub-topology. To start the refinement procedure, an element of the base mesh is passed to the function `refineElement` outlined in Algorithm 2. The function first checks, whether the passed element is marked for refinement. In this case, the new sub-elements are created by calling the function `createSubElements` on the passed element *cf.* Algorithm 3. Afterwards, the `refineElement` function is called recursively on each of the created sub-elements. In this way, the function-call traverses naturally through the complete refinement tree in a depth-first pattern.

The process of creating the sub-elements is illustrated in Algorithm 3. Here, the first step is to create the new sub-topology. To this end, the topological support of the element is passed to the function `createSubTopology`. The creation of the overlaying sub-topology is depicted in Figure 6.2. In the figures, the arrows exemplarily illustrate the list of adjacent elements for selected topological components.

As depicted in Figure 6.2b, the first step of refining the topological support of the element (in the present example a quadrilateral face) is to refine the nodes of the topological component. To this end, a new sub-node is added to each node of the topological support. These sub-nodes



(a) Unrefined base-mesh (b) Node refinement (c) Edge refinement

(d) Face refinement (e) Element creation (f) Refinement of second element

Figure 6.2: Creation of overlaying sub-topologies and sub-elements following [Zander et al., 2015]. The arrows depict the adjacency relation between the created topological sub-components and the elements [Zander et al., 2017].

Algorithm 2 Recursive refinement of an element [Zander et al., 2017]

```
1  function refineElement( elementToBeRefined )
2  {
3    // check whether the passed element shall be refined
4    if( isElementSelectedForRefinement( elementToBeRefined ) == true )
5    {
6      // create new sub-elements -> see Algorithm 3
7      createSubElements( elementToBeRefined )
8
9      // traverse through the refinement-tree by recursively
10     // calling the refinement function on each sub-element
11     for each subElement in elementToBeRefined->listOfSubElements
12     {
13       refineElement( subElement )
14     }
15   }
16   // check whether the passed element shall be coarsened
17   else if( isElementSelectedForCoarsening( elementToBeRefined ) == true )
18   {
19     // remove the sub-elements from the current element -> see Algorithm 4
20     removeSubElements( elementToBeRefined )
21   }
22 }
```

Algorithm 3 Creation of sub-elements [Zander et al., 2017]

```
1  function createSubElements( elementToBeRefined )
2  {
3    // first, refine the topological support of the element ...
4    topologicalSupport = elementToBeRefined->topologicalSupport
5    createSubTopology( topologicalSupport )

7    // ... and then create new elements on each of the sub-topologies
8    for each subTopology in topologicalSupport->getListOfSubTopologies
9    {
10     // first, create the new sub-element on the sub-topology ...
11     subElement = createElement( subTopology )

13     // ... and add new element to the list of sub elements of the parent
14     elementToBeRefined->addSubElement( subElement )

16     // then de-register the parent element from the adjacency
17     // list of the sub-topology ...
18     subTopology->deregisterAdjacentElement( elementToBeRefined )

20     // ... and register the sub-element instead
21     subTopology->registerAdjacentElement( subElement )
22   }
23 }
```

are geometrically identical to their parents and thus overlay them in the next refinement level. Furthermore, the sub-nodes inherit the list of adjacent elements from their parents. Therefore, at this point the sub-nodes are still considered to be adjacent to the original coarse elements. In Figure 6.2b, this is illustrated for the central sub-node.

In analogy to the nodes, the edges and the face itself are refined by creating new topological sub-components that overlay their parents in the next refinement level (see Figures 6.2c and d). Again, these new topological components inherit the adjacency list from their parents and therefore are still regarded as being connected to the coarse base elements.

An important decision at this point is whether to define the sub-topologies geometrically in the physical space or in the reference/index space of their parent. Defining the sub-topologies in the physical space offers the advantage that, in addition to the refinement of the actual solution space, also the details of the geometric model can be enhanced. However, if a refined geometric model is not available, the definition of the sub-components in the index space of the respective parent offers the advantage that the refined topology naturally follows the geometry of the possibly curved original component.

Without loss of generality, the latter approach is chosen in this work. As illustrated in Figure 6.3, the mapping between the local space $\hat{\Omega}_k$ of a sub-component on refinement level k and the global space Ω is a concatenated sequence of mappings:

$$\Psi = \Psi_0 \circ \Psi_1 \circ \ldots \circ \Psi_{k-1} \circ \Psi_k. \tag{6.1}$$

This sequence of mappings has to be taken into account when computing the derivatives of a function ϕ defined in $\hat{\Omega}_k$ with respect to the global coordinates $\boldsymbol{x}$. In case of the first derivative,

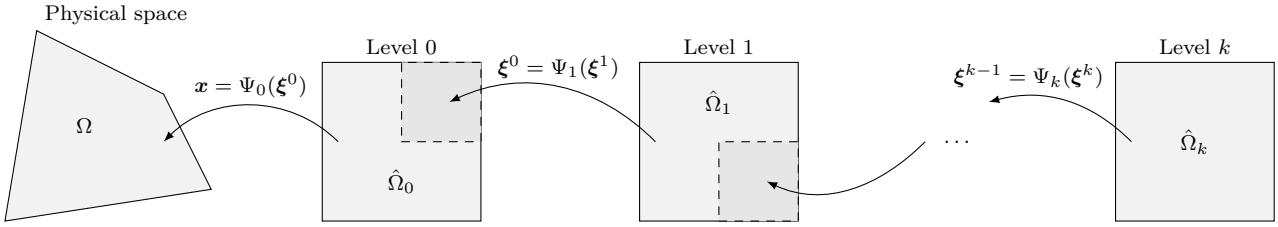


Figure 6.3: Concatenated geometry mapping: the sub-components are defined with respect to the index space of their parent topology [Zander et al., 2016].

the chain rule has to be applied as follows:

$$\frac{\partial \phi}{\partial x_i} = \frac{\partial \phi}{\partial \xi_j^k} \cdot \frac{\partial \xi_j^k}{\partial \xi_l^{k-1}} \cdot \ldots \cdot \frac{\partial \xi_n^1}{\partial \xi_m^0} \cdot \frac{\partial \xi_m^0}{\partial x_i}. \tag{6.2}$$

Here, $\boldsymbol{\xi^k}$ denote the index coordinates of the k^{th} refinement level. Using the inverse of the Jacobian matrix $\boldsymbol{J}_k$ of the mapping between the levels $k-1$ and k:

$$J_{ij}^k = \frac{\partial \xi_j^{k-1}}{\partial \xi_i^k} \quad \text{with} \quad J_{ij}^0 = \frac{\partial x_j}{\partial \xi_i^0}, \tag{6.3}$$

the above expression (6.2) can be re-written in matrix form as follows:

$$\nabla_{\boldsymbol{x}} \phi = \boldsymbol{J}_0^{-1} \cdot \boldsymbol{J}_1^{-1} \cdot \ldots \cdot \boldsymbol{J}_{k-1}^{-1} \cdot \boldsymbol{J}_k^{-1} \cdot \nabla_{\boldsymbol{\xi}^k} \phi. \tag{6.4}$$

The second derivative of the function ϕ defined in $\hat{\Omega}_k$ with respect to the next level $k-1$ is computed as:

$$\begin{aligned} \boldsymbol{H}^{k-1}(\phi) &= \boldsymbol{J}_k^{-\top} \cdot \left[\boldsymbol{H}^k(\phi) - \nabla_{\boldsymbol{\xi}^{k-1}} \phi \cdot \boldsymbol{H}^k(\boldsymbol{\xi}^{k-1}) \right] \cdot \boldsymbol{J}_k^{-1}, \\ \text{with} \quad H_{st}^k(\phi) &= \frac{\partial^2 \phi}{\partial \xi_s^k \, \partial \xi_t^k} \quad \text{and} \quad H_{rst}^k(\boldsymbol{\xi}^{k-1}) = \frac{\partial^2 \xi_r^{k-1}}{\partial \xi_s^k \, \partial \xi_t^k} \end{aligned} \tag{6.5}$$

With this setup, the derivative with respect to the global coordinates $\boldsymbol{x}$ can be computed recursively.

After the sub-topology has been created, new sub-elements can be created. To this end, Algorithm 3 loops over all the sub-components of the topological support. In the present case of a quadrilateral face, these sub-components are the four sub-faces. In the case of a hexahedral topological support, the sub-components would be eight octants. On each of these sub-components, a new element is created (Algorithm 3, line 11) and added as a sub-element to the parent element. It is important to remember that these sub-elements do not replace their parents but overlay them in the next refinement level. Therefore, the coarse base elements are still part of the discretization. However, the new sub-elements change the adjacency relation of the topological components. As indicated in the final part of Algorithm 3, the parent elements are de-registered from the adjacency list of the sub-topology and the new sub-elements are registered instead. Thereby, the functions `deregisterAdjacentElement` and `registerAdjacentElement` do not only affect the four sub-faces but also the connected nodes and edges. The updated adjacency relations are depicted Figure 6.2e.

Algorithm 4 Coarsening of refinement tree [Zander et al., 2017]

```
1  function removeSubElements( elementToBeCoarsend )
2  {
3    // first, coarsen all sub-elements of the sub-elements of the passed
4    // element To this end, traverse through the refinement-tree by
5    // recursively calling the coarsening function on each sub-element
6    for each subElement in elementToBeCoarsend->listOfSubElements
7    {
8      removeSubElements( subElement )
9    }

11   // then, de-register the direct sub-element of the passed element
12   // from the adjacency list of their topology
13   for each subElement in elementToBeCoarsend->listOfSubElements
14   {
15     deregisterElementFromTopologicalSupport( subElement )
16   }

18   // and finally delete the actual sub-elements
19   elementToBeCoarsend->listOfSubElements->clear()
20 }
```

If the next element of the base mesh is refined, the described procedure is repeated. Thereby, common topological components that have been refined before are not refined a second time. Instead, the previously created sub-components are used again. In this way, a topologically connected overlay mesh emerges. Also the adjacency relations are updated accordingly, as illustrated in Figure 6.2f.

In analogy to the refinement, the discretization can also be coarsened easily as illustrated in Algorithm 4. To this end, the depth of the refinement tree is decreased by recursively deleting the leaf-elements and updating the adjacency list of the topology.

6.3 Implementation of the rule-set for a linear-independent and compatible basis

As outlined in Chapter 5, the essential step in the multi-level *hp*-mesh construction is to ensure the compatibility and the linear independence of the shape functions. For this purpose, the "correct" topological components have to be deactivated.

Compatibility[h]

To ensure the compatibility, the rule set introduced in Section 5.2 requires to de-activate all topological components on the boundary of the refinement zone. Using the aforementioned data setup, these components can be identified easily by comparing the refinement level of all adjacent elements. This is done in the first part of the function `deactivateTopology` outlined in Algorithm 5, which is called on the topological support of each base element of the mesh. In the second part of the algorithm, the function is called recursively on all the connected

Algorithm 5 De-activation of topological components to ensure the compatibility and the linear independence of the basis functions [Zander et al., 2017]

```
function deactivateTopology( topologicalComponent )
{
  // check if the passed topology is on the boundary of the refinement
  // zone by running over each element adjacent to the component ...
  topologicalComponentIsOnBoundary = false
  for each adjacentElement in topologicalSupport->adjacentElements
  {
    // ... and comparing the refinement level of the adjacent element
    // to the refinement level of the topological component
    if( adjacentElement->refinementLevel !=
        topologicalComponent->refinementLevel )
    {
      topologicalComponentIsOnBoundary = true
      break
    }
  }
  // then, do the same for all connected nodes and edges by
  // calling deactivateTopology recursively on these components
  for each connectedComponent in
          topologicalComponent->getConnectedComponents()
  {
    deactivateTopology( connectedComponent )
  }
  // and then do the same for all the sub-components recursively
  for each subTopology in topologicalSupport->getListOfSubTopologies()
  {
    deactivateTopology( subTopology )
  }
  // and deactivate the passed component if it is either on the boundary
  // of the refinement zone or if its a leaf of the refinement tree
  if ( topologicalComponentIsOnBoundary == false &&
       hasChildren( topologicalComponent ) == false )
  {
    topologicalComponent->isActive = true
  }
  else
  {
    topologicalComponent->isActive = false
  }
  // finally, check whether the parent- or sub-nodes shall be activated...
  if ( activateBaseNodes == true &&
       isNode( topologicalComponent ) == true )
  {
    // ... and set the flag accordingly
    if ( topologicalComponentIsOnBoundary == false &&
         hasParent( topologicalComponent ) == false )
    {
      topologicalComponent->isActive = true
    }
    else
    {
      topologicalComponent->isActive = false
    }
  }
```

topological components. In the present case of a quadrilateral face, this concerns the four nodes and the four edges of the face. In the third step, the function `deactivateTopology` is again called recursively on all sub-topologies which were created during the refinement procedure. In this way, the function-call traverses naturally through the complete refinement tree and all topological components of the hierarchical mesh are reached. In line 31, Algorithm 5 checks whether the current component is on the boundary and sets the `isActive`-state accordingly.

Linear independence

As outlined in Section 5.2, the rule set for a linear independent basis can be formulated in two variants, which differ in the handling of the nodal modes. In both cases, edges, faces and solids are deactivated if the respective component has no children and is thus a leaf of the refinement tree. This check is performed in line 32 of Algorithm 5 and the `isActive`-state is set accordingly. In the first variant of the rule set, the same strategy applies to the nodes. Accordingly, no extension to the algorithm is necessary in this case. In the second variant, however, not the overlay nodes but the parent nodes are kept active. To reflect this variant, an extra check is added in line 41 of Algorithm 5.

6.4 Numerical integration and evaluation of shape functions

After the creation of the overlay mesh, the discrete bi-linear form and the right-hand-side vector have to be integrated. This is done using the standard Gaussian quadrature. However, the scheme has to be adapted to account for superposed elements as their shape functions are only C^0-continuous within the parent element. For this reason, the integration domain has to be partitioned such that standard Gaussian quadrature can be used in each sub-domain of the integral. As depicted in Figure 6.4, the coarsest partition of this kind is given by the leaf elements of the refinement tree. As within the individual integration domains all shape functions are polynomial, conventional Gaussian quadrature can be applied. It is worth noticing that this approach is only possible because each sub-element is fully contained within *one* parent element. Therefore, no complex subdivision of the integration domain is necessary. In approaches allowing for unstructured overlay meshes, the correct integration requires more complex subdivision solutions as discussed in e.g. [Fish et al., 1994].

The evaluation of the discrete bi-linear form at one of the resulting integration points translates into evaluating all non-zero shape functions in the complete hierarchy of superposed elements. These shape functions are considered as active in the following. It is essential to note that the set of active shape functions does change within each of the aforementioned integration domains. Hence, these domains can be considered as a conventional finite element equipped with a set of active shape functions. From this set, a location matrix can be derived, which allows to assemble the element stiffness matrix of the respective integration domain into the finial discrete system.

The evaluation of the active shape functions follows the recursive nature of refinement tree: first, the shape functions of the base elements are evaluated at the respective integration point. In a second step, this point is mapped into the index space of the next sub-element, and the evaluation function of the sub-element is called on the mapped point. In this recursive way, the

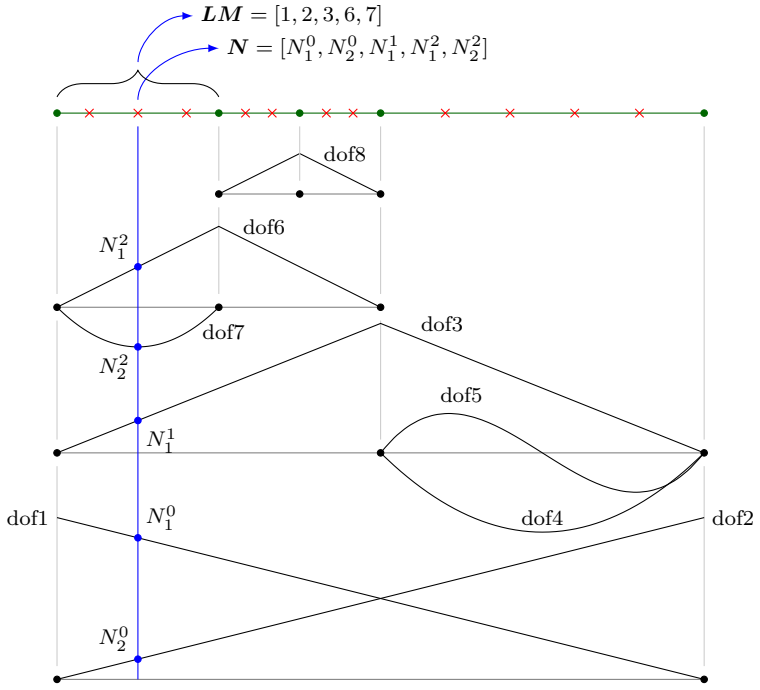


Figure 6.4: Integration on hierarchically refined mesh: the integration domain is partitioned according to the leaf elements. One each integration domain (green segments), standard Gaussian integration points are used for the numerical quadrature (red crosses). On each integration point, all non-zero shape function values (blue circles) present across the different mesh levels are concatenated into one shape function vector $\boldsymbol{N}$. In analogy, the location map $\boldsymbol{LM}$ of each integration domain is formed by concatenating the global dof ids of all shape functions that are non-zero within the respective integration domain.

program call naturally traverses through the refinement tree and all non-zero shape functions can be concatenated into one vector. With this active shape function set, the discrete bi-linear form can be evaluated and the stiffness matrix of the respective integration domain can be integrated.

Chapter 7

Error-controlled multi-level *hp*-adaptivity

The three previous chapters discussed the benefits an *hp*-mesh refinement and the simple realization of these irregular discretizations via the multi-level *hp*-overlay approach. An essential remaining aspect—which will be the focus of this chapter—is how to decide *where* and *how* to refine. Following e.g. [Babuška and Miller, 1984], the refinement process can be guided in an *a priori* or an *a posteriori* way.

In the case of an *a priori* refinement, the discretization is designed based on earlier or engineering experience. Possible scenarios in which this approach yields its full potential include stress concentrations in the vicinity of re-entrant corners, sharp material interfaces or sudden changes in the boundary conditions.

In an *a posteriori* scheme, a first finite element approximation is computed using an initial mesh. Carefully examining the resulting numerical solution, the initial mesh is refined, and a second approximation is computed. This iteration process is repeated until a pre-defined stopping criterion is reached. Since the mesh-design is optimized based on the previous solutions, Babuška *et al.* classify this process as a *feedback* technique for the construction of a sequence of finite element meshes $\{\mathcal{T}_h^i\}$ and the corresponding solutions $\{u_h^i\}$ [Babuška, 1986; Babuška and Miller, 1984, 1987]. Formally, this mesh update process is expressed as

$$\mathcal{T}_h^i = A\left(\mathcal{T}_h^1, \ldots, \mathcal{T}_h^{i-1}, u_h^1, \ldots, u_h^{i-1}\right), \tag{7.1}$$

where A denotes the *transition operator* used for the design of the new discretization.

Babuška [1986] classifies a transition operator A as *adaptive* with respect to the *convergence measure* if the sequence of numerical solutions converges against the analytical solution, i.e. $||u - u_h^i||_E \to 0$ as $i \to \infty$. If further, the convergence rate matches the *a priori* expectations, the operator is denoted as adaptive with respect to the *convergence rate measure*. The transition operator is classified as adaptive with respect to the *work measure* if the computational effort of creating the sequence of meshes and numerical solutions is bounded. If the feedback scheme is combined with an error estimator, the process is classified as adaptive with respect to an *effectivity index* if the estimated error converges against the true error as $i \to \infty$. A more detailed outline of these notions can be found in e.g. [Babuška, 1986, Section 1.5].

To design an automatic adaptive *hp*-scheme, the feedback algorithm has to decide *where* to refine and *how* to perform this refinement. Regarding the first question, it seems natural

to refine sub-domains that have a high contribution to the approximation error. As the approximation error is unknown in general cases, feedback algorithms commonly make use of *error estimation* techniques to guide the mesh refinement.

Having identified the regions for refinement, the second question is whether to increase the local approximation accuracy by decreasing the respective element sizes (h-refinement) or by elevating the approximation order of the polynomial shape functions (p-refinement). As discussed in Chapter 3, high-order approximations yield an exponential decay of the error if the target function is sufficiently smooth. In contrast, singular characteristics in the analytical solution are best resolved by a decrease of the element size. Accordingly, the refinement strategy has to be based on the local smoothness of the solution. Hence, a suitable *smoothness indicator* is the second integral part of an adaptive hp-refinement scheme.

Combining these two types of indicators, an automatic hp-refinement schemes has the following structure [Mitchell and McClain, 2011a,b]

1. Chose an initial discretization
2. **Repeat**
   (a) Form and solve the finite element equation system
   (b) For all elements: compute error indicator value
   (c) For all elements with a high error contribution:
      i. Estimate the smoothness of the element solution
      ii. If the local element solution is smooth, elevate ansatz order p
      iii. Else, perform h-refinement
3. **Until** some convergence criterion is met

In this chapter, the construction of an automatic error controlled multi-level hp-refinement scheme is outlined. To this end, Section 7.1 briefly reviews the characteristics of *a posteriori* error estimation techniques and discusses the essential ideas of the used explicit residual-based error estimator. Section 7.2 outlines the essential ideas of *a posteriori* smoothness estimation. In the final part of this chapter, the two techniques are combined into an error controlled multi-level hp-adaptive refinement scheme.

7.1 *A posteriori* error estimation

As discussed in Chapter 3, the aim of an *a priori* error estimation is to assess the convergence and the stability of the approximation scheme. Accordingly, these findings have a general nature and classify the approximation method itself (e.g h-, p-, or hp-FEM). On contrary, the central aim of an *a posteriori* error estimation technique is to give a quantitative estimate of the approximation error e of a *specific* numerical solution of a *specific* problem under consideration. The following two sub-sections briefly review the essential characteristics of *a posteriori* error estimators and discuss the explicit residual-based estimator used throughout this work.

7.1.1 Essential characteristics of *a posteriori* error estimators

Starting at the end of the 1970's, different error estimation strategies have been developed and fine tuned for the respective application scenarios. A comprehensive review of these

methods can be found in e.g. [Ainsworth, 2000; Ainsworth and Oden, 1997; Verfürth, 1996, 1999]. Closely following these works, this section outlines the essential idea of *a posteriori* error estimators and their quality constraints. To this end, the Poisson equation is chosen as a model problem

$$-\Delta u = \hat{s} \quad \forall \boldsymbol{x} \in \Omega \tag{7.2a}$$

$$\text{with} \quad u = 0 \quad \forall \boldsymbol{x} \in \Gamma_D \tag{7.2b}$$

$$\text{and} \quad \nabla u = \hat{q} \quad \forall \boldsymbol{x} \in \Gamma_N. \tag{7.2c}$$

As outlined in Chapter 2, the Galerkin method is based on the variational form of the above equation. For the present case, this reads

$$\text{Find } u \in \mathcal{V} \text{ such that } a(u,v) = f(v) + \int_{\partial\Gamma_N} v\hat{q}\,\mathrm{d}\Gamma \quad \forall\, v \in \mathcal{V}, \tag{7.3a}$$

$$\text{with} \quad a(u,v) = \int_\Omega \nabla u \cdot \nabla v\,\mathrm{d}\Omega, \quad f(v_h) = \int_\Omega v\hat{s}\,\mathrm{d}\Omega \tag{7.3b}$$

$$\text{and} \quad \mathcal{V} = \left\{ v \in H^1 : v = 0 \quad \forall \boldsymbol{x} \in \Gamma_D \right\}. \tag{7.3c}$$

Using the Finite Element Method, the Galerkin approximation u_h is found by restricting the above problem to a finite dimensional sub-space $\mathcal{V}_h \subset \mathcal{V}$. The difference between the analytical solution u and the numerical approximation u_h gives the approximation error $e = u - u_h$, which is to be estimated. In the present work, the error is measured in the energy norm.

The error $e \in \mathcal{V}$ is characterized by the following residual problem

$$a(e,v) = a(u-u_h,v) = f(v) - a(u_h,v) + \int_{\partial\Gamma_N} v\hat{q}\,\mathrm{d}\Gamma \quad \forall\, v \in \mathcal{V}. \tag{7.4}$$

Accordingly, the error could be estimated by solving the above problem numerically using a second finite element computation. However, the true error is a-orthogonal to all elements in $\mathcal{V}_h$ according to the Galerkin orthogonality, *cf.* (2.46). Therefore, the space for the second approximation would have to be chosen larger than the original space $\mathcal{V}_h$. This renders the direct approximation of the error more expensive than the original finite element approximation and thus unfeasible.

For this reason, error estimation techniques aim at providing an accurate approximation to the true error that can be computed at a significantly lower computational cost as the original finite element solution. Clearly, it is desirable that this estimate is *reliable* in the sense that—at least up to an (unknown) constant C_1—the value η provides an upper bound of the true error, i.e.

$$||e||_E \leq \frac{1}{C_1}\eta. \tag{7.5}$$

Similarly, it is of advantage if the estimate is *efficient* in the sense that—again up to an unknown constant C_2—the value η provides a lower bound of the true error, i.e.

$$\eta \leq C_2||e||_E. \tag{7.6}$$

If the estimate is both, reliable and efficient, i.e.

$$C_1||e||_E \leq \eta \leq C_2||e||_E, \tag{7.7}$$

the estimated error "tends to zero at the same rate as the true error" [Ainsworth, 2000, p. 18]. The ratio between the true and the estimated error is denoted as the *effectivity* index

$$\theta = \frac{\eta}{||e||_E}. \tag{7.8}$$

This index measures the under- or over-estimation of the error and therefore the quality of the estimation scheme. Clearly, the optimal value of θ is unity.

When using the estimator to guide a mesh refinement scheme, the estimator should also indicate the spatial distribution of the error. To this end, the global error estimator η is commonly composed of error *indicators* η_K, which are defined locally either on element or on patch level [Babuška, 1986; Babuška and Miller, 1984, 1987]. The global estimator is then given by the sum of these components

$$\eta^2 = \sum_{K \in \mathcal{T}_h} \eta_K^2. \tag{7.9}$$

In their classification of different estimation schemes, Ainsworth [2000]; Ainsworth and Oden [1997] distinguish between *explicit* and *implicit* schemes. Explicit approaches solely use directly available data $(u_h, \hat{s}, \hat{g})$, whereas implicit approaches commonly involve the solution of small auxiliary problems. Within this work, an explicit residual-based scheme is used to estimate the approximation error. The details of this approach are discussed in the next section. A review of alternative approaches such as implicit or recovery-based schemes can be found in e.g [Ainsworth, 2000; Verfürth, 1996, 1999].

7.1.2 Explicit residual-based *a posteriori* estimators

As mentioned above, the major aim of an explicit error estimator is to provide a computationally cheap approximation of the true error that can be directly computed from the available data set $(u_h, \hat{s}, \hat{g})$. To this end, this estimator is based on the *residual* r, which measures the numerical error in the strong form of the problem, i.e.

$$r = -\Delta e = \Delta(u_h - u) = \Delta u_h - \Delta u = \Delta u_h + \hat{s}. \tag{7.10}$$

As the residual is easily computable, the essential idea of explicit residual-based estimation schemes is to provide an upper bound of the true error in terms of the residual. This approach was first presented for one-dimensional finite elements in [Babuška and Rheinboldt, 1978a]. In [Babuška and Rheinboldt, 1978b, 1979], the idea was extended and combined with an mesh optimization scheme to yield an adaptive refinement algorithm. A more detailed mathematical analysis was presented in [Babuška and Rheinboldt, 1981] proving the properties of the estimator for general L_p-energy norms. In [Kelly et al., 1983], the idea was extended to two-dimensional finite element meshes, and in [Babuška and Miller, 1984, 1987] a sound mathematical analysis was provided with less limiting assumptions on the regularity of the analytical solution. Closely following the reviews given in e.g. [Ainsworth, 2000; Verfürth, 1996], this error estimation technique is outlined in this section.

The starting point is the aforementioned weak form of the residual equation, which is split into the individual element contribution as follows:

$$\begin{aligned} a(e,v) &= a(u,v) - a(u_h,v) = f(v) - a(u_h,v) + \int_{\partial\Gamma_N} v\hat{q}\,\mathrm{d}\Gamma \\ &= \int_\Omega v\hat{s}\,\mathrm{d}\Omega - \int_\Omega \nabla u_h \cdot \nabla v\,\mathrm{d}\Omega + \int_{\partial\Gamma_N} v\hat{q}\,\mathrm{d}\Gamma \\ &= \sum_{K\in\mathcal{T}_h} \left\{ \int_K v\hat{s}\,\mathrm{d}\Omega - \int_K \nabla u_h \cdot \nabla v\,\mathrm{d}\Omega + \int_{\partial K\cap\Gamma_N} v\hat{q}\,\mathrm{d}\Gamma \right\}. \end{aligned} \tag{7.11}$$

With the help of the Gauss'-divergence theorem, the second integral term can be reformulated in terms of the Laplace operator and the flux over the element boundary yielding

$$\begin{aligned} a(e,v) &= \sum_{K\in\mathcal{T}_h} \left\{ \int_K v\hat{s}\,\mathrm{d}\Omega + \int_K \Delta u_h v\,\mathrm{d}\Omega - \int_{\partial K} v\nabla u_h\cdot\boldsymbol{n}\,\mathrm{d}\Gamma + \int_{\partial K\cap\Gamma_N} v\hat{q}\,\mathrm{d}\Gamma \right\} \\ &= \sum_{K\in\mathcal{T}_h} \left\{ \int_K v\left(\hat{s}+\Delta u_h\right)\,\mathrm{d}\Omega + \int_{\partial K\cap\Gamma_N} v\left(\hat{q}-\nabla u_h\cdot\boldsymbol{n}\right)\,\mathrm{d}\Gamma - \int_{\partial K\setminus\Gamma_N} v\nabla u_h\cdot\boldsymbol{n}\,\mathrm{d}\Gamma \right\}. \end{aligned} \tag{7.12}$$

In the next step, the inter-element interfaces $\partial\mathcal{T}_h^{\mathrm{int}}$ are separated from the boundary interfaces. Further exploiting that $v = 0$ on Γ_D, this allows to write the error expression as

$$\begin{aligned} a(e,v) &= \sum_{K\in\mathcal{T}_h} \left\{ \int_K v\left(\hat{s}+\Delta u_h\right)\,\mathrm{d}\Omega + \int_{\partial K\cap\Gamma_N} v\left(\hat{q}-\nabla u_h\cdot\boldsymbol{n}\right)\,\mathrm{d}\Gamma \right\} \\ &\quad + \sum_{\gamma\in\partial\mathcal{T}_h^{\mathrm{int}}} \left\{ \int_\gamma v\left(\nabla u_h^{K_2} - \nabla u_h^{K_1}\right)\cdot\boldsymbol{n}_\gamma\,\mathrm{d}\Gamma \right\}. \end{aligned} \tag{7.13}$$

Here, the term $\nabla u_h^{K_1} - \nabla u_h^{K_2}$ denotes the jump of the gradient along the interface γ between the elements K_1 and K_2. Further, $\boldsymbol{n}_\gamma$ denotes the normal of the interface γ pointing outside of K_1.

Using the above separation allows to reformulate the error expression in terms of the element-internal residual r and the jump of the flux across element interfaces as follows

$$a(e,v) = \sum_{K\in\mathcal{T}_h} \left\{ \int_K vr(u_h)\,\mathrm{d}\Omega + \int_{\partial K} vj(u_h)\,\mathrm{d}\Gamma \right\} \tag{7.14a}$$

$$\text{with} \quad r(u_h) = \hat{s} + \Delta u_h \tag{7.14b}$$

$$\text{and} \quad j(u_h) = \begin{cases} \frac{1}{2}\left(\nabla u_h^{K_2} - \nabla u_h^{K_1}\right)\cdot\boldsymbol{n}_\gamma & \forall\,\boldsymbol{x}\in\partial K\setminus\Gamma_N \\ \hat{q} - \nabla u_h\cdot\boldsymbol{n} & \forall\,\boldsymbol{x}\in\partial K\cap\Gamma_N \\ 0 & \forall\,\boldsymbol{x}\in\partial K\cap\Gamma_D. \end{cases} \tag{7.14c}$$

Here, the jump term is extend by the factor 1/2 as every internal interface is visited once by both adjacent elements.

In the next step, the test function v is projected on the finite element space $\mathcal{V}_h$ using the Clément interpolation operator $\mathcal{I}_h : H^1 \to \mathcal{V}_h$ [Carstensen, 2006; Clément, 1975]. According to the Galerkin orthogonality,

$$a(e, \mathcal{I}_h(v)) = 0 \tag{7.15}$$

as $\mathcal{I}_h(v) \in \mathcal{V}_h$. Combining this with the Cauchy-Schwarz inequality yields

$$\begin{aligned} a(e,v) &= a(e, v - \mathcal{I}_h(v)) \\ &= \sum_{K \in \mathcal{T}_h} \left\{ \int_K (v - \mathcal{I}_h(v))\, r(u_h)\, \mathrm{d}\Omega + \int_{\partial K} (v - \mathcal{I}_h(v))\, j(u_h)\, \mathrm{d}\Gamma \right\} \\ &\leq \sum_{K \in \mathcal{T}_h} \left\{ ||v - \mathcal{I}_h(v)||_{L_2(K)}\, ||r(u_h)||_{L_2(K)} + ||v - \mathcal{I}_h(v)||_{L_2(\partial K)}\, ||j(u_h)||_{L_2(\partial K)} \right\} \end{aligned} \tag{7.16}$$

As show in e.g. [Ainsworth, 2000; Bernardi and Girault, 1998; Carstensen, 2006; Clément, 1975], the Clément interpolation error can be estimated in the following way

$$||v - \mathcal{I}_h(v)||_{L^2(K)} \leq C_{1,\mathcal{T}_h} h_K ||v||_{H^1(\tilde{K})} \qquad \forall K \in \mathcal{T}_h, v \in H^1(\Omega) \tag{7.17a}$$

$$||v - \mathcal{I}_h(v)||_{L^2(\partial K)} \leq C_{2,\mathcal{T}_h} \sqrt{h_K} ||v||_{H^1(\tilde{K})} \qquad \forall K \in \mathcal{T}_h, v \in H^1(\Omega), \tag{7.17b}$$

where $\tilde{K}$ denotes the union of K with all its adjacent elements and the constants $C_{1,\mathcal{T}_h}$ and $C_{2,\mathcal{T}_h}$ depend only on the regularity of the mesh.

Using these results and again the Cauchy-Schwarz inequality yields

$$\begin{aligned} a(e,v) &\leq \sum_{K \in \mathcal{T}_h} \left\{ \left(C_{1,\mathcal{T}_h} h_K\, ||r(u_h)||_{L_2(K)} + C_{2,\mathcal{T}_h} \sqrt{h_K}\, ||j(u_h)||_{L_2(\partial K)} \right) ||v||_{H^1(\tilde{K})} \right\} \\ &\leq \left\{ \sum_{K \in \mathcal{T}_h} C_{1,\mathcal{T}_h}^2 h_K^2\, ||r(u_h)||_{L_2(K)}^2 + C_{2,\mathcal{T}_h}^2 h_K\, ||j(u_h)||_{L_2(\partial K)}^2 \right\}^{1/2} \left\{ \sum_{K \in \mathcal{T}_h} ||v||_{H^1(\tilde{K})}^2 \right\}^{1/2} \\ &\leq C_{3,\mathcal{T}_h} ||v||_{H^1(\Omega)} \left\{ \sum_{K \in \mathcal{T}_h} C_{1,\mathcal{T}_h}^2 h_K^2\, ||r(u_h)||_{L_2(K)}^2 + C_{2,\mathcal{T}_h}^2 h_K\, ||j(u_h)||_{L_2(\partial K)}^2 \right\}^{1/2}. \end{aligned} \tag{7.18}$$

Here, the factor $C_{3,\mathcal{T}_h}$ takes into account that every element is counted several times when summing over all patches $\tilde{K}$ in the mesh.

The coercivity of the bi-linear form $a(\cdot,\cdot)$ allows bounding the H^1-norm in terms of the energy norm since

$$\forall w \in H^1(\Omega),\ \exists\, \beta \text{ such that } ||w||_{H^1(\Omega)} \leq \beta ||w||_{E(\Omega)}. \tag{7.19}$$

By further choosing $v = e$ in (7.18), the error expression can be written as

$$\begin{aligned} ||e||^2_{E(\Omega)} &= a(e,e) \\ &\leq C_{3,\mathcal{T}_h} ||e||_{H^1(\Omega)} \left\{ \sum_{K \in \mathcal{T}_h} C^2_{1,\mathcal{T}_h} h^2_K \, ||r(u_h)||^2_{L_2(K)} + C^2_{2,\mathcal{T}_h} h_K \, ||j(u_h)||^2_{L_2(\partial K)} \right\}^{1/2} \\ &\leq C_{3,\mathcal{T}_h} \beta ||e||_{E(\Omega)} \left\{ \sum_{K \in \mathcal{T}_h} C^2_{1,\mathcal{T}_h} h^2_K \, ||r(u_h)||^2_{L_2(K)} + C^2_{2,\mathcal{T}_h} h_K \, ||j(u_h)||^2_{L_2(\partial K)} \right\}^{1/2} \end{aligned} \tag{7.20}$$

Accordingly, the error in the energy norm can be estimated as

$$||e||_{E(\Omega)} \leq C_{3,\mathcal{T}_h} \beta \left\{ \sum_{K \in \mathcal{T}_h} C^2_{1,\mathcal{T}_h} h^2_K \, ||r(u_h)||^2_{L_2(K)} + C^2_{2,\mathcal{T}_h} h_K \, ||j(u_h)||^2_{L_2(\partial K)} \right\}^{1/2} \tag{7.21}$$

or equivalently

$$\begin{aligned} &||e||^2_{E(\Omega)} \leq C \left\{ \sum_{K \in \mathcal{T}_h} C^2_{1,\mathcal{T}_h} h^2_K \, ||r(u_h)||^2_{L_2(K)} + C^2_{2,\mathcal{T}_h} h_K \, ||j(u_h)||^2_{L_2(\partial K)} \right\} \\ &\text{with } C = C^2_{3,\mathcal{T}_h} \beta^2. \end{aligned} \tag{7.22}$$

These results demonstrate that—up to a constant—the error in the energy norm can be estimated explicitly in terms of the element-internal residual r and the jump of the numerical flux j across the element boundaries. Moreover, the estimator can be computed locally on element level. This suggests the definition of an element error *indicator* as

$$\eta^2_K = C^2_{1,\mathcal{T}_h} h^2_K \, ||r(u_h)||^2_{L_2(K)} + C^2_{2,\mathcal{T}_h} h_K \, ||j(u_h)||^2_{L_2(\partial K)}, \tag{7.23}$$

which measures the contribution of element K to the global error bound. The estimator of the global error is then given by

$$||e||^2_{E(\Omega)} \leq \eta^2 = C \sum_{K \in \mathcal{T}_h} \eta^2_K. \tag{7.24}$$

Accordingly, the scheme provides an upper estimate for the global approximation error of the numerical solution *and* its spatial distribution across the elements of the mesh. This renders the estimator suitable to guide the adaptive feedback scheme. In case of finite elements with linear shape functions, the estimator also provides a lower bound for the error in the energy norm as shown in e.g. [Ainsworth, 2000]. This renders the estimator *efficient*.

7.2 Smoothness estimation

As outlined in the introduction of this chapter, the *hp*-version of the Finite Element Method offers two different refinement techniques: the accuracy can be increased by either decreasing the element size (*h*-refinement) or by elevating the approximation order of the finite element shape functions (*p*-elevation/refinement). After having identified the elements with the largest

error contribution using one of the aforementioned estimation techniques, it is essential to decide for the most efficient type of refinement. To this end, the local regularity of the solution has to be estimated. Over the past decades, various different techniques have been developed to extract this information from the current numerical approximation. A detailed review and comparison of thirteen different strategies is given in [Mitchell and McClain, 2011a,b, 2014]. Closely following these works, the current section briefly reviews the different approaches.

A priori knowledge

A natural approach to decide between h- and p-refinement is to use *a priori* knowledge or engineering experience to design the hp-mesh. Possible application scenarios include cases of re-entrant corners, changing boundary conditions, or non-smooth material interfaces.

Ainsworth and Senior [1999] follow this approach by "flagging" nodes of the mesh, where the solution is expected to be irregular. All elements adjacent to these candidates are then h-refined. On all other elements a p-refinement is performed. In [Bernardi et al., 2001], a similar idea is used in the context of Stokes flow.

Type parameter

Gui and Babuška [1986c] suggest the estimation of the regularity in terms of the ratio between the estimated error of two consecutive p-refinements

$$R(K) = \begin{cases} \frac{\eta_{K,p}}{\eta_{K,p-1}} & \text{if } \eta_{K,p-1} \neq 0 \\ 0 & \text{if } \eta_{K,p-1} = 0. \end{cases} \tag{7.25}$$

The refinement type is decided based on a user-defined *type* parameter γ. An element is h-refined if $R(K) > \gamma$ and p-refinement if $R(K) \leq \gamma$. In [Adjerid et al., 1999], this idea is applied in the context of convection diffusion problems.

Local regularity estimation using larger p estimates

A similar idea presented in [Ainsworth and Senior, 1998] is to estimate the regularity by solving a local residual problem of the error estimator for $p+1$, $p+2$, and $p+3$. In addition to the estimate of the local approximation error, this approach also provides the decay rate of the error and with it the regularity of the local solution. If the regularity is high, the respective element is marked for p-refinement. Otherwise, the local approximation quality is increased by an h-refinement.

h- and p-based error estimates

A similar approach is suggested [Schmidt and Siebert, 2000]. It is based on the observation that the solution of a local residual problem of the error estimator indicates how much the error will be reduced by increasing the discrete function space $\mathcal{V}_K$. This insight is used to quantify the quality of the different refinement strategies. To this end, the local residual problem is solved once using a locally p-refined and once using a locally h-refined discretization. Based on the resulting two estimates, the better refinement strategy for the respective element is selected.

Predicted error reduction

Melenk and Wohlmuth [2001] follow a similar idea to choose the refinement strategy. But instead of actually performing the two different types of refinement, the error level after the refinement is predicted by means of suitable *a priori* estimates. If the actual error of the refined discretization is larger than expected, the solution is assumed to be non-smooth. Therefore, the element is marked for h-refinement. If the actual error is smaller, the approximation accuracy is increased by elevating the ansatz order of the element.

Texas 3 step

The idea of the Texas 3 step approach, introduced in [Bey et al., 1995; Oden and Patra, 1995], is to perform the h- and p-refinements separately. To this end, the first step is to refine an initially coarse mesh by performing an almost uniform h-refinement until the asymptotic range is reached. In the second step, an adaptive h-refinement is performed until a pre-defined intermediate tolerance is reached. In the third and final step, an adaptive p-elevation is used to reach the final accuracy specified by the user.

Mesh optimization via a reference solution

The idea of this approach is to choose the appropriate refinement strategy by comparing the current numerical solution $u_{h,p}$ to a reference solution u_{ref}. For benchmark problems, the analytical solution is the natural choice for the reference. For general applications, Demkowicz [2007]; Demkowicz et al. [2002]; Šolín [2004]; Šolín and Demkowicz [2004] suggest performing a global h- and p-refinement for all elements and to use the respective solution $u_{h/2,p+1}$ as the reference solution. To limit the computational costs of this second computation, the information of the coarse mesh solution $u_{h,p}$ is exploited by computing the fine mesh solution $u_{h/2,p+1}$ using a multi-grid solver.

Using the fine mesh reference solution, the gain in accuracy of the different refinement possibilities is compared by locally projecting the reference solution onto the refined finite element space. The strategy yielding the largest error decrease is then selected for the respective element.

Decay of Legendre coefficients

An alternative approach suggested in [Mavriplis, 1994] is to estimate the smoothness of the numerical solution in terms of its Legendre expansion. To this end, the current approximation on the respective element is written in the form

$$u_h(x)|_K = \sum_{j=0}^{p_K} a_j L_j(x), \tag{7.26}$$

where $L_j(\boldsymbol{x})$ is the Legendre polynomial of degree j *cf.* Section 3.2.1, and a_j is the corresponding coefficient of the expansion. This reformulation of the numerical solution allows to quantify its regularity by considering the decay rate of the coefficients a_j. To this end, Mavriplis [1994] suggests to perform a least squares fit of the last four coefficients to the exponential decay

function $Ce^{-\sigma j}$. The local solution is considered to be regular if the decay rate σ is larger than one. Accordingly, the element is p-refined in this case and h-refined if $\sigma \leq 1$.

In Houston et al. [2003]; Hsu and Mavriplis [1998]; Mavriplis [1994] the applicability of this scheme is demonstrated for Poisson's and Navier-Stokes problems.

7.3 Adaptive multi-level *hp*-refinement[1]

In the two previous sections, the essential ideas of different error and smoothness estimators were outlined. The aim of this third section is to combine these techniques with the suggested overlay refinement approach to yield an error-controlled multi-level *hp*-adaptive refinement scheme. Accordingly, a suitable strategy for both types of estimators has to be chosen.

To decide *where* to refine, the explicit residual-based error estimator outlined in Section 7.1.2 is used as it has shown to perform well in the context of p-FEM in e.g. [Düster, 2002; Düster et al., 2001]. To decide *how* to refine, the results of the detailed study presented in [Mitchell and McClain, 2011a,b, 2014] are used. Here, the different smoothness estimation techniques discussed in the previous section were compared for a benchmark suite consisting of 21 test cases covering Poisson's, linear elastic, convection diffusion, Helmholtz, and interface problems. The presented results demonstrate that the decay-rate estimator suggested in [Mavriplis, 1994] yields excellent results for singular problems at engineering accuracy (1% error) and still good results at high accuracy (10^{-4}% error). At the same time, the simplicity of the implementation and the low computational overhead are convincing. For these reasons, the scheme suggested in [Mavriplis, 1994] is used to decide the correct type of refinement.

However, both types of estimators are formulated in the context of the classical refine-by-replacement approach. Therefore, the techniques have to be re-formulated for their application in combination with the suggested refine-by-superposition strategy.

For the error estimator, this requires quantifying the approximation error in the hierarchy of overlay meshes. This is surprisingly simple as it only demands a re-interpretation of the overlay mesh structure. The motivation for this process is the derivation of the explicit residual-based error estimator, which in (7.12) makes use of the Gauss'-divergence theorem. For this step to be valid, the numerical solution u_h has to be twice continuously differentiable within each K. This is not given when considering an element K that has been refined by overlay elements as the overlay shape functions are only C^0-continuous within the base element K. Accordingly, the element domain K has to be partitioned into a set of non-overlapping sub-domains $\{P_i\}$ with $\bigcup_i P_i = K$ such that the numerical solution is in $C^2(P_i)$ for each sub-domain. In this way, a partition-wise error indication can be computed

$$\eta_{P_i}^2 = C_{1,\mathcal{T}_h}^2 h_{P_i}^2 \, ||r(u_h)||_{L_2(P_i)}^2 + C_{2,\mathcal{T}_h}^2 h_{P_i} \, ||j(u_h)||_{L_2(\partial P_i)}^2 \tag{7.27}$$

for each sub-partition of the respective element K.

In analogy to the integration of the stiffness matrix discussed in Section 6.4, the coarsest partition of that kind is the set of all leaf elements, *cf.* Figure 7.1. Using this natural partition, the error indicator values can be associated directly to the leaf elements of the mesh. It is essential to note that this does *not* imply that only the shape functions of this highest

[1] This section is based on the Master's thesis [D'Angella, 2015] that was supervised by the author of this work. The resulting publication [D'Angella et al., 2016] was co-authored by the author of this work.

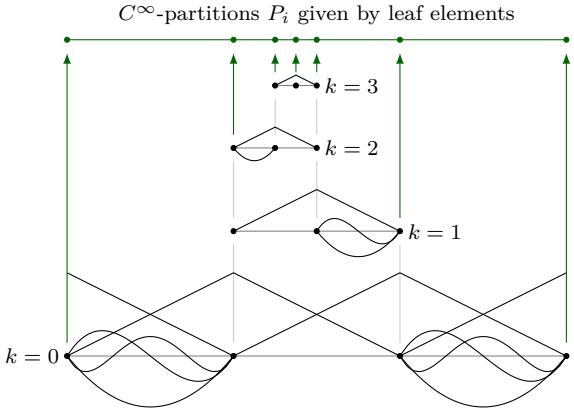


Figure 7.1: Partitioning of hierarchically refined mesh into sub-domains P_i in which the numerical solution is C^∞-continuous. The coarsest partition of that kind is given by the projection of the leaf elements.

element level are used to compute the residual. Instead, the *complete* numerical solution u_h has to be considered. Accordingly, all non-zero shape functions of all elements in the hierarchical mesh are evaluated to compute the residual. Therefore, the error estimator is not divided hierarchically according to the refinement levels but split according to the geometry of the leaf elements. This is of great advantage as it allows guiding the adaptive feedback scheme by marking all leaf elements for refinement that have an estimated error contribution $\eta_{P_i} \geq C_E \max_{P_j \in \mathcal{T}_h}(\eta_{P_j})$. Here, C_E is a user-defined threshold level relative to the maximum error in the mesh.

In analogy to the error estimator, the smoothness estimator is also considered partition-wise. To this end, the extension to higher dimensions suggested in [Houston et al., 2003] is followed by computing e.g. the two-dimensional Legendre coefficient matrix $\boldsymbol{\alpha}$ as

$$\alpha_{ij} = \frac{2j+1}{2}\frac{2i+1}{2} \int_{s=-1}^{1} \int_{r=-1}^{1} u_h(x(r,s), y(r,s)) L_i(r) L_j(s) |\boldsymbol{J}| \, \mathrm{d}r \, \mathrm{d}s \tag{7.28}$$
$$\text{with} \quad 0 \leq i \leq p_x^K, 0 \leq j \leq p_y^K$$

for each leaf element in the hierarchical mesh. Here, $\boldsymbol{J}$ denotes the Jacobian matrix of the mapping between the reference and physical element. As before, not only the shape functions of the respective leaf element but the complete numerical solution u_h is considered. From the coefficient matrix $\boldsymbol{\alpha}$, two one-dimensional sets of coefficients are obtained by summing the rows and columns of $\boldsymbol{\alpha}$. Using these two sets, the local smoothness of the current numerical solution is assessed by the exponential decay rates σ^r and σ^s of the coefficients. A p-refinement is performed if $\sigma = \max(\sigma^r, \sigma^s) \geq C_S$, with C_S being a user defined threshold for the smoothness. Otherwise, the accuracy is increased by an isotropic h-refinement.

Combining both types of estimators, the final error-controlled multi-level hp-feedback scheme is given by Algorithm 6. To simplify the notation, all leaf elements of the hierarchically refinement mesh $\mathcal{T}_h$ are denoted by K^l.

Algorithm 6 Error controlled multi-level *hp*-adaptivity

function MULTI-LEVEL *hp*-ADAPTIVITY(initial mesh $\mathcal{T}_h^0$, target accuracy TOL, max. number of mesh adaption steps $k_{\max}$)
  $k = 0$
  for $k < k_{\max}$ **do**
    Solve problem on $\mathcal{T}_h^k$
    Compute $\eta_{K_i^l} \, \forall \, K_i^l \in \mathcal{T}_h^k$
    if $\sum_{K_i^l \in \mathcal{T}_h^k} \eta_{K_i^l}^2 \leq TOL^2$ **then**
      break
    end if
    Mark leaf elements K_i^l for which $\eta_{K_i^l} \geq C_E \max_{K_i^l \in \mathcal{T}_h^k} (\eta_{K_i^l})$
    for each marked leaf element K_i^l **do**
      Compute decay rate σ of the Legendre coefficients on $\eta_{K_i^l}$
      if $\sigma \geq C_S$ **then**
        Update $\mathcal{T}_h^k$ by performing p-refinement on K_i^l
      else
        Update $\mathcal{T}_h^k$ by performing multi-level h-refinement on K_i^l
      end if
    end for
    $k \leftarrow k + 1$
  end for
end function

Part II

Application of multi-level hp-FEM

Chapter 8

Benchmark problems with singular or non-smooth solutions

In this first numerical examples chapter, the approximation quality of the multi-level *hp*-refinement strategy is analyzed for problems having a non-smooth or even singular solutions. The most prevalent cause of such demanding solution characteristics are re-entrant corners in the geometry of the domain Ω. Different benchmarks and applications of this kind are presented and discussed in Section 8.2. Besides the geometry, also the boundary conditions can cause singularities in the analytical solution. This aspect is addressed in Section 8.3. A third application demanding for *hp*-refinement are internal- or boundary-layer effects in which the solution changes its value with a high gradient. The use of the multi-level *hp*-method for this problem kind is analyzed in Section 8.4. In most of the following examples, the approximation quality of the *hp-d-* and the multi-level *hp*-method are compared. For this reason, the chapter begins with a motivating detailed analysis of the two approaches in the context of a one-dimensional bi-material example in Section 8.1.1. In a second motivating example, the suggested multi-level *hp*-approach is compared to the classical refine-by-replacement *hp*-formulation using the well know one-dimensional, singular benchmark of [Gui and Babuška, 1986a].

If not noted otherwise, the approximation quality is expressed in terms of the error in the energy norm. Following e.g. [Düster et al., 2008], the error is calculated as

$$||\boldsymbol{e}||_E = \sqrt{\frac{a(\boldsymbol{u}-\boldsymbol{u}_h, \boldsymbol{u}-\boldsymbol{u}_h)}{a(\boldsymbol{u},\boldsymbol{u})}} \cdot 100[\%] \tag{8.1a}$$

$$= \sqrt{\frac{|\, a(\boldsymbol{u},\boldsymbol{u}) - a(\boldsymbol{u}_h,\boldsymbol{u}_h)\,|}{a(\boldsymbol{u},\boldsymbol{u})}} \cdot 100[\%]. \tag{8.1b}$$

The above expression is based on the Galerkin orthogonality

$$a(\boldsymbol{u}-\boldsymbol{u}_h, \boldsymbol{u}-\boldsymbol{u}_h) = a(\boldsymbol{u},\boldsymbol{u}) - a(\boldsymbol{u}_h,\boldsymbol{u}_h), \tag{8.2}$$

discussed in Section 2.2. However, the equivalence of the expressions (8.1a) and (8.1b) is only given in case of exact arithmetic. When instead using finite precision computation, the way of computing the error significantly affects the nominal error value. Following the IEEE-standard for floating point arithmetic [IEEE Computer Society et al., 2008], the difference

$a(\boldsymbol{u}, \boldsymbol{u}) - a(\boldsymbol{u}_h, \boldsymbol{u}_h)$ in (8.1b) can be computed with an accuracy of 16 decimal digits using double precision (64 bits). Accordingly, the machine accuracy for the subtraction is 10^{-16}. Taking into account the square root and the percentage in (8.1b), the machine accuracy in the energy norm is given by $10^{-8} = 10^{-6}\%$.

Interestingly, this apparent decrease in accuracy does not occur in computing the error in the energy norm according to (8.1a). The reason is that $a(\boldsymbol{u} - \boldsymbol{u}_h, \boldsymbol{u} - \boldsymbol{u}_h)$ computes the square of the difference of $\boldsymbol{u}$ and $\boldsymbol{u}_h$ (whereas (8.1b) computes the difference of the squares). Thus, this expression can be computed up to a difference of 10^{-32} when using double precision. Again taking into account the square root and the percentage, the machine accuracy in the energy norm (8.1a) is $10^{-16} = 10^{-14}\%$.

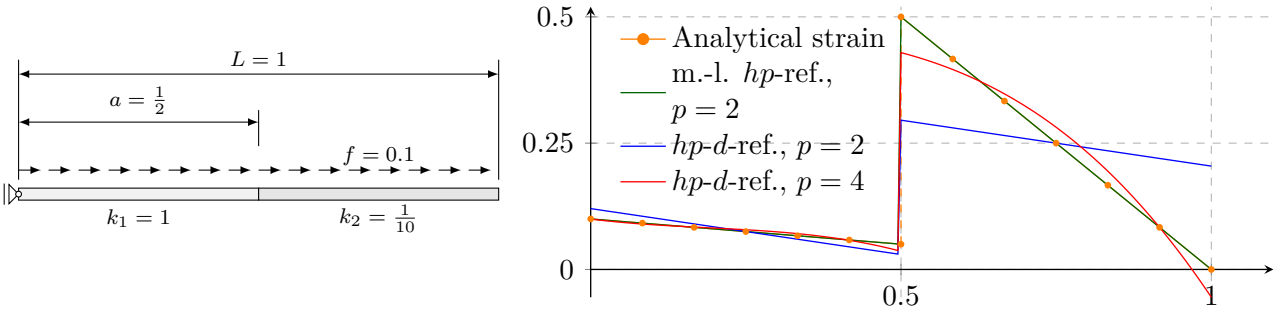
This apparent difference in the maximal accuracy has to be considered when assessing the accuracy of the approximation. If not noted otherwise, the relative error is computed according to (8.1b). Hence, a relative error in the energy norm of $10^{-6}\%$ corresponds to machine accuracy of 64-bit double precision.

8.1 Motivating one-dimensional examples

8.1.1 Comparison of *hp*-*d*- and multi-level *hp*-refinement

Before assessing the approximation quality of the multi-level *hp*-refinement method in the context of complex two- and three-dimensional settings, this first examples aims at comparing the numerical properties of the *hp*-*d*- and the multi-level *hp*-strategy in a simple, one-dimensional setup. To this end, an elastic, bi-material rod is considered. As depicted in Figure 8.1a, the structure is fixed at the left end and loaded with a constant body load. The rod is of length one and the material interface is positioned in the middle. The analytical displacement resulting from the loading is given by the following piecewise quadratic distribution:

$$u(x) = \begin{cases} \frac{f}{k_1} x \left(L - \frac{x}{2}\right) & \forall\, x \in [0, \frac{L}{2}) \\ \frac{f}{k_2} x \left(L - \frac{x}{2}\right) + f a (L - \frac{a}{2}) \frac{k_2 - k_1}{k_1 k_2} & \forall\, x \in [\frac{L}{2}, L] \end{cases} \tag{8.3}$$



(a) Setup of bi-material rod.

(b) Approximation of piecewise constant strain by different discretizations.

Figure 8.1: Bi-material rod benchmark. Used abbreviations: m.-l. *hp*: multi-level *hp*, ref.: refinement.

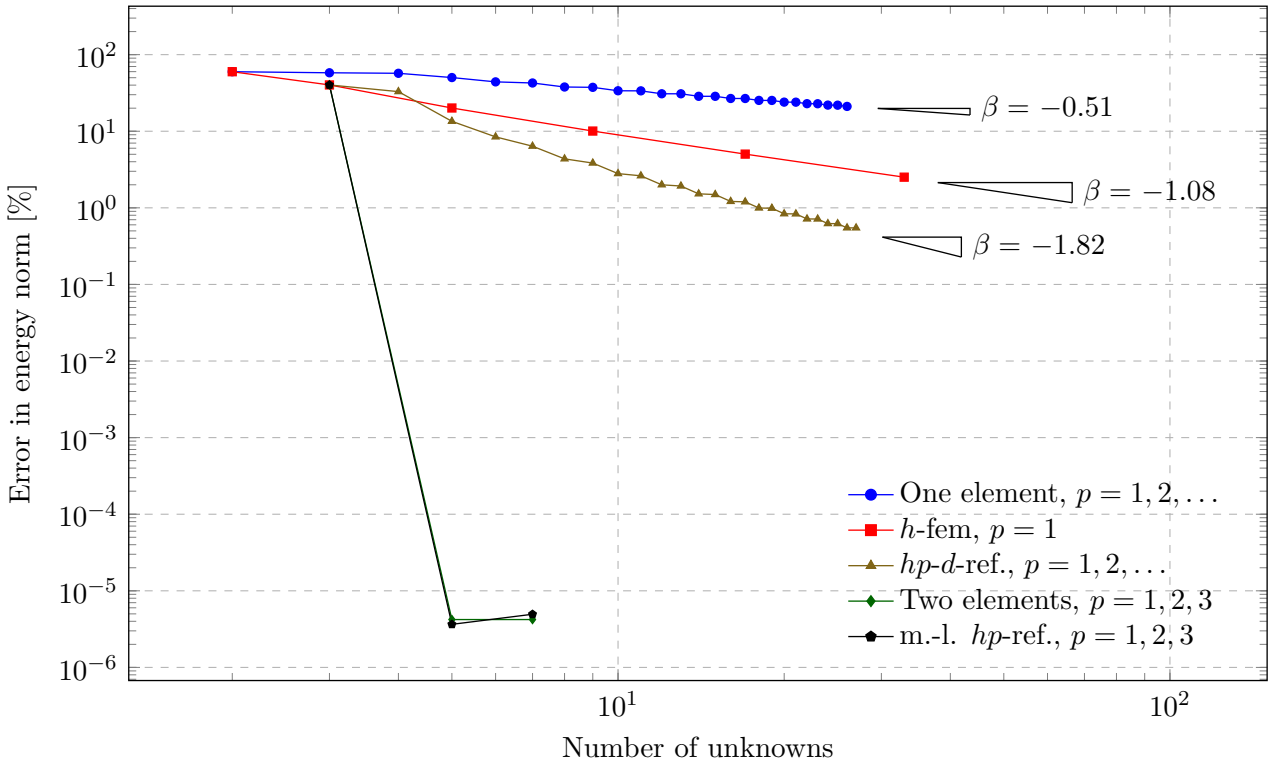


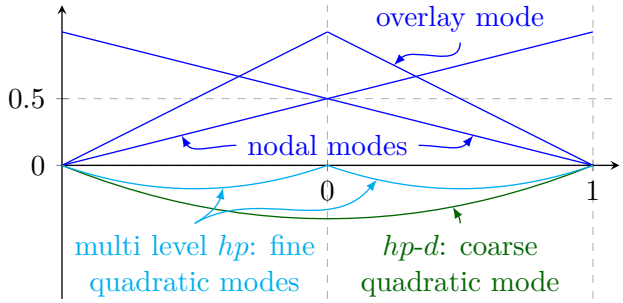
Figure 8.2: Convergence of bi-material rod benchmark. Used abbreviations: m.-l. *hp*: multi-level *hp*, ref.: refinement.

Accordingly, the resulting strain depicted in Figure 8.1b is a piece-wise linear function with different slopes in the two parts of the rod and a jump at the material interface.

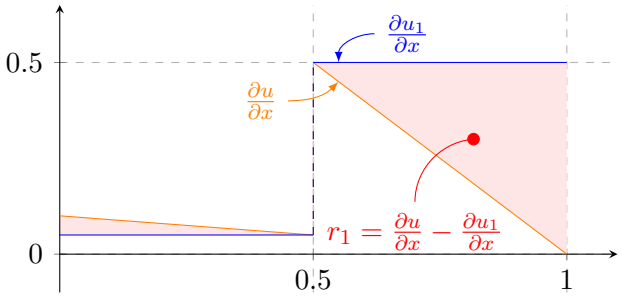
When using only *one* high-order element for the numerical approximation of the solution, the polynomial shape functions can not represent this kink-characteristic in the middle of the rod. Hence, the analytical solution cannot be captured exactly but only approximated in this numerical setup. The accuracy of this approximation can be increased by elevating the polynomial order of the used shape functions. However, since the material interface is not resolved by element boundaries, the solution is non-smooth *within* the element. For this reason, the approximation error only decays algebraically under a p-elevation as demonstrated in Figure 8.2.

These first results demonstrate that the exact representation of the solution demands for the resolution of the kink-characteristic by the mesh. Clearly, this is possible when refining the discretization by using two quadratic elements, whose central node resolves the interface. Accordingly, the error in the energy norm decreases to machine accuracy when using shape functions of order two as demonstrated in Figure 8.2.

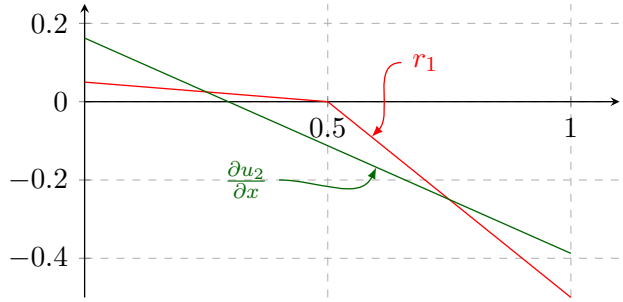
Alternatively, the quadratic base element can be refined following the *hp-d*-strategy by introducing an additional hat-function in the middle of the element. This is illustrated in Figure 8.3a. The refined set of basis functions can represent the kink-characteristic of the solution. However, even with this refinement the numerical approximation is not exact but only an approximation as results depicted in Figure 8.1b demonstrate. Moreover, when elevating the polynomial order of the shape functions, the convergence of the error remains algebraic,



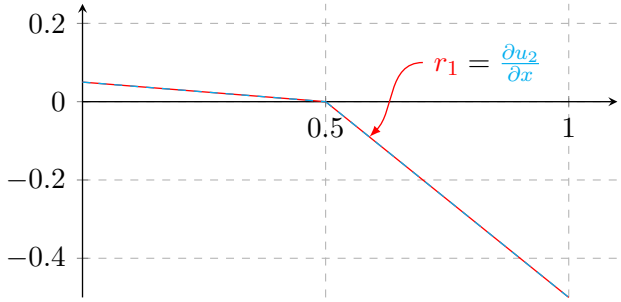
(a) Refined shape functions



(b) Approximation of piecewise linear strain with nodal and overlay modes.



(c) *hp-d*-approximation of residual with coarse high-order mode: the piecewise linear residual cannot be captured exactly on neither side of the interface.



(d) Multi-level *hp*-approximation of residual with fine high-order modes: the residual is captured exactly on both sides of the interface.

Figure 8.3: Approximation of piecewise quadratic function using the *hp-d*- and the multi-level *hp*-refinement strategy.

although with a higher rate than in the unrefined case.

This remaining approximation error can be explained by grouping the final three linear shape functions depicted in Figure 8.3a in a piecewise linear contribution u_1. In this way, the final numerical approximation can be written as:

$$u_{\text{num}} = u_1 + u_2, \tag{8.4}$$

with u_2 denoting the contribution of the quadratic shape function. The additional linear overlay mode introduced by the refinement breaks the C^∞-continuity of the final approximation in the middle of the coarse base element. Accordingly, the linear solution contribution u_1 can represent a piecewise constant strain, which accounts for the jump of the strain at the material interface as shown in Figure 8.3b. However, the residual between analytical strain and the approximation by u_1 remains piecewise linear with different slopes in the two parts of the rod. Thus, the linear overlay mode only shifts the kink from the displacement to the strain but does not resolve it. This characteristic in the strain cannot be captured by the remaining *single* quadratic mode u_2, as illustrated in Figure 8.3c: since this function has support on the original coarse base element, it cannot compensate the two different slopes of the residual in the left and right part of the rod. Thus, the analytical solution is not captured exactly but only approximated. Elevating the order of the approximation u_2 also does not resolve the problem as the support of these shape functions remains on the coarse base element. Hence, the situation is similar to the previously considered unrefined approximation, with the differ-

ence that the kink is not a characteristic of the displacement but of the strain. As this only weakens the strength of the irregularity, the approximation by the *hp-d*-refinement achieves higher convergence rates but is not exact.

This analysis demonstrates that the *hp-d*-refinement only breaks the continuity of the shape functions, while their support remains unchanged. Therefore, the superposition of a linear mode does not correspond to an *h*-refinement of the high-order base element.

The situation is fundamentally different when using the multi-level *hp*-approach as illustrated in Figure 8.3d. Since the high-order shape functions are moved from the base to the leaf-elements, their support is reduced. Hence, the two distinct parts of the residual r_1 can be compensated by the two quadratic overlay modes yielding an exact representation of the solution. Accordingly, the approximation error decays to machine accuracy as shown in Figure 8.2. This demonstrates that only the multi-level *hp*-overlay corresponds to a high-order *h*-refinement.

Admittedly, this tailored, one-dimensional benchmark is not representative for more complex applications. However, the example gives a clear reasoning for the superior approximation properties of the multi-level *hp*-approach compared to the *hp-d*-strategy, which is confirmed in various scenarios by the remaining examples following in this work.

8.1.2 Comparison of classical and multi-level *hp*-refinement[g]

The aim of this second example is to compare the classical refine-by-replacement *hp*-formulation of the Finite Element Method and the multi-level *hp*-refinement scheme in the context of a one-dimensional, singular example. To this end, the one-dimensional benchmark of [Gui and Babuška, 1986b] discussed in Section 3.1 is considered. Restating (3.2), the problem reads

$$\text{Find } u \text{ such that } u'' = -\alpha(\alpha-1)x^{\alpha-2} \quad \forall\, x \in (0,1) \tag{8.5a}$$

$$\text{with} \quad u(0) = 0, \quad u'(1) = 0 \quad \text{and} \quad \alpha = 0.65. \tag{8.5b}$$

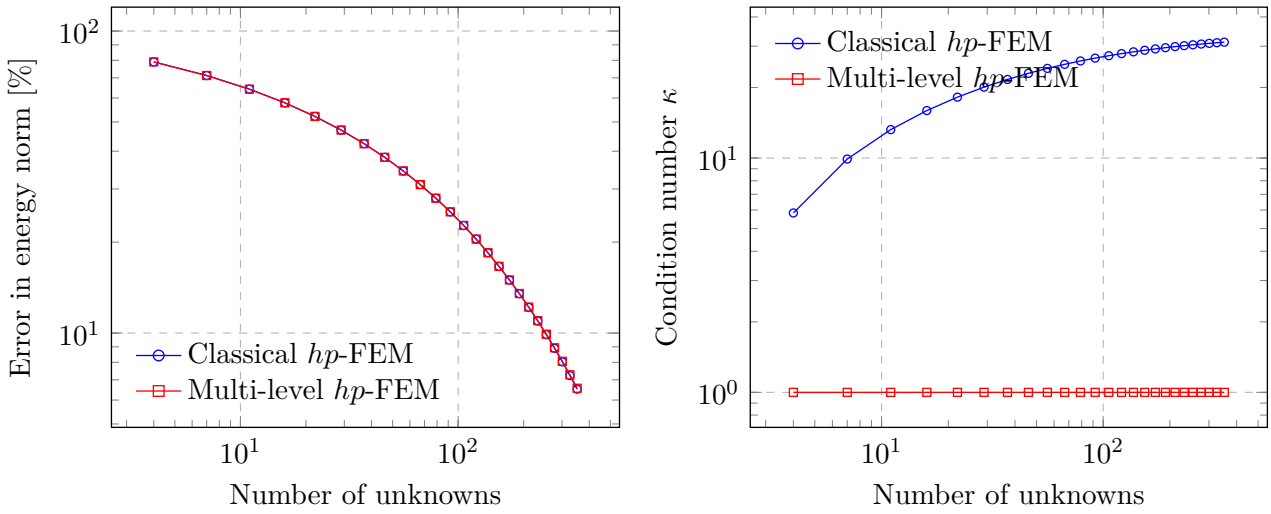
The analytical solution of this boundary value problem is given by

$$u(x) = -x^{\alpha} + \alpha x \quad \forall\, x \in [0,1]. \tag{8.6}$$

The singular nature of the analytical solution demands for the use of an *hp*-refinement to achieve an exponential convergence of the approximation error. To this end, the element size is graded geometrically towards the origin in a ratio of $q = 0.5$, and the polynomial degree of the ansatz functions is elevated linearly away from the singularity [Gui and Babuška, 1986b].

As illustrated in Figure 8.4a, this mesh design yields an exponential decay of the approximation error, which is identical for both, the classical and the multi-level *hp*-strategy. This is not surprising as both approaches construct the same ansatz space—at least for one-dimensional problems.

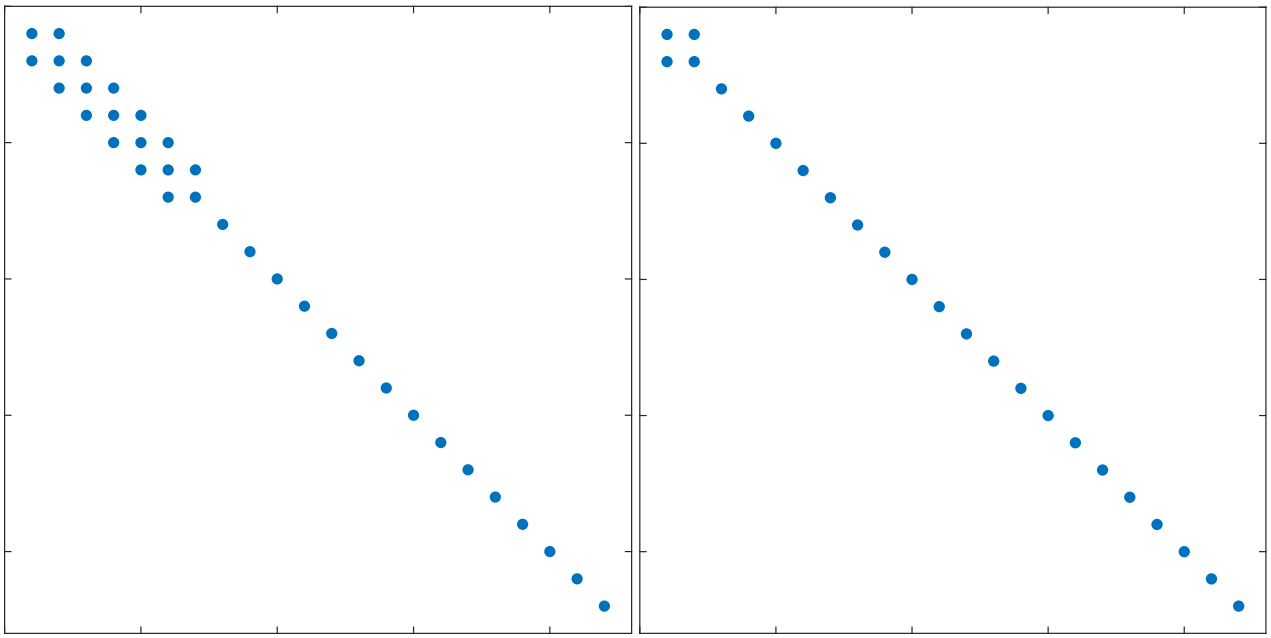
However, the different basis functions used by the two approaches lead to a significant difference in the condition number of the resulting stiffness matrix as illustrated in Figure 8.4b. While the condition number of the classical *hp*-approach is monotonically increasing, the multi-level *hp*-approach yields a constant condition number $\kappa = 1$ for all discretizations. This improved characteristic originates from the orthogonality of the hierarchical shape functions. As discussed in Section 3.2.1, the used integrated Legendre shape functions are designed



(a) Error in energy norm

(b) Condition number κ after Jacobi pre-conditioning

Figure 8.4: Comparison of classical and multi-level *hp*-FEM [Zander et al., 2016].



(a) Classical *hp*-FEM

(b) Multi-level *hp*-FEM

Figure 8.5: Comparison of sparsity patterns obtained using the classical and the multi-level *hp*-version of the finite element method (Number of unknowns: 22) [Zander et al., 2016].

to be orthogonal with respect to the bi-linear form $a(\cdot,\cdot)$. Interestingly, the same property also holds for the superposed, fine-scale hat functions as these are orthogonal to all shape functions on all other refinement levels. Therefore, the multi-level *hp*-approach yields an almost fully orthogonal *hp*-basis when starting the analysis with one base element. Only the coarse linear shape functions on the base mesh remain non-orthogonal. Hence, the resulting stiffness matrix is almost diagonal with only two off-diagonal entries resulting from the two linear shape functions the base mesh, *cf.* Figure 8.5b. After applying the Dirichlet boundary condition and performing a Jacobi pre-conditioning, the basis is fully orthonormal and the condition number is independent of the number of elements.

The situation is fundamentally different when using the classical refine-by-replacement approach. Here, all linear shape functions are non-orthogonal, resulting in more off-diagonal entries in the stiffness matrix as illustrated in Figure 8.5a. Accordingly, the simple Jacobi pre-conditioning is not sufficient to fully orthonormalize the stiffness matrix. This leads to a constant growth of the condition number when refining the discretization.

Just as for the integrated Legendre shape functions, the orthogonality properties of the superposed hat functions do *not* carry over directly to cases involving more complex geometric domains in higher dimensions. For this reason, the solvability of the final equation system for two- and three-dimensional problems is an important aspect, which is studied in the next sections.

8.2 Singularities due to re-entrant corners

8.2.1 L-shaped domain benchmark[a]

The aim of this first example is to analyze the convergence properties of the proposed refinement methods in the context of non-smooth problems. For this purpose, the well known L-shaped domain problem is considered following e.g. [Szabó and Babuška, 1991].[f]

8.2.1.1 Problem definition[f]

Consider the Poisson's problem

$$\Delta\phi = 0 \qquad \forall\,(r,\theta) \in \Omega \tag{8.7}$$

defined on the L-shaped domain depicted in Figure 8.6a, with the boundary conditions defined as

$$\phi = 0 \qquad \forall\,(r,\theta) \in \Gamma_D \tag{8.8}$$

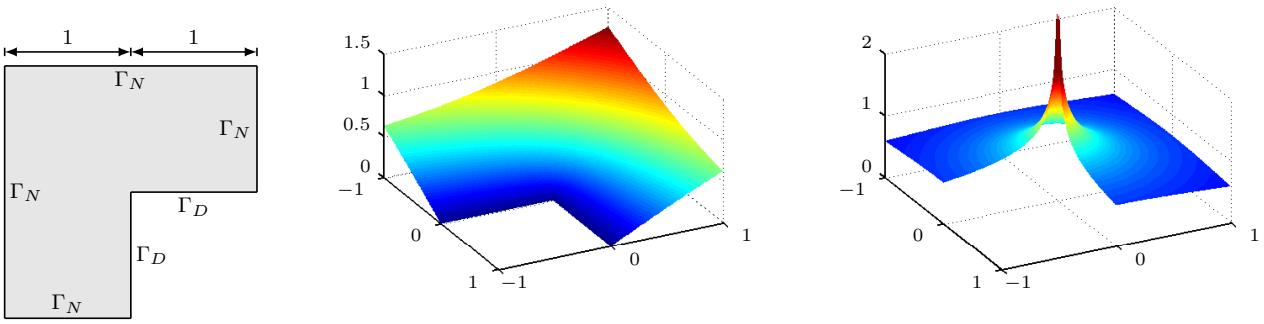
$$\nabla\phi \cdot \boldsymbol{n} = \frac{2}{3} r^{-\frac{4}{3}} \begin{bmatrix} x\sin\left(\frac{2}{3}\theta\right) - y\cos\left(\frac{2}{3}\theta\right) \\ y\sin\left(\frac{2}{3}\theta\right) + x\cos\left(\frac{2}{3}\theta\right) \end{bmatrix} \cdot \boldsymbol{n} \qquad \forall\,(r,\theta) \in \Gamma_N. \tag{8.9}$$

Here, Γ_D and Γ_N define the Dirichlet and Neumann boundaries, respectively, such that

$$\Gamma_D \cup \Gamma_N = \partial\Omega \quad \text{and} \quad \Gamma_D \cap \Gamma_N = \emptyset.$$

As shown in e.g. [Demkowicz, 2007], the analytical solution ϕ can be given in polar coordinates r and θ as

$$\phi(r,\theta) = r^{\lambda}\sin\left(\frac{2}{3}\theta\right) \quad \forall\,(r,\theta) \in \Omega, \quad \text{with } \lambda = 2/3. \tag{8.10}$$



(a) L-shaped domain (b) Analytical solution ϕ (c) Gradient magnitude $q = ||\nabla\phi||$

Figure 8.6: Setup and solution of L-shaped domain problem [Zander et al., 2015].

8.2.1.2 Numerical results[f]

To approximate the analytical temperature distribution ϕ numerically, the depicted domain is discretized with an initial element size $h = 1/2$. This base mesh is refined towards the re-entrant corner using the hierarchical *hp-d-* and the multi-level *hp*-approaches.

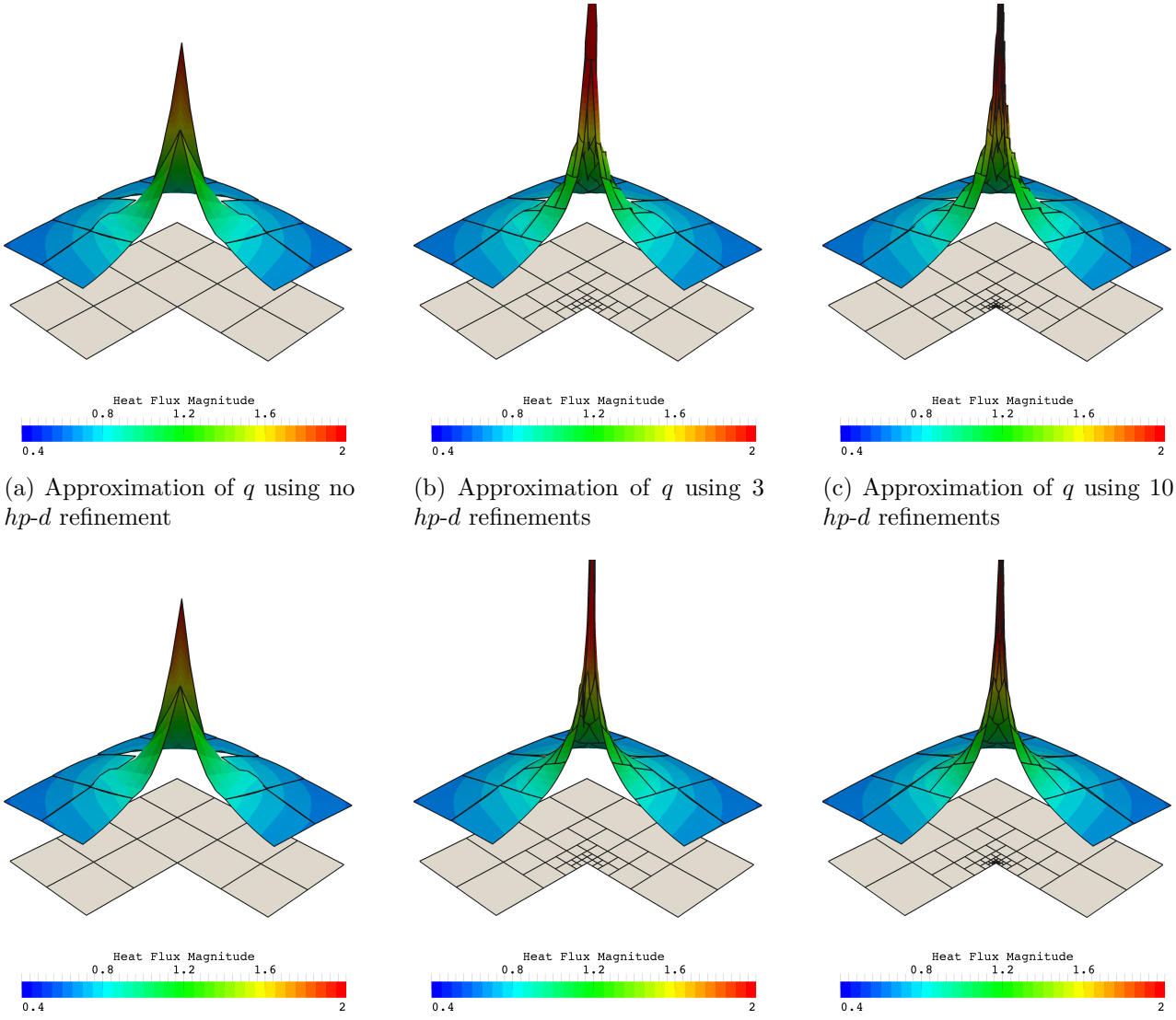
The numerical approximation of the analytical gradient magnitude $q = ||\nabla\phi||$ obtained with the *hp-d-* and multi-level *hp*-approaches are depicted in Figure 8.7. The comparison to the analytical solution in Figure 8.6 shows that the overall solution characteristics are captured correctly by both strategies. In the case of the *hp-d*-method, however, the elements near the singularity suffer from severe discontinuities in the flux. As shown in e.g. [Rank and Zienkiewicz, 1987], these discontinuities directly correspond to the discretization error. Increasing the number of recursive refinements reduces this defect, as the effects of the singularity are bound to the higher refinement levels. But even with ten recursive refinements, the numerical solution still shows large discontinuities in the flux field. In contrast, the results obtained by the multi-level *hp*-approach approximate the singularity significantly better with only minor discontinuities in the derivative field, which qualitatively shows the advantage of this new approach.

8.2.1.3 Convergence comparison

For a quantitative comparison of the result quality, the convergence characteristics are analyzed. For this purpose, the performance of four different refinement strategies are compared in the following.

Strategy 1: uniform p-elevation with fixed number of linear overlay meshes

In first refinement strategy, a p-convergence study is performed on a geometrically graded mesh. To this end, the elements at the entrant corner are refined k-times following the *hp-d-*method. On this graded mesh, the function space is enlarged by increasing the polynomial order of the employed shape functions while keeping the element size unchanged. In the context of an *hp-d*-refinement, this only effects the order of the base mesh whereas the overlay meshes remain linear as illustrated in Figure 8.8a.



(a) Approximation of q using no *hp-d* refinement

(b) Approximation of q using 3 *hp-d* refinements

(c) Approximation of q using 10 *hp-d* refinements

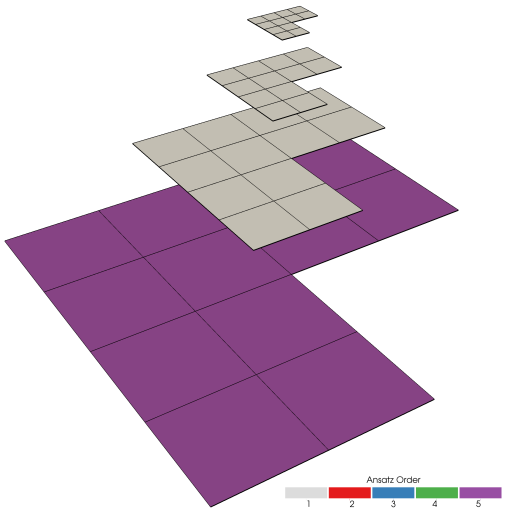
(d) Approximation of q using no multi-level *hp*-refinements

(e) Approximation of q using 3 multi-level *hp*-refinements

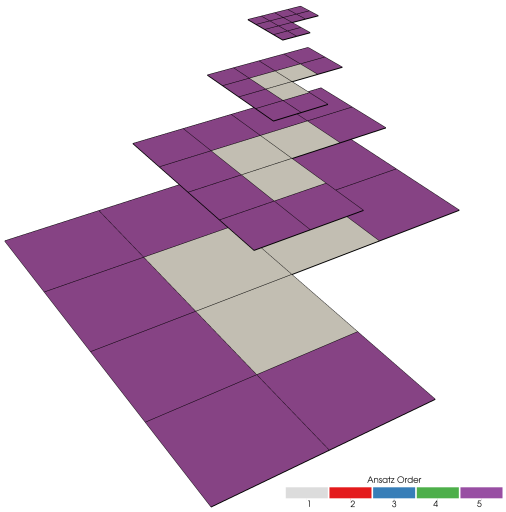
(f) Approximation of q using 10 multi-level *hp*-refinements

Figure 8.7: Numerical approximation of the flux q obtained with $p = 2$ [Zander et al., 2015].

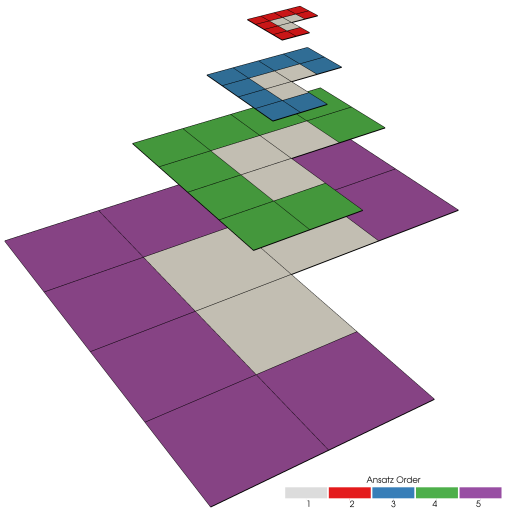
The decrease of the relative discretization error in the energy norm (8.1b) obtained by this type of refinement is depicted in Figure 8.9. As discussed e.g. in [Guo and Babuška, 1986a; Szabó and Babuška, 1991; Szabó et al., 2004], *algebraic* convergence with a rate $\beta = \lambda = \frac{2}{3}$ is to be expected when a p-elevation is applied to the singular problem under consideration. As shown in Figure 8.9, these theoretical values are met when no refinement is employed (i.e. $k = 0$). If the p-elevation is performed on a mesh refined by the *hp-d*-scheme, the first two to three p-refinement steps yield a significant improvement of the numerical error. For the remaining steps, however, the convergence rates significantly reduce towards the theoretical value of $\beta = 2/3$ and asymptotically, an algebraic convergence is observed (see Figure 8.9a). In particular, neither the absolute error value nor the convergence characteristics are affected



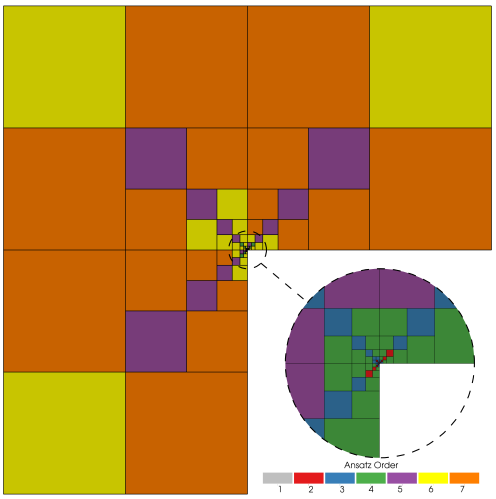
(a) *hp-d*-method



(b) Uniform distribution in multi-level *hp*-method



(c) Linear distribution in multi-level *hp*-method



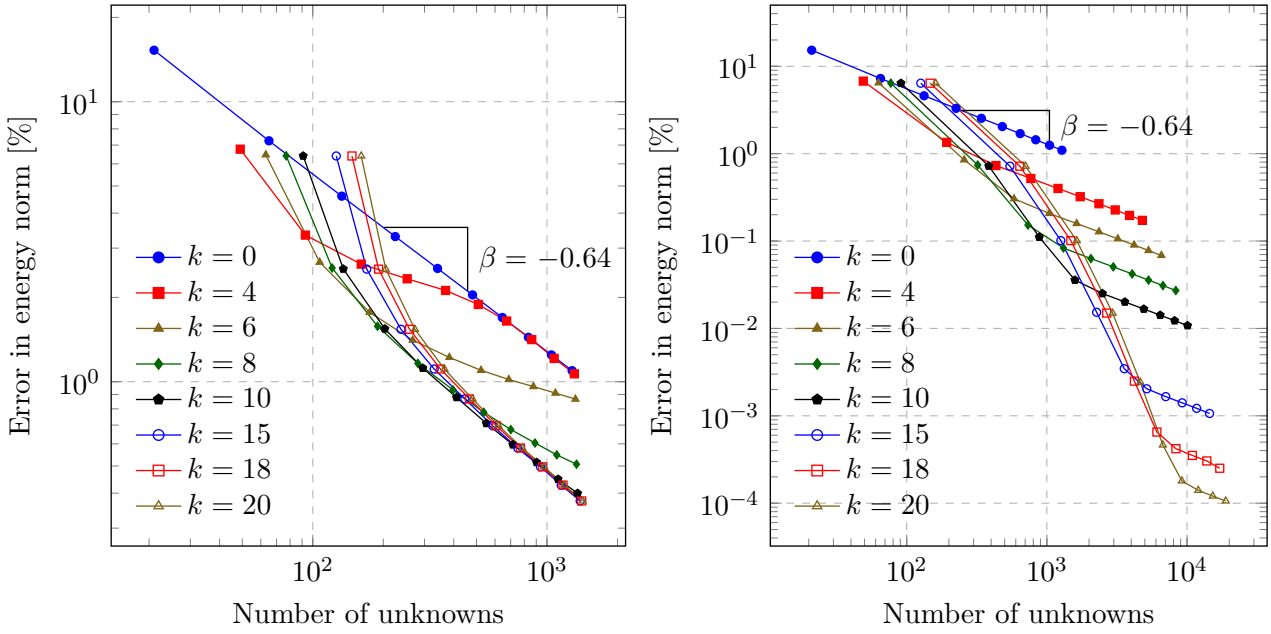
(d) Adaptive multi-level *hp*-FEM after 85 refinement steps

Figure 8.8: Comparison of p-distribution of the *hp-d-* and multi-level *hp*-method.

significantly by the employed number of mesh refinements. These results confirm the findings of e.g. [Schillinger et al., 2012b].[f]

Strategy 2: uniform p-elevation with fixed number of high-order overlay meshes

In analogy to the first refinement strategy, this second approach performs a uniform order elevation on a mesh that pre-graded geometrically towards the re-entrant corner. However, in contrast to the first approach, the discretization is refined in the multi-level *hp*-sense using high-order overlay meshes. This is illustrated exemplarily in Figure 8.8b.



(a) Convergence of *hp-d*-method

(b) Convergence of multi-level *hp*-method

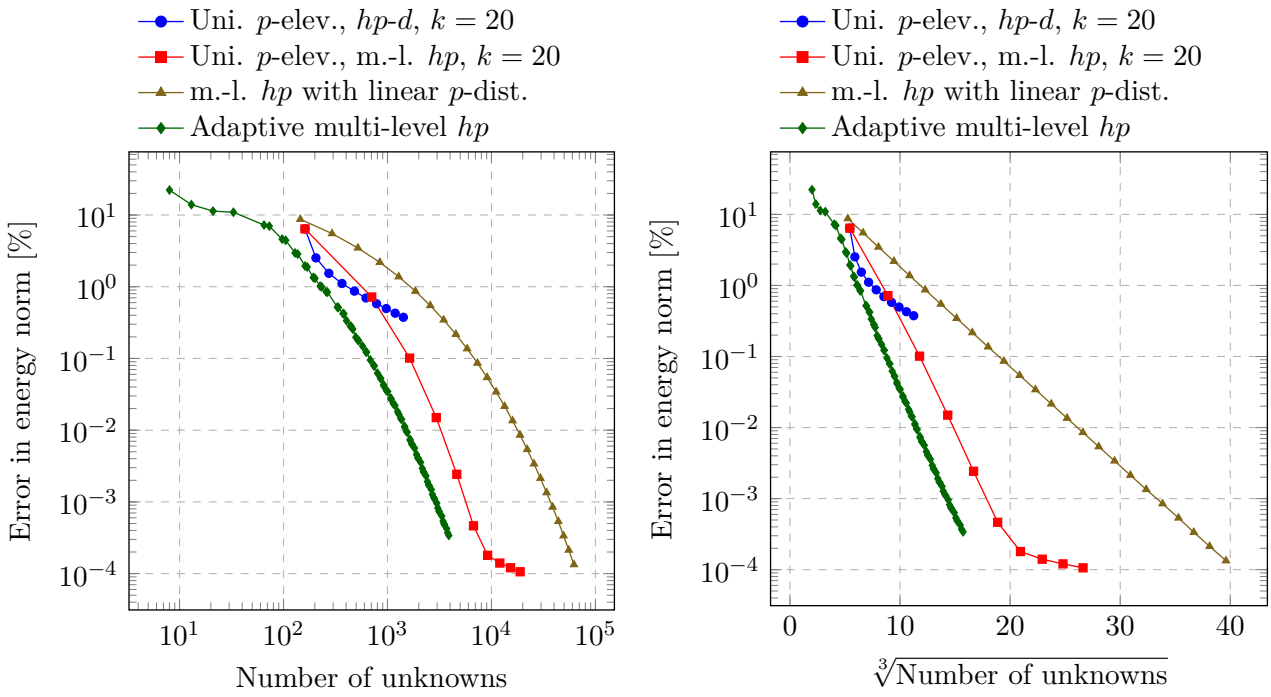
Figure 8.9: *p*-convergence results on a geometrically graded mesh (grading factor $q = 0.5$) created with the *hp-d-* and the multi-level *hp*-method using a uniform *p*-distribution $p = 1, 2, \ldots, 10$. Used abbreviations: k: number of refinements [Zander et al., 2015].

The results presented in Figure 8.9b demonstrate that grading the mesh with the multi-level *hp*-refinement changes the convergence characteristics substantially. Although the asymptotic convergence is still algebraic, the convergence rates continuously increase for the first refinement steps if the mesh grading is strong enough. Only when the polynomial order of the shape functions exceeds the optimal value for the current mesh, the convergence rates decrease and tend towards the expected algebraic value of $\beta = 2/3$ in the asymptotic range. In contrast to the previously presented results, however, the comparison of different refinement levels demonstrates that this transition point can be shifted significantly to higher accuracy if more refinement levels are employed. This allows reaching the desired accuracy with a high convergence rate. The direct comparison of the two methods, depicted in Figure 8.10a, demonstrates that the multi-level *hp*-scheme improves the approximation accuracy by several orders of magnitude compared to the *hp-d*-approach.[f]

For a more detailed analysis of the pre-asymptotic convergence, the representation is changed into a log-$\sqrt[3]{}$ scaling in Figure 8.10b. In this way, the error decay achieved on a multi-level *hp*-graded mesh appears linear until the asymptotic range is entered. Hence, the pre-asymptotic convergence can be identified as being exponential in the form

$$||e||_E \leq C \exp\left(\gamma N^{\theta}\right) \quad \text{with } \theta = 1/3. \tag{8.11}$$

This matches the theoretical expectations of an *hp*-approximation in the presence of a singularity given in Section 3.3 and e.g. [Szabó et al., 2004; Yosibash, 2012].



(a) Convergence in log-log scale

(b) Convergence in semi-log scale

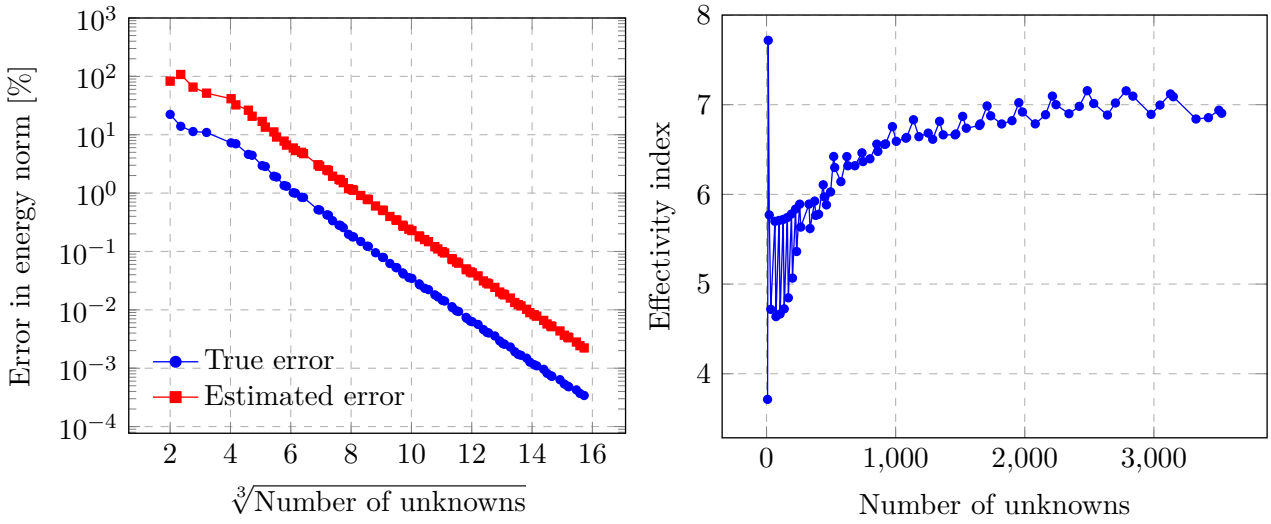
Figure 8.10: Direct comparison of the different refinement strategies [D'Angella et al., 2016; Zander et al., 2015]. Used abbreviations: uni.: uniform, elev.: elevation, m.-l. *hp*: multi-level *hp*. dist.: distribution

Strategy 3: simultaneous change of h and p using a linear p-distribution

The convergence characteristic of the second refinement strategy turns algebraic as soon as the error contribution of the elements at the singularity dominate the overall error. Hence, an exponential convergence in the asymptotic sense demands for a simultaneous elevation of the shape function order p and a graded refinement of the element size h. For this reason, the following second refinement strategy increases the polynomial degree linearly over the different refinement levels as illustrated in Figure 8.8c. In this configuration, the fine elements near the singularity are of lower order, whereas the large elements approximate the solution with high order shape functions. The approximation space is increased subsequently following the *hp*-idea by refining the innermost elements and elevating the order of polynomial shape functions at the same time. The decrease of the error obtained by this second refinement strategy is depicted in Figure 8.10. The results demonstrate that this *hp*-mesh family constructed by multi-level *hp*-method achieves exponential convergence also in the asymptotic sense.

Strategy 4: adaptive multi-level *hp*-refinement using automatic error control[d]

It is noteworthy that the uniform order elevation on a graded mesh yield significantly higher convergence rates in the pre-asymptotic phase. This suggests that the chosen linear p-distribution—although yielding exponential convergence—is not the optimal distribution. To



(a) Convergence

(b) Convergence in semi-log scale

Figure 8.11: Comparison of true and estimated error following [D'Angella et al., 2016].

find a better configuration of the discretization, the third refinement strategy uses the error-controlled feedback scheme described in Section 7.3 to guide the refinement. The thresholds for the two indicators are chosen heuristically as $C_E = 0.8$ and $C_S = 1.75$, respectively.

The obtained convergence results presented in Figure 8.10 confirm the expectations: similar to the previous refinement strategies, the convergence is exponential in form of (8.11). But the absolute error value is significantly better than the results obtained by the two *a priori* refinement strategies. One possible reason for the better performance of the automatic scheme can be derived from the distribution of the polynomial degree depicted in Figure 8.8d. The distribution obtained after performing 85 automatic refinement steps suggests that an heterogeneous p-distribution with a lower polynomial degree in the diagonal elements improves the result quality. This seems reasonable as the approximation properties of the used tensor space is higher over the diagonal of the element than along its principle axis. Thus, a lower polynomial degree here balances the error contribution of the different elements yielding a even higher exponential convergence rate.

The comparison depicted in Figure 8.11a demonstrates that the estimation scheme overestimates the true error. This deviation is to be expected since the used explicit residual-based error estimator only provides an upper bound up to an—in general unknown—constant. Therefore, it is more important to note that the error is overestimated by a constant factor. This is reflected in the converging effectivity index shown in Figure 8.11b. Hence, the true and the estimated error converge with the same rate. Thus, the feedback-based hp-refinement scheme can be identified as *adaptive* with respect to the *convergence rate measure* according to the definition given in Babuška [1986].

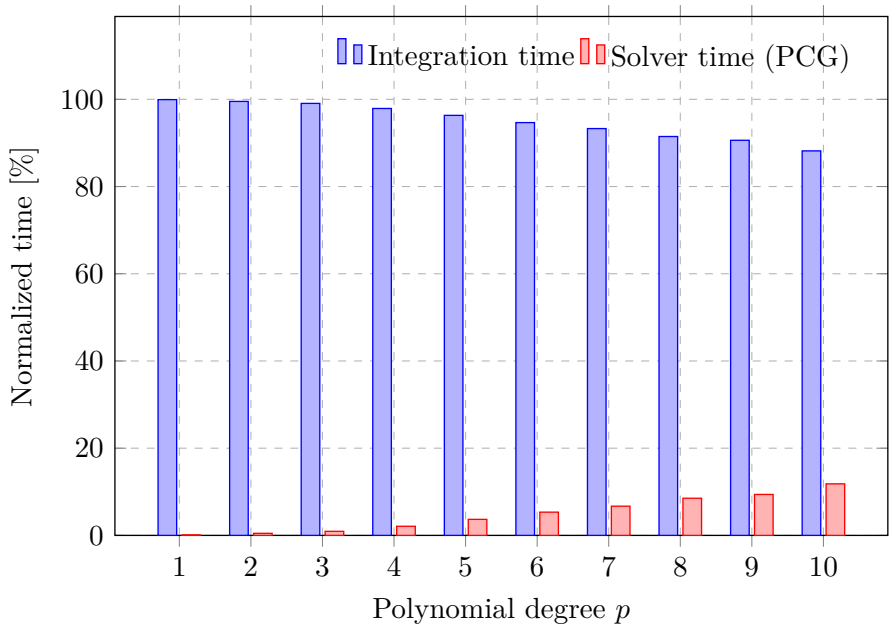


Figure 8.12: Comparison of integration and solver time of the multi-level *hp*-method, $k = 20$.

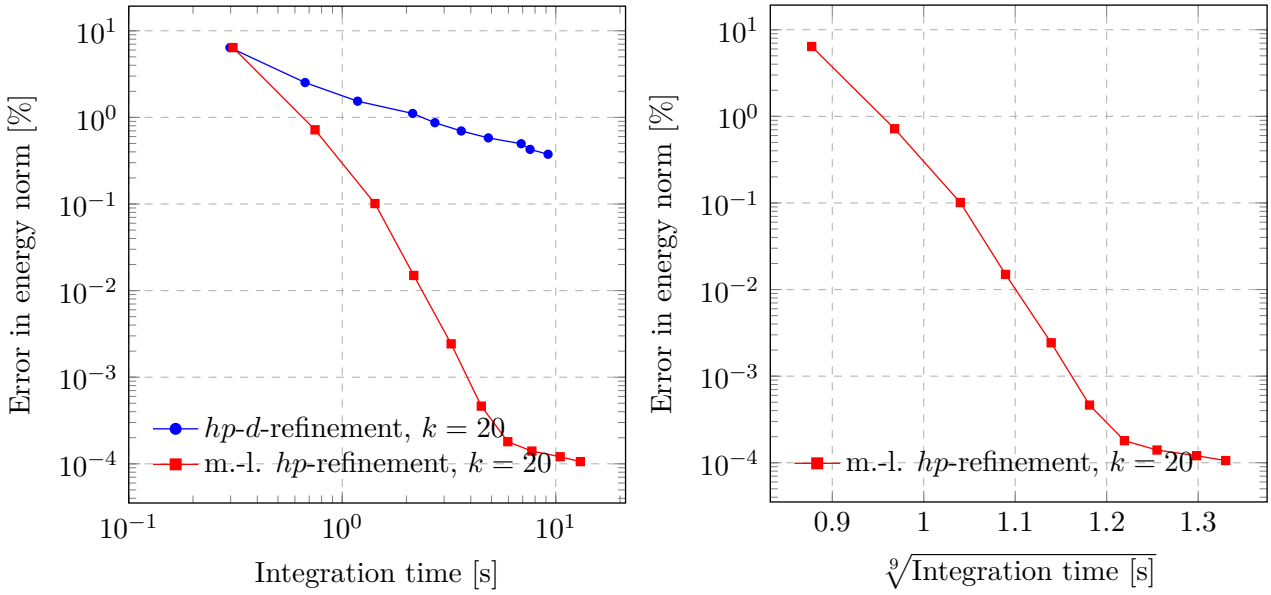
8.2.1.4 Runtime comparison

As discussed in Section 5.2, the superposition approach results in a larger support of the shape functions than a conventional *hp*-approach. This leads to larger element matrices and a denser system of equations with an increased bandwidth. To assess this effect, the convergence of the error with respect to the run-time is compared for the *hp-d*- and the multi-level *hp*-method in the following. For this purpose, the second refinement strategy of the previous study is chosen by elevating the polynomial order of the shape functions uniformly on a pre-graded mesh ($k = 20$).

Due to the approximation quality of higher-order shape functions, high accuracy can be achieved with a comparably small number of unknowns when using the *p*-version of the finite element method. At the same time, however, the number of shape functions per element significantly increases. This increases the time required for integrating the element matrices, such that it typically dominates the time for solving the final system of equations. To analyze this effect, the time required for integrating and the time required for solving the system of equations are compared in Figure 8.12. A conjugate gradient solver with a simple diagonal pre-conditioning is used for solving the system of equations iteratively (see e.g. [Shewchuk, 1994]).[f]

The results show that the time for solving the system of equations is not dominating. Therefore, it can be deduced that the increase of the bandwidth and the reduction of the sparsity of the final equation system—although present—do not have a significant influence on the total computation time.[f]

To assess whether the integration time is in balance with the gain in accuracy, the discretization error in the energy norm is plotted versus the required time for integrating and assembling the element matrices in Figure 8.13. As in the previous studies, this plot shows that the pre-asymptotic and the asymptotic range of the multi-level *hp*-approach can be clearly



(a) log-log-scale (b) semi-log-scale

Figure 8.13: Convergence with respect to the integration time of the multi-level *hp* method with 20 levels of refinements and $p = 1, 2, 3, \ldots$

identified also in terms of the integration time. Within the pre-asymptotic range, the time for integration increases by a factor of approximately 20, while the error reduces by a factor of about 35,000.[f]

This is in significant contrasts to the *hp-d*-strategy, which only converges algebraically and yields an accuracy increase in the magnitude of ten in the same time. In fact, the time required for integration is comparable in both methods. This is reasonable since the number of non-zero (active) shape functions per integration domain is the same in both cases as outlined in Section 5.2. However, the support of the high-order shape functions is smaller in case of multi-level *hp*-FEM. Accordingly, the accuracy gain per integration time is significantly higher in the multi-level *hp*-case.

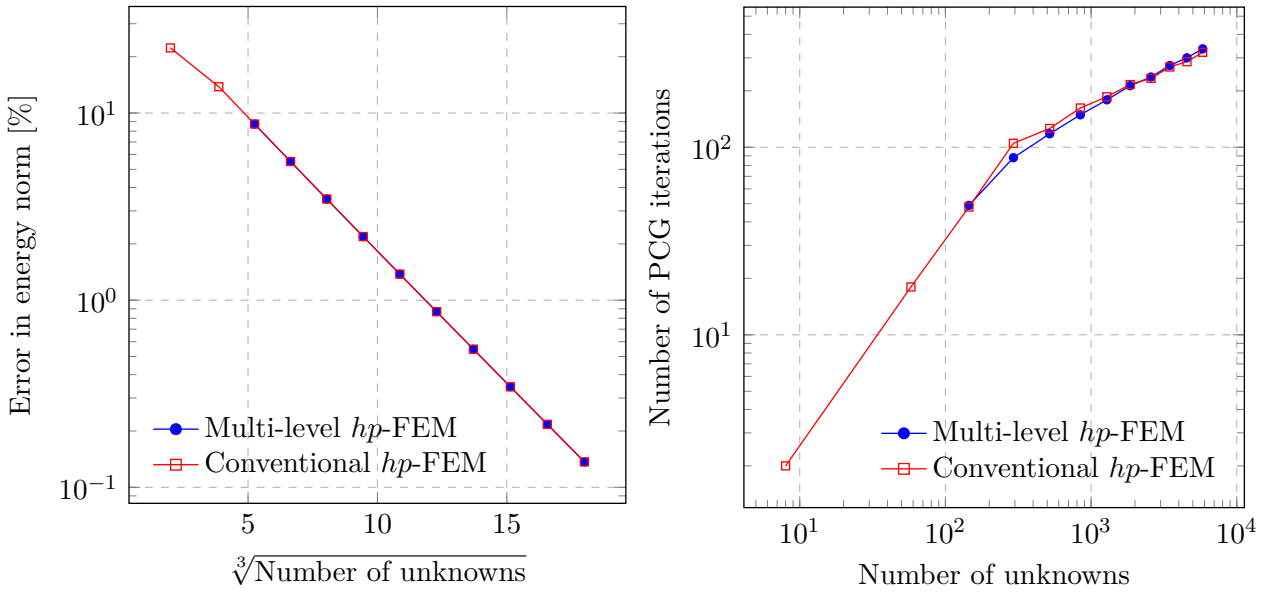
Changing to a semi-log scaling in Figure 8.13b allows identifying the pre-asymptotic convergence as exponential with respect to the 9^{th}-root of the integration time. This is in excellent agreement with the theoretical expectations when considering that the work for integration scales as

$$W = \mathcal{O}(N^3), \tag{8.12}$$

as discussed in e.g. [Melenk et al., 2001]. In combination with the error estimate (8.11), the following bound for the convergence in terms of work can be given:

$$||e||_E \leq D \exp\left(\omega W^{1/9}\right). \tag{8.13}$$

This demonstrates that the multi-level *hp*-method yields the theoretical exponential convergence characteristics for non-smooth problems not only in terms of degrees of freedom but also in terms of the run-time.



(a) Error in energy norm

(b) Number of conjugate gradient iterations diagonal pre-conditioning (PCG)

Figure 8.14: Comparison conventional and multi-level hp-FEM. In both cases, the results are obtained by a simultaneous change of h and p using a linear p-distribution.

8.2.1.5 Comparison to conventional hp-approaches

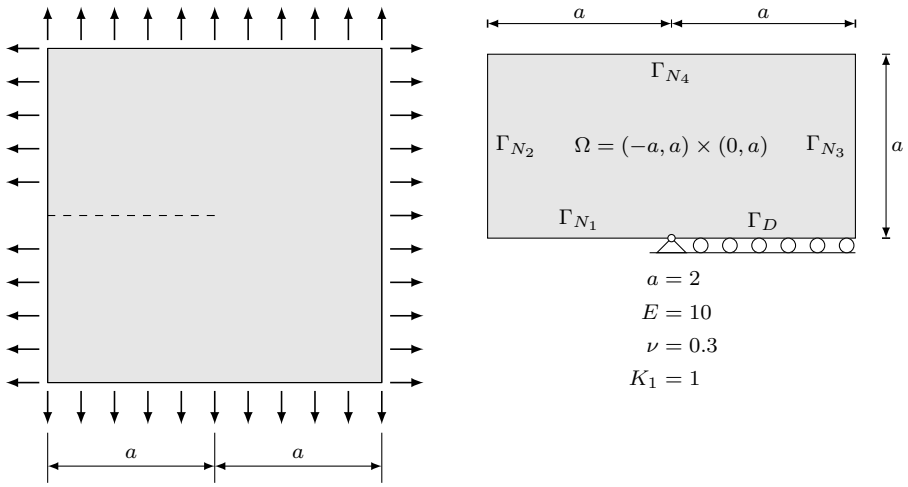
After having compared different refinement strategies in the previous analysis, this section concludes with a comparison of the multi-level hp- and classical hp-approaches of constructing graded meshes. To this end, the aforementioned results obtained by a simultaneous change of h and p using a linear p-distribution are compared to the results obtained by the conventional refine-by-replacement approach presented in [Di Stolfo et al., 2016]. The comparison depicted in Figure 8.14a suggests that—at least for the present L-shaped domain benchmark—both refinement schemes yield the same accuracy.

Besides the approximation accuracy, the support of the shape function also affects the solvability of the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. For this reason, the number of conjugate gradient cycles required to solve the equation system up to a relative residual

$$r_{\text{rel}} = \frac{||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2}{||\boldsymbol{b}||_2} < 10^{-15} \tag{8.14}$$

using a diagonal pre-conditioning are depicted in Figure 8.14b. The comparison reveals that also with respect to the solvability, the two methods show virtually the same quality.

Together with the runtime analysis presented before, these results demonstrate that the increased support of the shape functions has no detrimental effect on the approximation quality or the numerical performance—at least for the benchmark under consideration in this section.



(a) Physical setup (b) Numerical setup

Figure 8.15: Setup of cracked panel (plane strain) [Zander et al., 2017].

8.2.2 Multi-level *hp*-FEM for elastic problems: the cracked panel benchmark[h]

The aim of this second example is to assess the approximation properties of multi-level *hp*-FEM for linear elastic problems with re-entrant corners. To this end, the cracked panel benchmark depicted in Figure 8.15a is considered. Following e.g. [Szabó, 1986], the analytical stress state of a pure mode I deformation under plane strain conditions is given by:

$$\sigma_{xx} = \frac{K_1}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left(1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{8.15a}$$

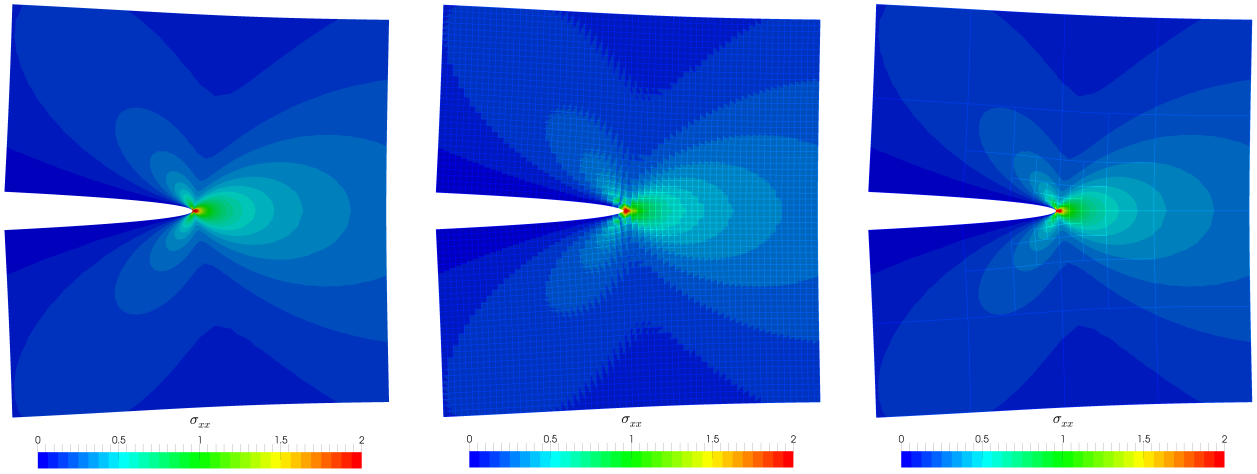
$$\sigma_{yy} = \frac{K_1}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left(1 + \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{8.15b}$$

$$\sigma_{xy} = \frac{K_1}{\sqrt{2\pi r}} \sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2}. \tag{8.15c}$$

In the above equations, (r, θ) are the polar coordinates, and K_1 denotes the stress intensity factor, which is chosen as $K_1 = 1.0$ the present example. The analytical stress distribution σ_{xx} is depicted in Figure 8.16a.

8.2.2.1 Numerical setup

For the numerical approximation of the analytical solution, the variational boundary value problem (2.23) governing linear elastic deformation is solved using the Finite Element Method. The symmetry of the problem is exploited by only considering the upper half of the panel. The respective symmetry Dirichlet boundary conditions are applied as depicted in Figure 8.15b. On the remaining boundary, the analytical stress distributions given in (8.15) are used as



(a) Analytical stress σ_{xx}

(b) Approximation on a uniformly h-refined mesh: 64×32 elements, $p = 1$, 4,290 unknowns

(c) Approximation on a multi-level hp-mesh: $k = 18$, $p = 5$ (uniform), 5,682 unknowns

Figure 8.16: Analytical distribution of σ_{xx} and its approximation on different discretizations (due to symmetry, only the upper half is computed) [Zander et al., 2017].

traction values of the Neumann boundary conditions:

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \mathbf{0} \qquad \forall \boldsymbol{x} \in \Gamma_{N_1}, \quad \boldsymbol{\sigma} \cdot \boldsymbol{n} = -\begin{bmatrix} \sigma_{xx} \\ \sigma_{xy} \end{bmatrix} \quad \forall \boldsymbol{x} \in \Gamma_{N_2}, \tag{8.16a}$$

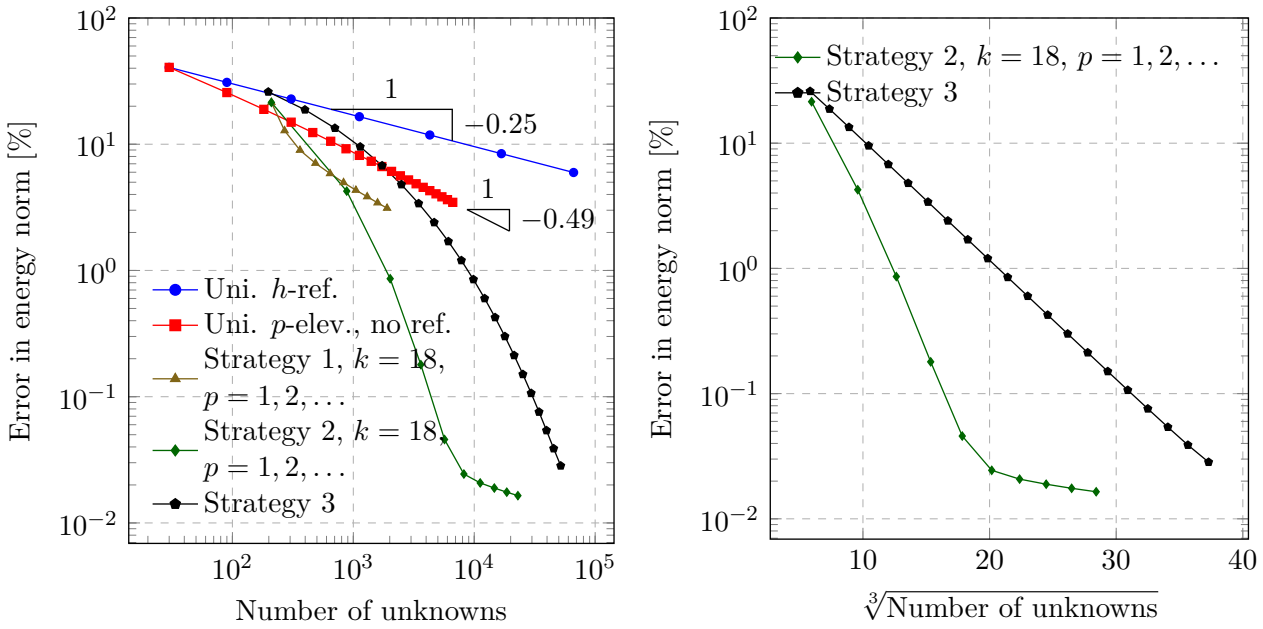
$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = +\begin{bmatrix} \sigma_{xy} \\ \sigma_{yy} \end{bmatrix} \quad \forall \boldsymbol{x} \in \Gamma_{N_4}, \quad \boldsymbol{\sigma} \cdot \boldsymbol{n} = +\begin{bmatrix} \sigma_{xx} \\ \sigma_{xy} \end{bmatrix} \quad \forall \boldsymbol{x} \in \Gamma_{N_3}. \tag{8.16b}$$

The symmetric part of the domain is discretized with a initial mesh of 4×2 elements. To account for the singular stress state at the crack tip, this base discretization is refined geometrically towards the crack tip using hp-d- and multi-level hp-overlay meshes, respectively. The resulting approximation to the analytical stress distribution σ_{xx} is exemplarily depicted in Figure 8.16c.

8.2.2.2 Convergence comparison

As for the previously discussed L-shaped domain problem, the effect of the hp-d- and the multi-level hp-refinement on the approximation quality are compared in the following. Additionally, a uniform h-refinement and a uniform order elevation is considered. The approximation quality is quantified using the error in the energy norm (8.1b) with the following reference value given in [Szabó, 1986]

$$\frac{1}{2} \cdot a(\boldsymbol{u}_{\text{ex}}, \boldsymbol{u}_{\text{ex}}) \approx 0.23706469 \cdot K_1^2 \cdot \frac{a}{E}. \tag{8.17}$$



(a) Comparison in log-log-scale

(b) Comparison in semi-log-scale

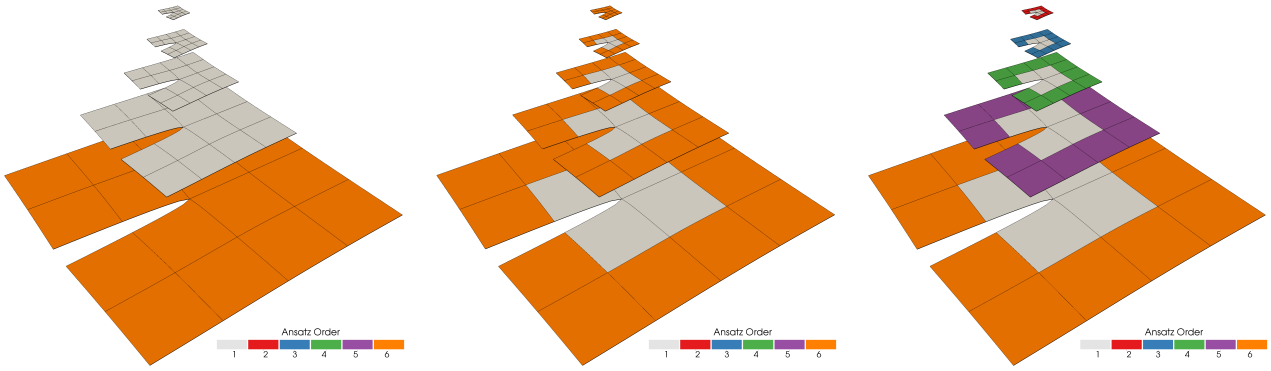
Figure 8.17: Direct comparison of the accuracy gained by different refinement strategies. Used abbreviations: uni.: uniform, ref.: refinement, elev.: elevation, k: number of recursive refinements [Zander et al., 2017].

Uniform h- and p-refinement

The simplest idea to increase the approximation quality is to uniformly decrease the element size h or to elevate the approximation order p of all elements. The error decrease resulting from these extensions are depicted in Figure 8.17a. The results show that a uniform h-refinement yields an algebraic convergence of the approximation error at a rate $\beta = 0.25$. This is in excellent agreement with the *a priori* estimates presented in e.g. [Szabó et al., 2004; Yosibash, 2012], which predict that the convergence rate in the presence of a singularity cannot be higher than $\lambda/2$. The convergence resulting from a uniform p-elevation is also algebraic but at a rate $\beta = 0.49 \approx \lambda$. These results also meet the *a priori* estimates given in e.g. [Szabó et al., 2004; Yosibash, 2012], which predict that a p-refinement achieves twice the convergence rate of a uniform h-refinement. This allows for a significantly higher accuracy using the same number of unknowns. However, both refinement approaches only yield an algebraic decay of the error.

Strategy 1: uniform p-elevation with fixed number of linear overlay meshes

As for the L-shaped benchmark, the first method used for a non-uniform refinement is the hp-d-approach. To this end, a high-order base mesh is refined towards the crack tip by recursively superposing linear overlay meshes. An example of such an hp-d-discretization is depicted in Figure 8.18a. For the convergence curve presented in Figure 8.17a, an 18-times recursively refined mesh is used and the approximation order p is elevated from one to ten. The results show a high convergence of the error for the first refinement steps. However, the

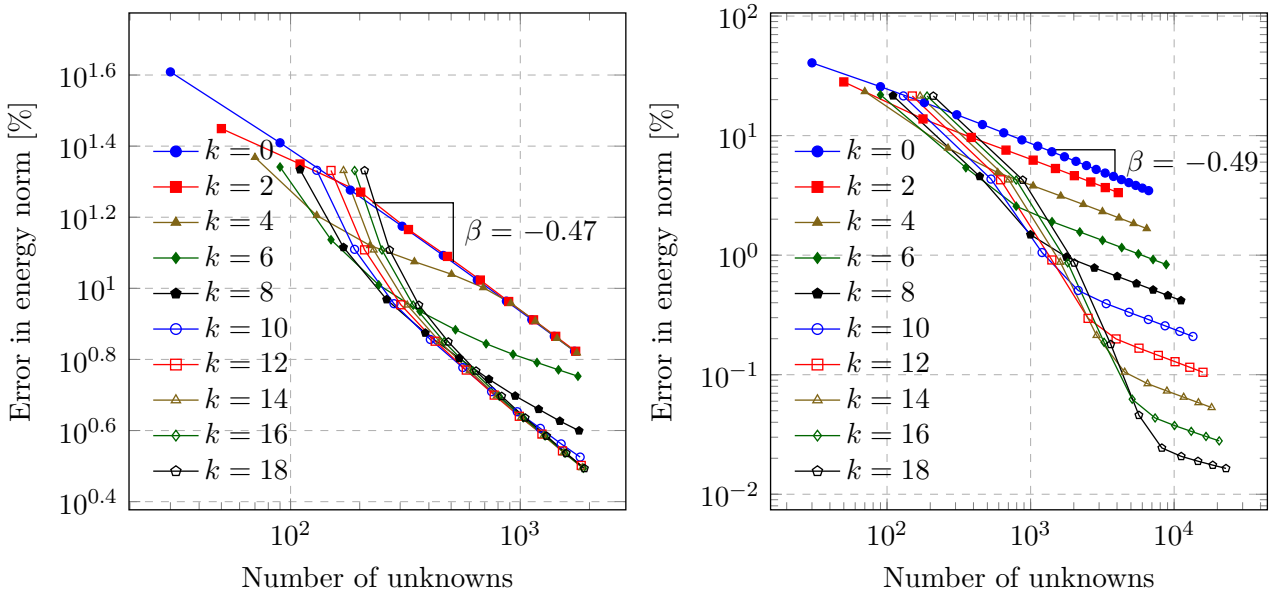


(a) Strategy 1 (*hp-d*-FEM): the order is elevated on the base mesh while the overlay mesh is linear.

(b) Strategy 2: leaf-elements are of high-order. On refined elements, only linear shape functions are active.

(c) Strategy 3: h and p are changed simultaneously and the ansatz order increases linearly away from the singularity.

Figure 8.18: Comparison of p-distribution of the analyzed refinement strategies with four levels of refinement.



(a) Convergence of strategy 1 (*hp-d*-method).

(b) Convergence of strategy 2 (multi-level *hp*-method).

Figure 8.19: p-convergence results on a geometrically graded mesh (grading factor $q = 0.5$) created with the *hp-d*- and the multi-level *hp*-method using a uniform p-distribution $p = 1, 2, 3, \ldots$. Used abbreviations: k: number of refinements.

rates of convergence gradually decay for the later steps, and the convergence turns algebraic in the asymptotic limit with $\beta \approx \lambda$. Furthermore, the levels of refinement k do not significantly influence the error level as shown in Figure 8.19a. This demonstrates that refining the base discretization by linear overlay meshes does not improve the approximation quality substantially.

Strategy 2: uniform p-elevation with fixed number of high-order overlay meshes

The second refinement approach considered is the multi-level *hp*-approach. To this end, the high-order shape functions are moved from the base- to the leaf-elements. This idea is illustrated exemplarily in Figure 8.18b. As in the previous case, an 18-times recursively refined mesh is used and the approximation order p is elevated successively from one to ten. The resulting convergence curve presented in Figure 8.17a demonstrates that the use of high-order overlay meshes has a substantial effect on the convergence characteristic. Although the asymptotic convergence is still algebraic, the convergence rates in the pre-asymptotic range are significantly higher. Only when the polynomial order of the shape functions exceeds the optimal value for the current mesh, the convergence rates decrease and tend towards the expected algebraic value of $\beta \approx \lambda$ in the asymptotic range. However, when using an 18-times refined mesh for the present example, this turning point lies in the range of $||e||_E = 10^{-2}\%$. Therefore, the accuracy increases by three orders of magnitude in the pre-asymptotic range. This gain in accuracy could not be achieved with any of the previous refinement strategies. As illustrated in Figure 8.19b, the accuracy level at which the convergence turns algebraic can be adjusted by the number of mesh refinement levels k.

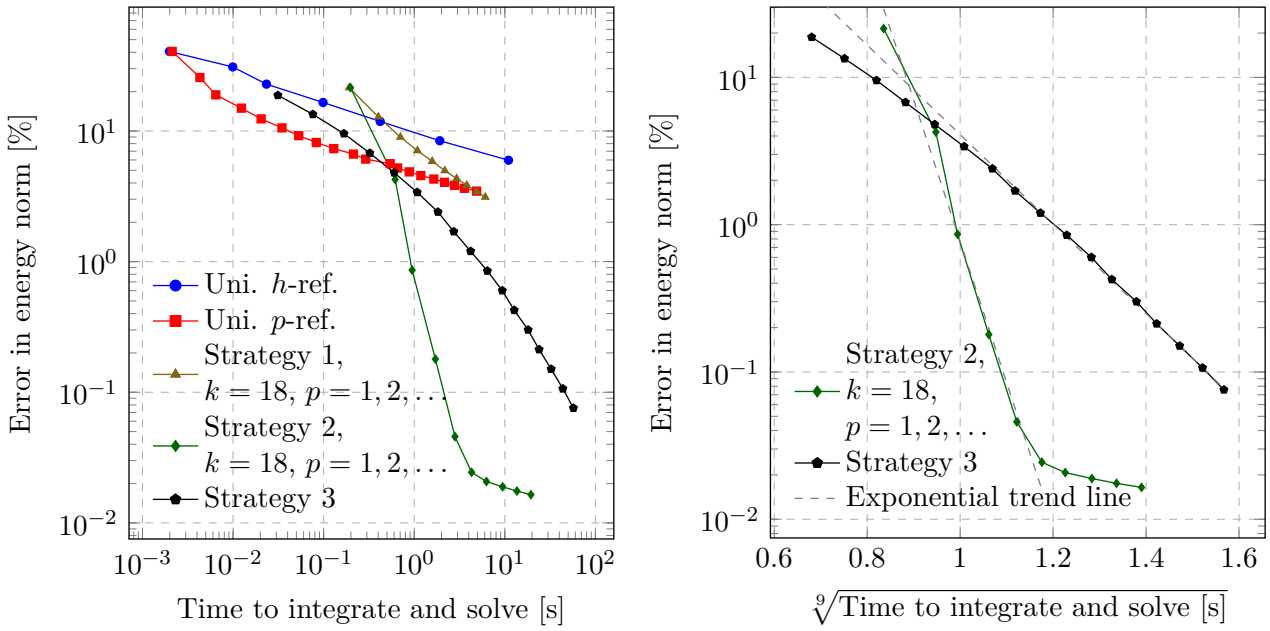
When depicting the results in a log-$\sqrt[3]{}$-scaling in Figure 8.17b, the error decay appears linear until the asymptotic range is entered. This allows to identify the pre-asymptotic convergence as exponential in the form

$$||e||_E \leq C \exp\left(\gamma N^{\theta}\right) \quad \text{with } \theta = 1/3, \tag{8.18}$$

which matches the theoretical expectations of an *hp*-approximation in the presence of a singularity given in Section 3.3 and e.g. [Babuška and Guo, 1988; Schwab, 1998; Szabó et al., 2004; Yosibash, 2012].

Strategy 3: simultaneous change of h and p using a linear p-distribution

In the second refinement strategy, the convergence rate eventually turns algebraic. This happens because the error contribution of the elements at the singularity only decays algebraically [Gui and Babuška, 1986b] and therefore eventually dominates the overall convergence. Hence, an exponential convergence in the asymptotic sense requires a decrease of the finest element size h and a simultaneous elevation of the approximation order p. This is done in the third refinement strategy considered here. Following the standard idea of an *hp*-mesh design presented e.g. in [Gui and Babuška, 1986b; Schwab, 1998], the approximation order p is increased linearly away from the singularity. An example of such a discretization is depicted in Figure 8.18c. For the convergence curve presented in Figure 8.17a, the approximation quality is increased by successively superposing an addition layer of refined elements at the crack tip. Simultaneously, the polynomial order of all other elements is elevated by one. The results depicted in Figure 8.17 show that this refinement approach yields an exponential convergence.



(a) Comparison in log-log-scale

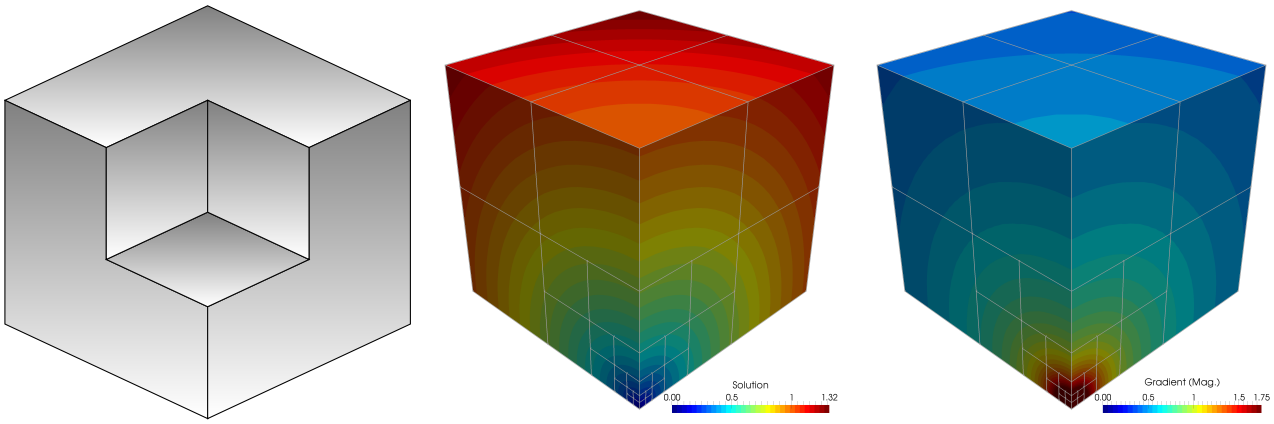
(b) Comparison in semi-log-scale

Figure 8.20: Convergence of approximation error with respect to the time needed to integrate and solve the finite element system. The runtime measurements were performed on a desktop workstation using a single core of an Intel® Core™ i7-4790 CPU and the same finite element code framework. k: number of recursive refinements [Zander et al., 2017].

8.2.2.3 Runtime comparison

The comparison closes with analyzing the decay of the approximation error with respect to the computational time. To this end, Figure 8.20a depicts the error against the time needed to integrate, assemble and solve the finite element equation system. A conjugate gradient solver with a simple diagonal pre-conditioning is used [Shewchuk, 1994]. The results show that the super-algebraic convergence character achieved by the multi-level *hp*-refinement also carries over to the computational performance. In contrast, the asymptotic convergence achieved by the *hp-d*-refinement strategy remains algebraic also when considering the runtime of the simulation.

As for the L-shaped domain example, the *a priori* estimate (8.13) is used as an upper bound for the approximation error in terms of the workload. To compare the numerical performance of the refinement methods to this theoretical estimate, the runtime-results are depicted in a log-$\sqrt[9]{t}$ scale in Figure 8.20b. In this scaling, the pre-asymptotic convergence of both multi-level *hp*-refinement strategies appear virtually linear. Therefore, these can be identified as exponential with respect to the 9^{th}-root of the run time, which is in excellent agreement with estimate (8.13). This demonstrates that the exponential convergence character of the multi-level *hp*-method translates to computational effort in the theoretical correct way also for linear elastic problems.



(a) Fichera corner problem (b) Solution on one octant (c) Gradient on one octant

Figure 8.21: Numerical approximation of the solution and its gradient using four levels of multi-level *hp*-refinement and $p = 6$ (uniform) [Zander et al., 2016].

8.2.3 Fichera corner singularity[b]

The aim of the next example is to analyze the convergence properties of the multi-level *hp*-approach in the presence of a re-entrant corner in a three-dimensional setting. For this purpose, the Fichera corner problem depicted in Figure 8.21a is considered.

8.2.3.1 Problem definition[g]

Following e.g. [Beilina et al., 2005; Kus, 2011; Kus et al., 2014; Schillinger et al., 2014], the Laplace problem is defined as

$$\Delta u = -\lambda \cdot (\lambda + 1) \cdot r^{\lambda - 2} \quad \forall \boldsymbol{x} \in \Omega \tag{8.19a}$$

$$\nabla u \cdot \boldsymbol{n} = \lambda \cdot r^{\lambda - 2} \cdot \boldsymbol{x} \cdot \boldsymbol{n} \quad \forall \boldsymbol{x} \in \partial\Omega \tag{8.19b}$$

$$u = 0 \quad \boldsymbol{x} = \boldsymbol{0}, \tag{8.19c}$$

with r being the radial coordinate

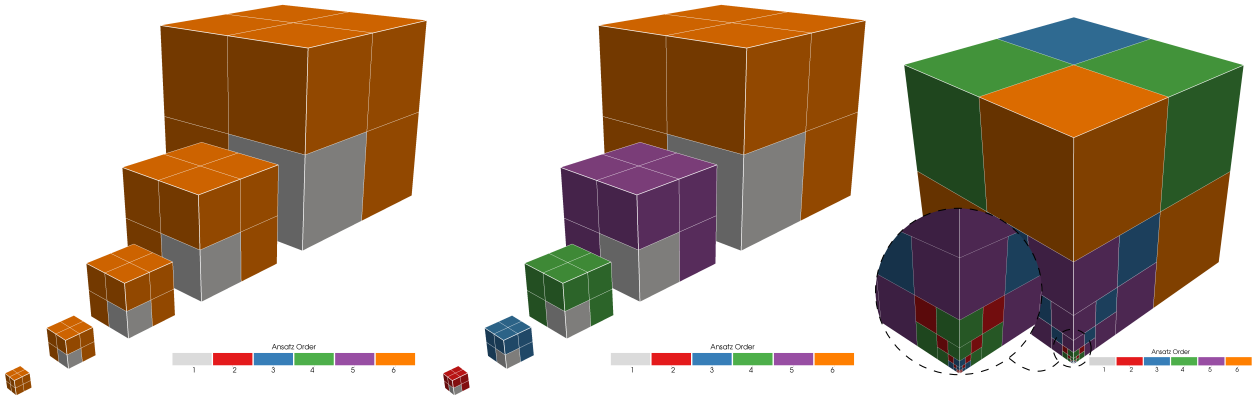
$$r = \sqrt{x^2 + y^2 + z^2}, \tag{8.20}$$

$\boldsymbol{n}$ denoting the outward-pointing normal vector on the boundary and choosing $\lambda = \frac{1}{2}$. The analytical solution to this partial differential equation reads

$$u = r^{\lambda}. \tag{8.21}$$

Due to the symmetry of the solution, only one octant of the Fichera corner domain is considered:

$$\Omega = (0, 1)^3. \tag{8.22}$$



(a) Strategy 1: uniform p-distribution (b) Strategy 2: linear p-distribution (c) Strategy 3: adaptive multi-level hp-FEM

Figure 8.22: Different discretizations of Fichera corner problem [Zander et al., 2016].

8.2.3.2 Convergence comparison

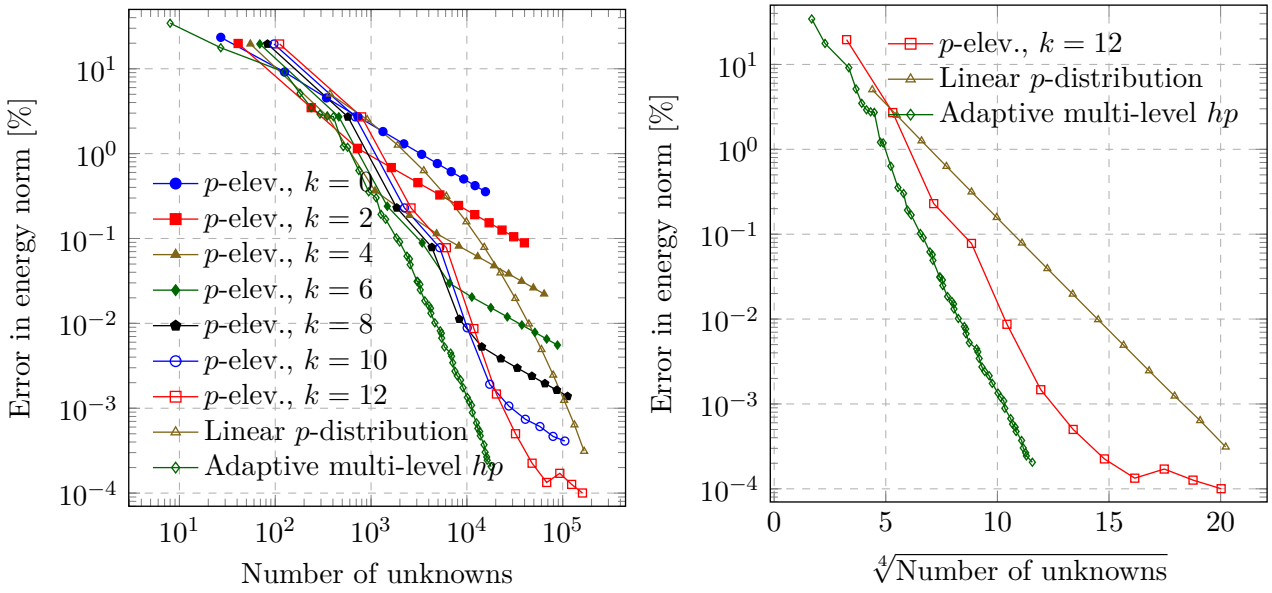
As λ is chosen smaller than unity, the solution is singular at the origin $r = 0$. To assess the approximation quality of the proposed refinement scheme for this vertex-singularity, the error in the energy norm (8.1b) is considered. In the present example, the exact reference value reads[g]

$$\frac{1}{2}a(\boldsymbol{u}_{\text{ex}}, \boldsymbol{u}_{\text{ex}}) = \frac{1}{2}\lambda^2 \int_{\Omega} r^{2\lambda-2} \, \mathrm{d}\Omega \approx 1.48754835248779 \cdot 10^{-1}. \tag{8.23}$$

For the numerical approximation of the solution (8.21), the domain Ω is first discretized using a grid of $2\times2\times2$ elements. This base mesh is refined using the multi-level hp-approach to better capture the singularity. For this purpose, the refinement is steered geometrically towards the origin as depicted in Figure 8.22. The resulting discrete weak form is strongly over-integrated to accurately capture the singular, non-polynomial right hand side in (8.19a). The solution and the gradient computed with this setup are depicted exemplarily in Figure 8.21. Similar to the previously considered cases, the discretization is adapted using three different refinement strategies in the following.[g]

Strategy 1: uniform p-elevation with fixed number of high-order overlay meshes[g]

In the first approach, the solution is approximated using a mesh with a fixed number of refinements. The accuracy is increased by elevating the ansatz order of the leaf-elements *uniformly* as shown in Figures 5.6b and 8.22a. As depicted in Figure 8.23a, pure p-enrichment on the coarse base mesh results in the expected algebraic rate of convergence [Szabó et al., 2004]. This characteristic changes when elevating the ansatz order on a pre-refined mesh. Although the asymptotic convergence is still algebraic, the rates continuously increase over the first refinement steps until the polynomial degree exceeds an optimal value for the given mesh configuration. This S-shaped behavior corresponds to the characteristics of pure p-enrichment on a mesh graded towards a singularity [Szabó and Babuška, 1991]. It is important to note



(a) Convergence in log-log scale

(b) Convergence in semi-log scale

Figure 8.23: Convergence of singular cube example for different refinement configurations using a $2\times2\times2$ base mesh (used abbreviations: p-elev.: uniform order elevation on pre-graded mesh, k: refinement depth) [Zander et al., 2016]

that the turning-point can be shifted to higher accuracy by choosing more refinement levels. This allows to achieve a desired accuracy with a high convergence-rate by choosing the number of refinements accordingly.

By changing the representation into a $\log||e||$-$\sqrt[4]{N}$ scaling in Figure 8.23b, the decay of the error appears linear until the asymptotic range is entered. Hence, the pre-asymptotic convergence can be identified as being exponential in the following form:

$$||e||_E \leq C \exp\left(\gamma N^{\theta}\right) \quad \text{with } \theta = 1/4, \tag{8.24}$$

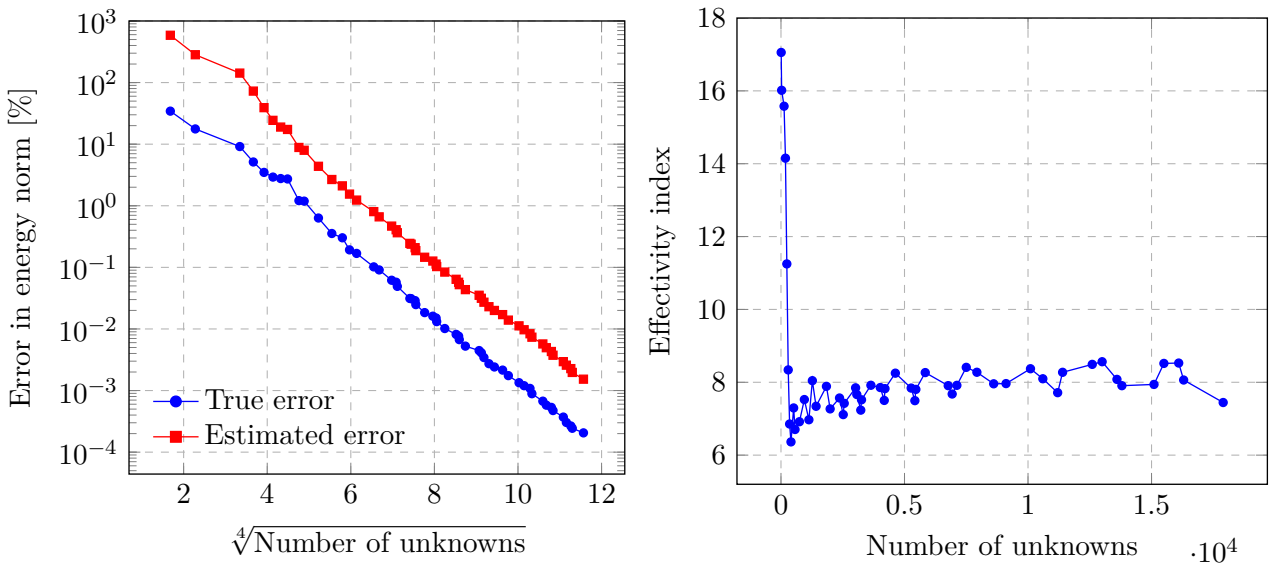
which matches the expectations of an hp-approximation in the presence of a vertex-singularity [Babuška and Guo, 1996; Yosibash, 2012].

Strategy 2: simultaneous change of h and p using a linear p-distribution[g]

In the second refinement approach, both h and p are adjusted simultaneously to obtain exponential convergence also in the asymptotic limit. For this purpose, a linear p-distribution is chosen, where the ansatz order decreases linearly over the refinement levels i:

$$p = i + 1, \quad i = 1, 2, \ldots, k. \tag{8.25}$$

As illustrated in Figure 8.22b, the fine elements near the singular point are of low-order, while the coarse elements use high-order shape functions to approximate the solution where it is smooth. For the convergence graph depicted in Figure 8.23, the accuracy is increased by introducing an extra level of refinement at the singularity, while the ansatz order of all other



(a) Convergence of approximation error

(b) Effectivity index

Figure 8.24: Comparison of true and estimated error.

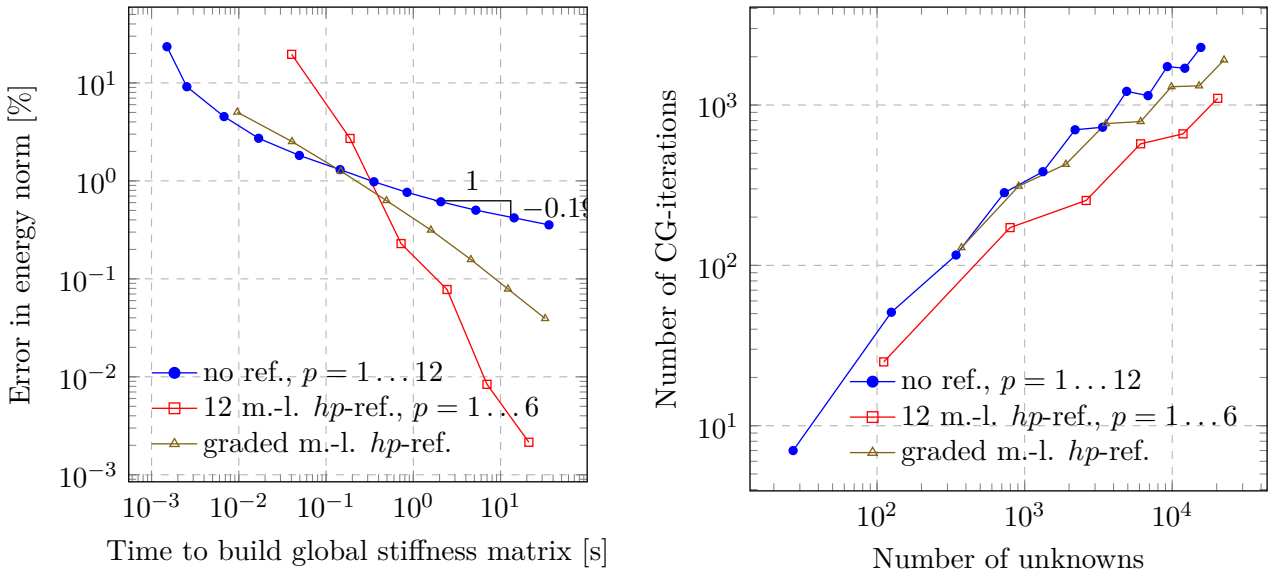
leaf elements is elevated. The resulting error decay depicted in Figure 8.23 demonstrates that the *hp*-mesh family created by this approach yields exponential convergence also in the asymptotic sense.

Strategy 3: adaptive multi-level *hp*-refinement using automatic error control[d]

As in the case of the L-shaped domain, the pre-asymptotic convergence rates obtained by the uniform order elevation are higher than the exponential rates resulting from the linear *p*-distribution. In analogy to the L-shaped domain, this motivates the use of the error-controlled refinement scheme. To this end, the error and smoothness estimators described in Section 7.3 are used to guide the refinement. The thresholds for the two indicators are chosen heuristically as $C_E = 0.8$ and $C_S = 1.75$, respectively.

The comparison presented in Figure 8.23 confirms that the accuracy achieved by the error controlled refinement scheme not only increases exponentially but is also higher than any of the previously considered schemes. The reason seems to be again the in-homogeneous increase of the ansatz order obtained by the adaptive scheme. As illustrated in Figure 8.22c, the polynomial degree in elements across the diagonal are selected lower as in elements along the principal axis.

The comparison depicted in Figure 8.24a demonstrates that the error estimator overestimates the true error. As discussed in the context of the L-shaped domain example, this is due to the—in general unknown—constant of the employed explicit residual-based error estimator. However, as illustrated in Figure 8.24b, the effectivity index converges to a constant value. This implies that the true and the estimated error converge at the same rate. According to the definition given in Babuška [1986], also the three-dimensional *hp*-feedback scheme can thus be identified as *adaptive* with respect to the *convergence rate measure*.



(a) Runtime comparison using a single thread on an Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz.

(b) Number of CG-iterations necessary for solving the FE-equation system up to an accuracy of 10^{-15} using a Jacobi pre-conditioning.

Figure 8.25: Comparison of refinement schemes in terms of computational performance.[Zander et al., 2016]

8.2.3.3 Runtime comparison[g]

The final aspect addressed in this study is the computational efficiency of the different refinement schemes. In particular, the cost of the additional overhead of the hierarchical meshes is of interest. To assess this point, Figure 8.25a depicts the decay of the error in the energy norm versus the time required to integrate and assemble the global stiffness matrix. The time needed for building the right-hand-side vector is not considered as the singular load function demands for a significant over-integration, which would distort the analysis. All measurements are performed using the same in-house code and a single thread on an Intel(R) Core(TM) i7-4790 CPU running at a clock speed of 3.60GHz.

The comparison presented in Figure 8.25a shows that the algebraic convergence character of a uniform order elevation carries over to the computational performance if p is sufficiently high. In contrast, both hp-schemes show an exponential convergence. Although this exponential character is only slightly pronounced, the obtained convergence rate is significantly higher than the rate resulting from a pure p-refinement. For the present example, the three refinement schemes break even at an error level of about 1%. Beyond this point, the hierarchical hp-refinement achieves a higher accuracy than a pure p-elevation for the same computational time.

The second aspect to be considered is the solution of the finite element equation system. To this end, Figure 8.25b compares the number of conjugate gradient iterations needed to solve the respective equation system up to an accuracy of 10^{-15} using a Jacobi pre-conditioning scheme, *cf.* [Shewchuk, 1994]. The depicted results demonstrate that, when using an hp-refinement, the number of necessary CG-iterations is not more than in case of a pure order

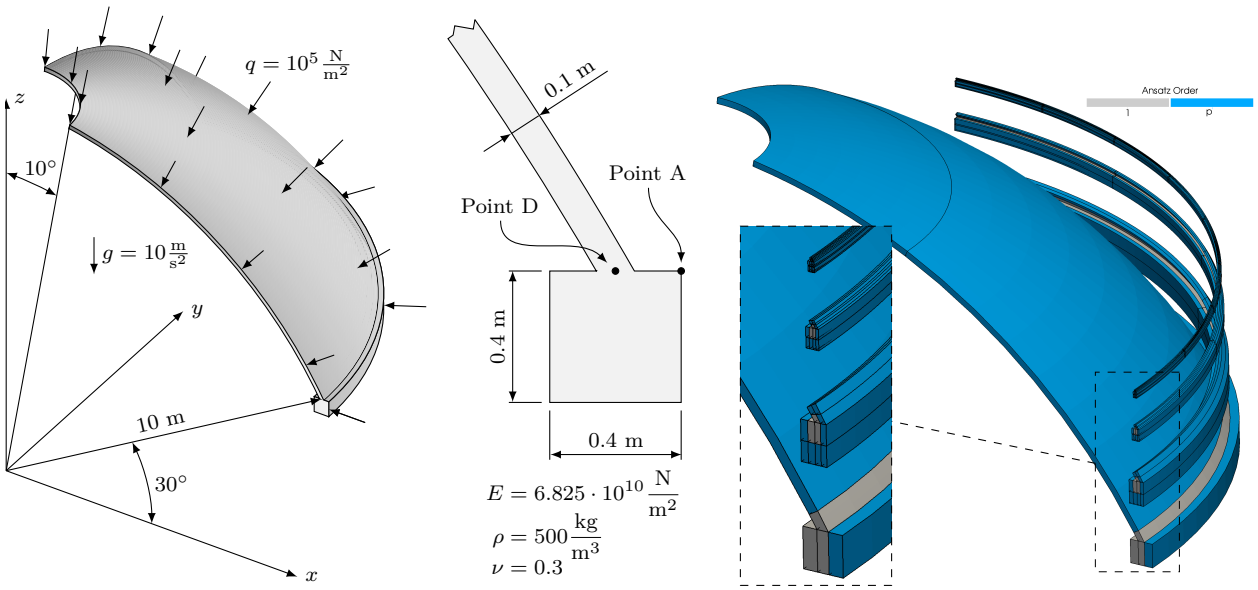
elevation. This indicates that the use of high-order shape functions is dominating the overall convergence characteristic of the CG-scheme, and the use of finer overlay meshes has no negative effect.

8.2.4 Multi-level *hp*-refinement for curved, three-dimensional, elastic structures[g]

The aim of this last example is to study the applicability of the suggested refinement scheme for more complex examples including curved geometries. To this end, the three-dimensional extension of the Girkmann-problem [Girkmann, 1956] introduced in [Rank et al., 2005] is considered. This benchmark was already investigated in the context of shell formulations [Kiendl et al., 2010], isogeometric analysis [Cottrell et al., 2009, 2007; Hughes et al., 2005], T-splines [Bazilevs et al., 2010], and adaptive volumetric shells [Düster et al., 2007b; Scholz et al., 2006].

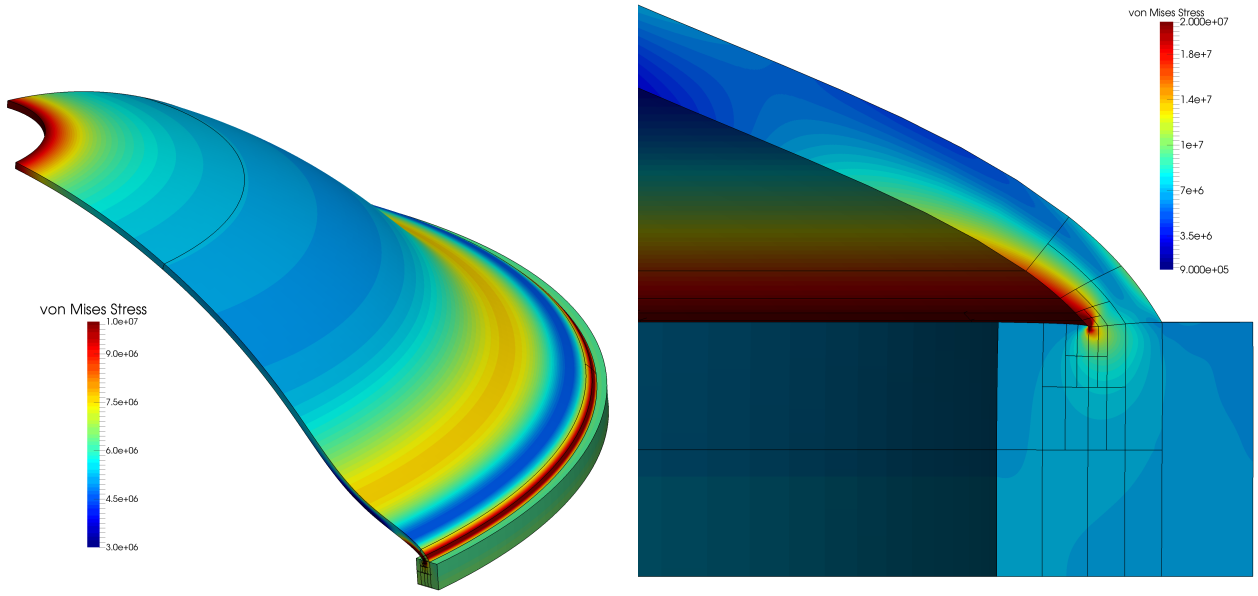
The geometric and mechanical properties are depicted in Figures 8.26a and b. Due to the symmetry of the problem, only one quarter of the structure is considered. Accordingly, the respective displacements normal to the yz- and zx-planes are constrained to zero. The stiffener is simply supported at the bottom surface by constraining the vertical displacement u_z to zero. The structure is loaded by self-weight and a pressure q on the outer surface of the shell and the stiffener.

For the numerical approximation of the resulting deformation, the domain is first discretized using six high-order, hexahedral elements depicted in Figure 8.26c. As shown in [Rank et al., 2005], such a coarse base mesh is capable of capturing the global deformation state accurately. Due to the applied load, the vertical displacement u_z is constant along the latitude. Hence, the use of linear polynomials for the vertical displacement component in this direction is



(a) Problem definition (b) Details of stiffener (c) Refined discretization

Figure 8.26: Hemispherical shell with stiffener [Zander et al., 2016].



(a) Stress distribution on outer shell (b) Zoom-in on connection between shell and stiffener

Figure 8.27: von Mises stress in $\mathrm{N/m^2}$ on deformed structure (scaling factor = 500, $p = 10$) [Zander et al., 2016].

sufficient. For this reason, the displacement is approximated by a trunk tensor space formed by the following polynomial degree template

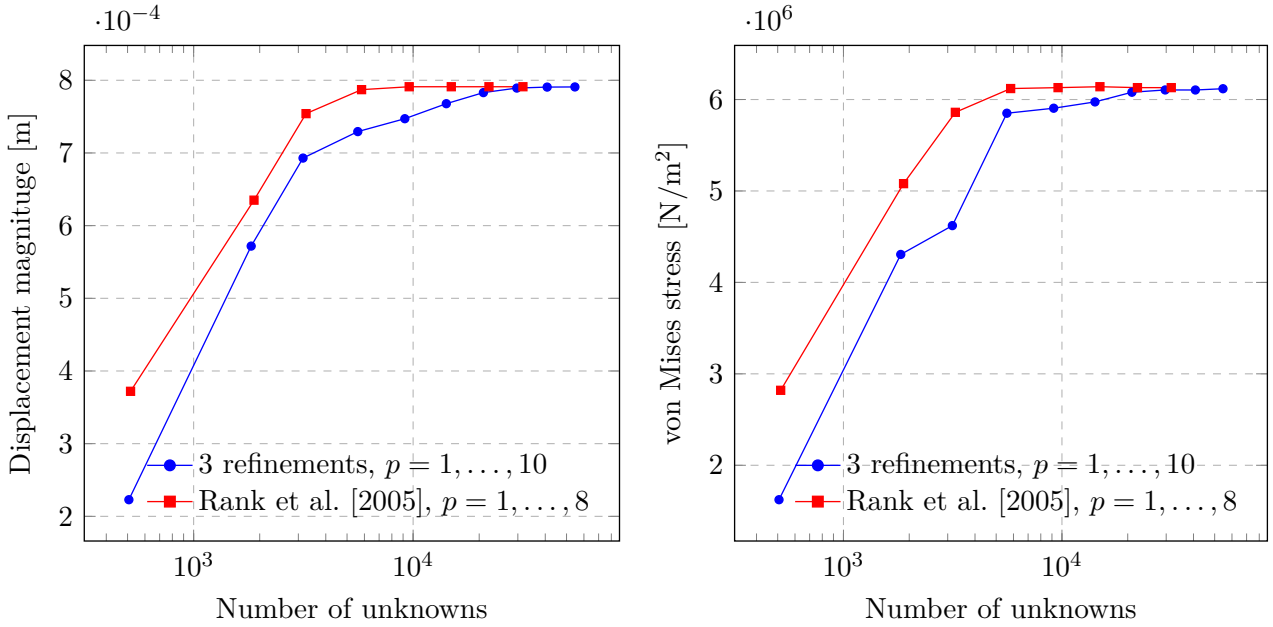
$$\boldsymbol{P} = \begin{array}{c} \\ r \\ s \\ t \end{array} \begin{array}{c} \begin{array}{ccc} u_x & u_y & u_z \end{array} \\ \begin{pmatrix} p & p & p \\ p & p & p \\ p & p & 1 \end{pmatrix} \end{array}. \tag{8.26}$$

Here, p denotes the polynomial degree used for the remaining solution components. The variables (r, s, t) represent the coordinate directions in the reference space of the element, where t corresponds to the latitude direction.

To capture the expected singularity, the base discretization is refined at the connection between the shell and the stiffener by superposing three levels of overlay elements as shown in Figure 8.26c. As the overlay elements are defined in the index space of their corresponding parent elements, the shape of these fine elements naturally follows the curved geometry.

The approximation obtained with the refined discretization is depicted in Figure 8.27. The results show that the coarse base mesh efficiently captures the global deformation and the resulting state of stress. At the same time, the local refinement accurately resolves the stress concentration at the re-entrant corner.

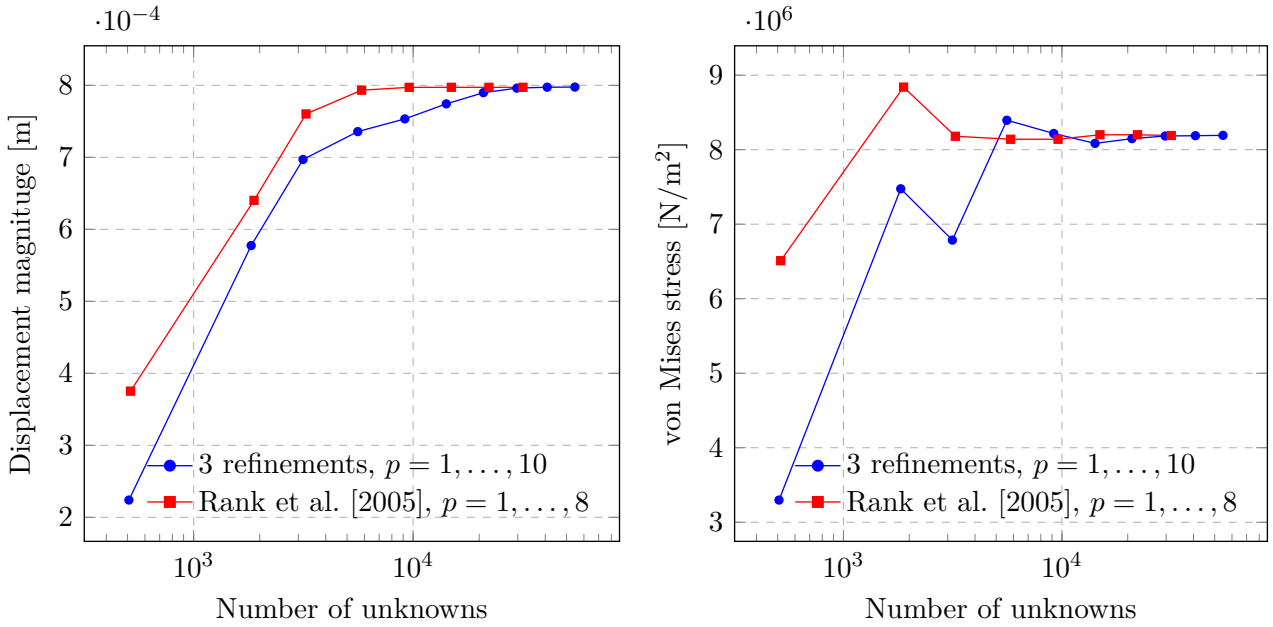
To assess the quality of the numerical approximation in more detail, displacements and stresses are recorded at the points labeled A and D in Figure 8.26b. The results are compared to the reference values presented in [Rank et al., 2005] in Figure 8.28 and 8.29. The comparisons show an excellent agreement of the convergence limits. This demonstrates that



(a) Displacement magnitude at point A

(b) Von Mises stress at point A

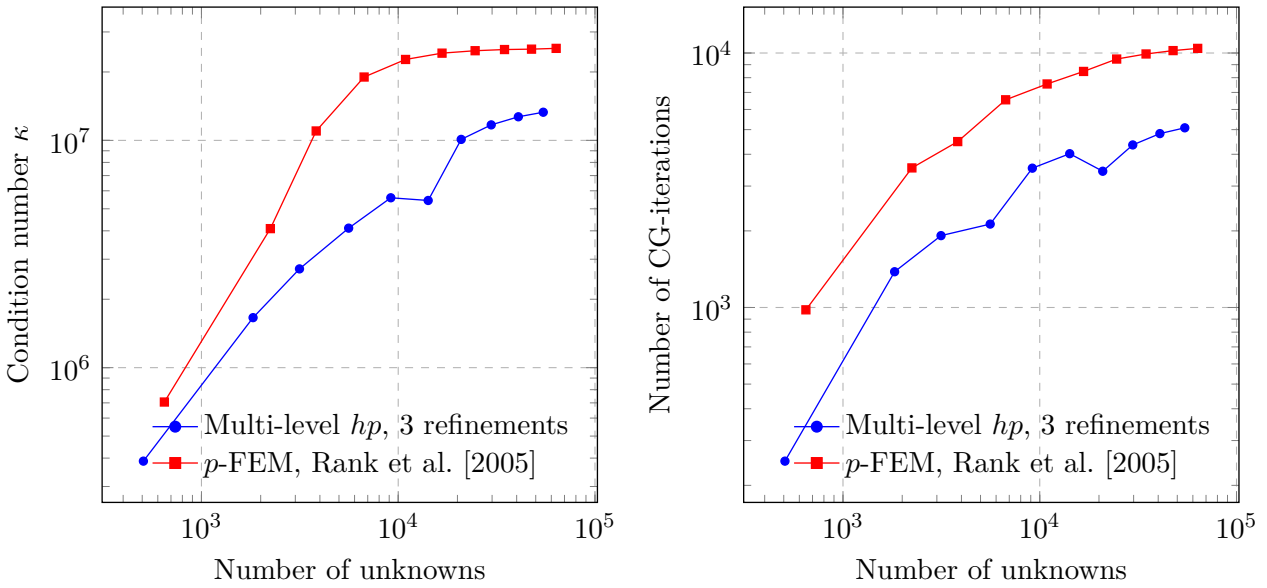
Figure 8.28: Convergence of displacement and stress at point A [Zander et al., 2016].



(a) Displacement magnitude at point D

(b) Von Mises stress at point D

Figure 8.29: Convergence of displacement and stress at point D [Zander et al., 2016].



(a) Estimated condition number computed using the scheme suggested in [Hager, 1984] implemented in MATLAB `condest`.

(b) Number of CG-iterations necessary for solving the FE-equation system up to an accuracy of 10^{-15}.

Figure 8.30: Comparison of computational performance for $p = 1 \dots 10$ after application of a Jacobi pre-conditioner [Zander et al., 2016].

the advantageous approximation properties of the multi-level *hp*-scheme, observed for the simply-shaped benchmarks discussed in the previous sections, also carry over to complex geometries. The solutions obtained by [Rank et al., 2005] show a slightly faster convergence because Rank *et al.* used a problem-tailored mesh that was graded with a geometric progression of 0.15 towards the re-entrant corners. To achieve a similar performance, the presented scheme would need to be extended to also feature anisotropic refinement mimicking general geometric progressing factors.

The final aspect to be considered is the computational performance of the suggested refinement scheme for this benchmark with a curved geometry. To this end, the condition number of the stiffness matrix obtained by the multi-level *hp*-refinement scheme is compared to the reference obtained by a pure *p*-elevation on the geometrically graded mesh used by Rank *et al.* in Figure 8.30a. The depicted results demonstrate that the hierarchical overlay approach yields a condition number that is in the same order of magnitude but slightly lower than the classically graded *p*-mesh. As shown in Figure 8.30b, this advantage is also reflected in the number of conjugate gradient iterations necessary to solve the respective finite element equation systems up to an accuracy of 10^{-15}. It seems likely that the better conditioning of the equation system is a consequence of the orthogonality properties observed for one-dimensional problems in Section 8.1.2.

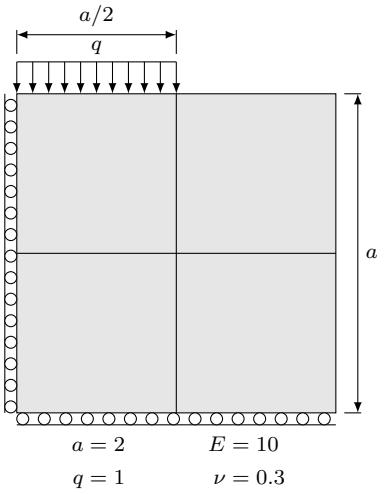


Figure 8.31: Setup of discontinuous traction example with a base discretization of 2×2 elements (plane strain).

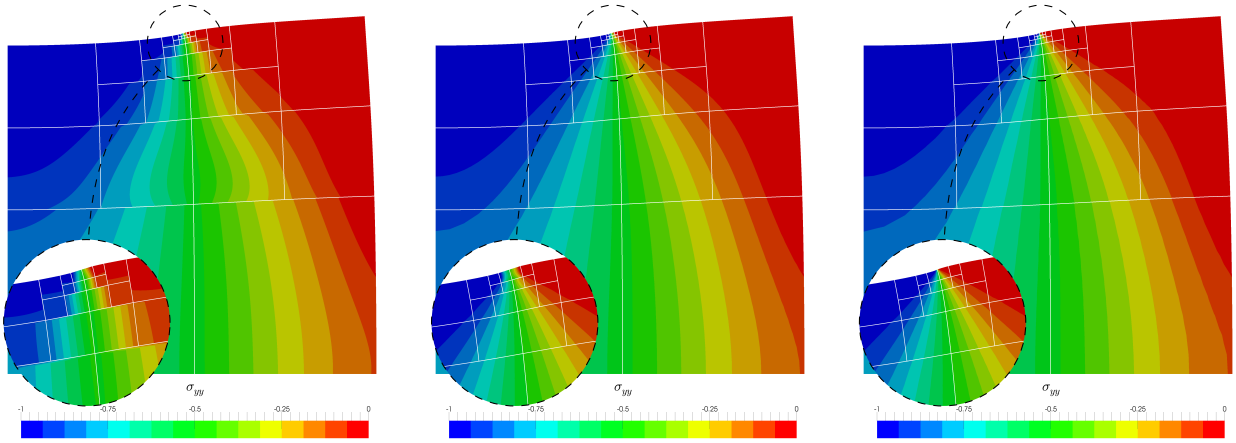
8.3 Singularities due to boundary conditions

The results presented in the previous section demonstrate that the multi-level *hp*-method is well suited for the approximation of stress concentrations and singularities caused by re-entrant corners in the domain. The aim of the present section is to analyze the applicability of the method for situations in which the applied boundary conditions reduce the regularity of the solution. For this purpose, the elastic continuum depicted in Figure 8.31 is considered, which is loaded by a discontinuous traction on the top edge. To approximate the resulting deformation, the variational boundary value problem (2.23) governing the linear elastic response is solved. On the lower and the left side of the domain, symmetry boundary conditions are applied by constraining the displacement normal to the respective edge. The left half of the top edge is loaded by a uniform pressure $q = 1$, whereas homogeneous Neumann boundary conditions are applied on the remaining parts of the boundary.

For the numerical approximation of the deformation resulting from the described loading, the domain is first discretized with 2×2 elements depicted in Figure 8.31. Due to the discontinuity of the traction applied on the top edge of the domain, the vertical stress component σ_{yy} is expected to be discontinuous at the point $(a/2, 2)$. To account for this challenging solution characteristic, the base discretization is refined by superposing an overlay mesh that is geometrically graded towards this point. In analogy to the previously considered benchmarks, the overlay mesh is created following the *hp-d-* or the multi-level *hp*-method, respectively.

In Figure 8.32, the approximations of the vertical stress component σ_{yy} obtained by different discretizations with the same refinement depth are compared. According to the applied boundary conditions, σ_{yy} is expected to be -1 on the left and 0 on the right half of the top edge. This discontinuous characteristics can be captured sharply when using overlay meshes created by the multi-level *hp*-method as demonstrated in Figures 8.32b and c. In contrast, the stress approximation appears piece-wise linear when using an *hp-d*-overlay mesh (see Figures 8.32a).

To qualify this difference in more detail, the numerical stress approximations along the top



(a) *hp-d*-grading, $k = 6$, $p = 4$ uni., 226 dofs

(b) multi-level *hp*-grading, $k = 6$, linear p-distribution, 1,978 dofs

(c) multi-level *hp*-grading, $k = 6$, $p = 4$ uni., 1,282 dofs

Figure 8.32: Approximation of the vertical stress component σ_{yy} obtained by different discretizations.

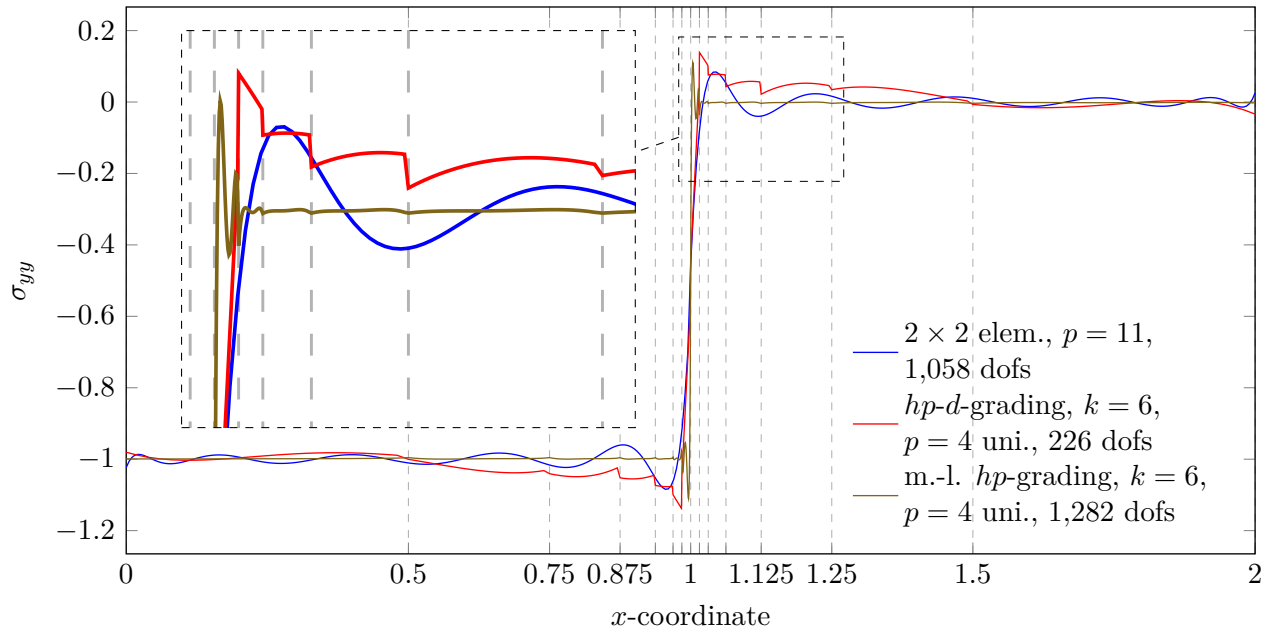


Figure 8.33: Numerical approximation of σ_{yy}-stress component on top edge ($y = a$). The vertical lines correspond to element edges of 6-times refined mesh. Used abbreviations: elem.: element, dofs: degrees of freedom, k: refinement levels, m.-l. *hp*: multi-level *hp*, dist.: distribution.

edge are plotted in Figure 8.33. In the illustration, the divisions along the abscissa correspond to the element edges of the six-times refined mesh. The comparison first demonstrates that a pure order-elevation on the original coarse mesh consisting of 2×2 elements yields an approximation that suffers from severe oscillations. This Gibbs-like phenomenon is well known when using polynomial shape functions for the approximation of discontinuous solutions. Interestingly, however, refining the mesh by the *hp-d*-mesh does not decrease this deficiency. Instead, both, the amplitude *and* the spread of the oscillations remain unchanged. The reason is that the *hp-d*-overlay mesh approximates the solution only with linear shape functions, whereas the high-order modes remain on the original coarse element. Hence, the linear overlay functions only break the continuity on the final approximation, while the support—and thus the approximation power—of the high-order mode remains unchanged. Accordingly, the superposition of the linear overlay mesh does not correspond to a high-order mesh refinement. The situation is therefore similar to the bi-material truss discussed in the motivating example 8.1.1.

This shortcoming of the refinement scheme can even be seen in Figure 8.33: at the edge of the overlay mesh, the approximation of the *hp-d*-refined mesh is discontinuous. However, the underlying high-order approximation appears continuous but only shifted by the constant offset function introduced by the derivative of the linear overlays. This gives a visual explanation for the large spread of the oscillations.

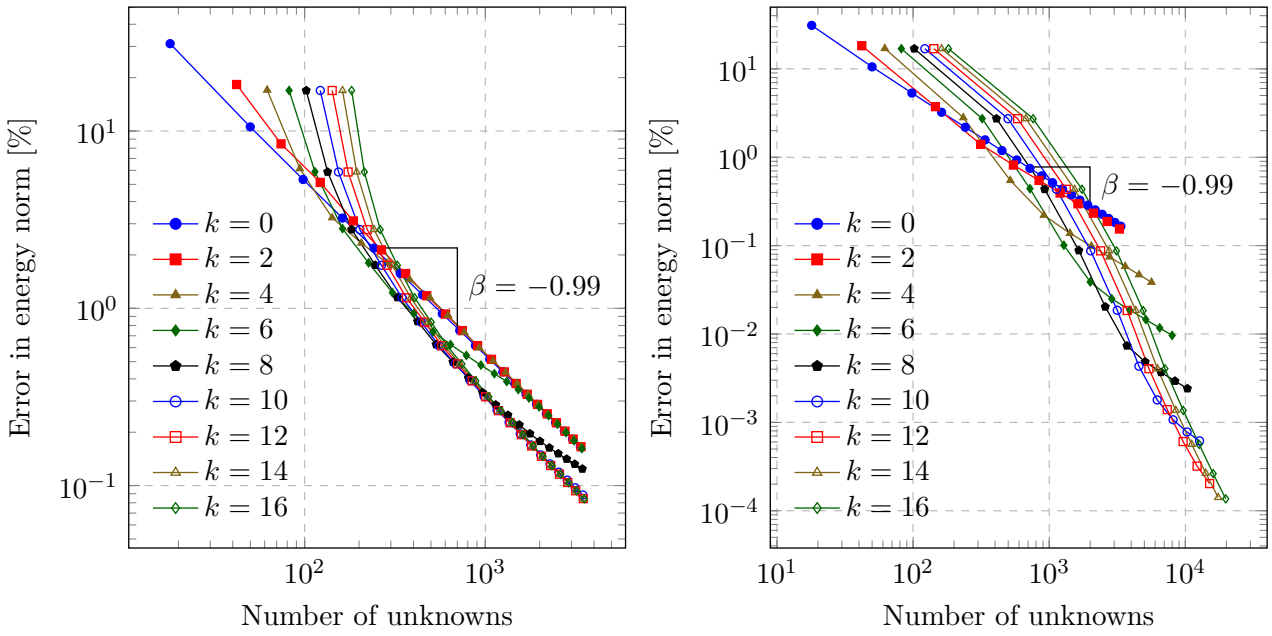
As discussed in Section 8.1.1, the situation changes fundamentally when using high-order shape functions in the multi-level *hp*-overlay mesh. In this case, the support of the high-order modes is reduced to the overlay elements, which increases the approximation properties. The results depicted in Figure 8.33 demonstrate that the multi-level *hp*-refinement limits the oscillations to the finest elements.

When evaluating the approximation accuracy, it has to be taken into account that the employed multi-level *hp*-overlay mesh requires about six times more degrees of freedoms than the respective *hp-d*-mesh. To assess this aspect, the convergence of the error in the energy norm (8.1b) is considered in the following. For this purpose, an overkill solution is computed using 18 recursive refinements and a uniform ansatz order $p = 20$ ($7.2146 \cdot 10^4$ degrees of freedom) to obtain the following reference value:

$$\frac{1}{2} \cdot a(\boldsymbol{u}_{\text{ex}}, \boldsymbol{u}_{\text{ex}}) \approx 7.2694419479162997 \cdot 10^{-2}. \tag{8.27}$$

As for the benchmarks discussed in the previous section, first a uniform order elevation on a pre-graded mesh created by the *hp-d*- or multi-level *hp*-method, respectively, is analyzed. The corresponding results depicted in Figures 8.34 demonstrate that a pure *p*-elevation on the unrefined base mesh yields an algebraic convergence of the error with a rate $\beta \approx 1$. Due to the discontinuity of the stress state, this convergence characteristic matches the expectations. However, the use of a graded *hp-d*-overlay mesh does not change this nature. Instead, the convergence turns algebraic in the asymptotic limit. Further, Figure 8.34a demonstrates that also the gain in accuracy is only marginal.

In analogy to the previously discussed benchmarks, the performance of the uniform order elevation is fundamentally different when using a pre-graded mesh created by the multi-level *hp*-method. Although the asymptotic characteristic is still algebraic, the pre-asymptotic convergence is exponential, and the turning point can be shifted to higher accuracy by using more levels of refinement as shown in Figure 8.34b.



(a) Convergence of *hp-d*-method

(b) Convergence of multi-level *hp*-method

Figure 8.34: *p*-convergence results on a geometrically graded mesh (grading factor $q = 0.5$) created with the *hp-d-* and the multi-level *hp*-method using a uniform *p*-distribution $p = 1, 2, 3, \ldots$. Used abbreviations: k: number of refinements.

The direct comparison of the *hp-d-* and the multi-level *hp*-method in Figure 8.35a shows that the previously mentioned increase in the number of unknowns is compensated by a significant gain of accuracy. The figure also demonstrates that by changing from a uniform to a linear distribution of p over the refinement levels and a simultaneous change of h and p, the exponential convergence characteristic can extended to the asymptotic range. Further, the change to a log-$\sqrt[3]{}$-scaling in Figure 8.35b demonstrates that the convergence is exponential in the form

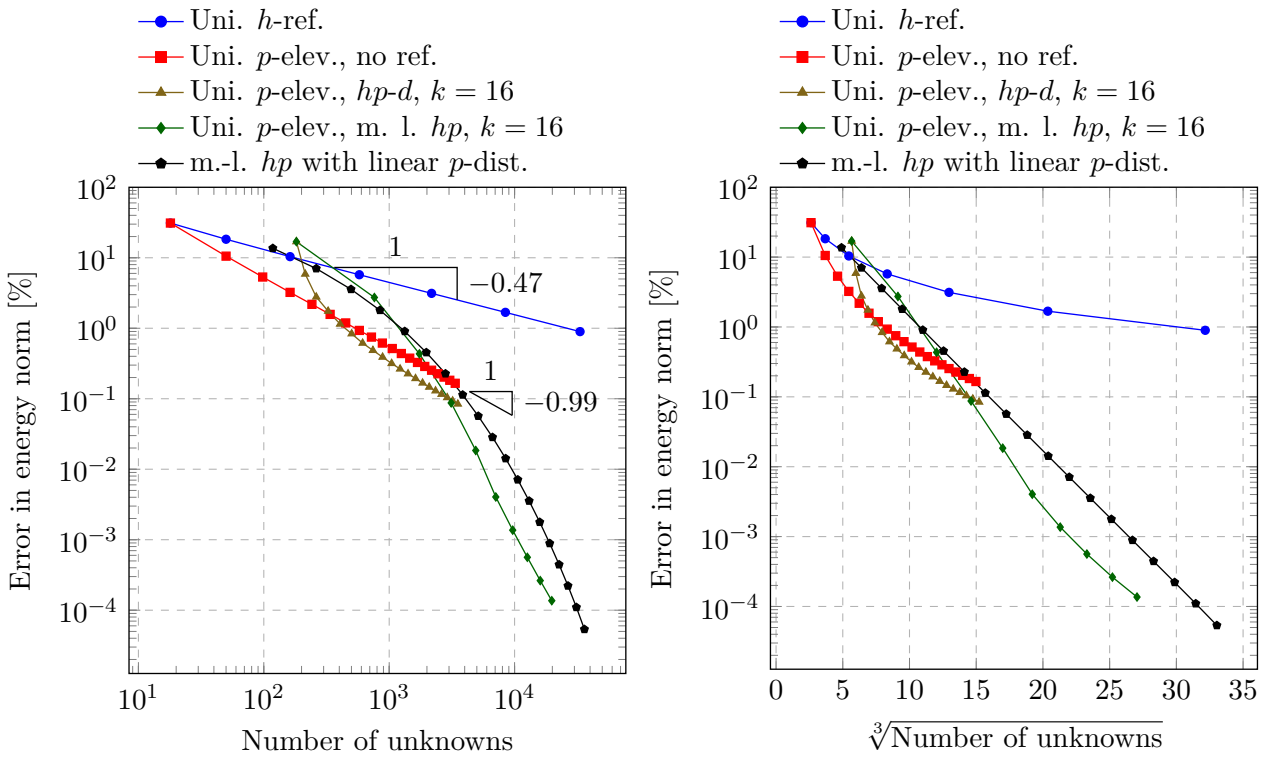
$$||e||_E \leq C \exp\left(\gamma N^{\theta}\right) \quad \text{with } \theta = 1/3, \tag{8.28}$$

which matches the theoretical expectations.

The present study confirms the findings presented in the context of singularities originating form re-entrant corners. This demonstrates that the multi-level *hp*-method is also well applicable to capture discontinuous stress states with high accuracy. This point is addressed again in the context of dynamic mesh refinement in Chapter 9.

8.4 Problems with interior layers: the shock benchmark[g]

The results presented in the previous section demonstrate that the multi-level *hp*-refinement scheme yields exponential convergence for singular problems. The following example analyses



(a) Comparison in log-log-scale

(b) Comparison in semi-log-scale

Figure 8.35: Direct comparison of the different refinement strategies. Used abbreviations: uni.: uniform, ref.: refinement, elev.: elevation, m.-l. *hp*: multi-level *hp*, dist.: distribution.

whether the suggested refinement scheme is also applicable for problems featuring interior or boundary layers. These scenarios are particular challenging as they typically demand complex, three-dimensional mesh refinement patterns. To asses the performance of the multi-level *hp*-method for this kind of applications, the well-known shock problem is considered by following e.g. [Rachowicz et al., 2006].

The domain is defined as

$$\Omega = (0,1)^3, \tag{8.29}$$

and the radial coordinate is considered with respect to a shifted origin:

$$r = \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2} \quad \text{with } x_0 = y_0 = z_0 = -1/4. \tag{8.30}$$

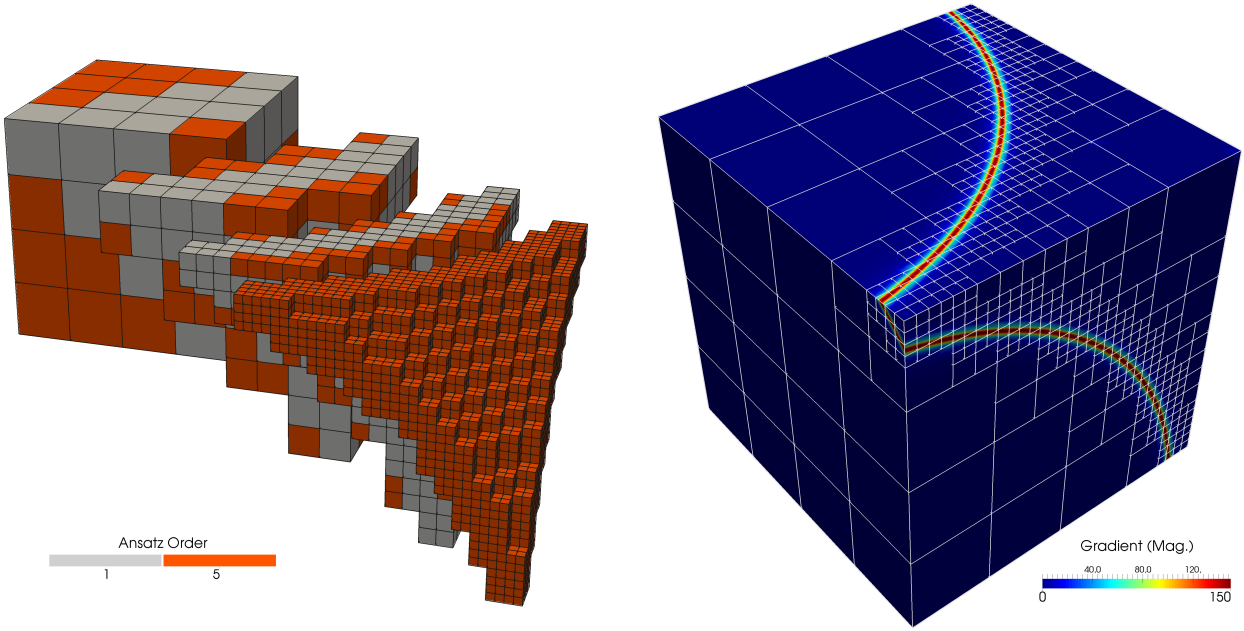
The manufactured solution is then defined as a shock-like function:

$$u = \tan^{-1}(\alpha(r - r_0)) \quad \text{with } r_0 = \sqrt{3} \quad \text{and } \alpha \in \{40, 80, 160\}. \tag{8.31}$$

Here, the scalar factor α determines the sharpness of the shock. This solution is approximated by solving the following partial differential equation numerically

$$\Delta u = s \quad \forall \boldsymbol{x} \in \Omega \tag{8.32a}$$

$$\nabla u \cdot \boldsymbol{n} = g \quad \forall \boldsymbol{x} \in \partial\Omega, \tag{8.32b}$$



(a) Overlay mesh construction for the multi-level *hp* refinement.

(b) Numerical approximation of the gradient ($\alpha = 160$) using three levels of multi-level *hp*-refinement and $p = 5$

Figure 8.36: Local *hp*-mesh refinement for the shock problem [Zander et al., 2016].

in which the source term s and the surface flux g follow from the previously defined solution:

$$s = 2\frac{\alpha}{rf}\left(1 - \alpha^2\frac{r}{f}(r - r_0)\right) \quad \text{and} \quad g = \frac{\alpha}{rf}\begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} \cdot \boldsymbol{n} \tag{8.33a}$$

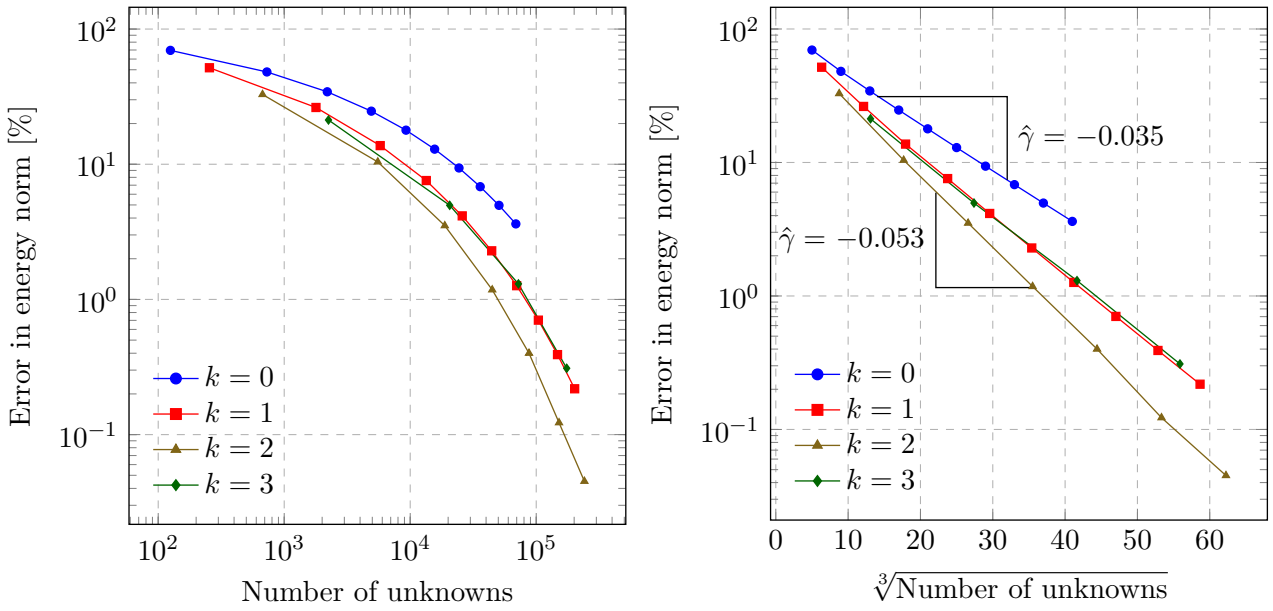
$$\text{with } f = 1 + \alpha^2(r - r_0)^2. \tag{8.33b}$$

To render the solution unique, an additional Dirichlet boundary condition is applied at the point $(1, 1, 1)$. As in the previous example, the error is considered in the energy norm (8.1b) using the following reference values:

$$a(\boldsymbol{u}_{\text{ex}}, \boldsymbol{u}_{\text{ex}}) = \frac{1}{2}\int_\Omega \left(\frac{\alpha}{f}\right)^2 \mathrm{d}\Omega \tag{8.34}$$

$$\begin{aligned} \alpha = 40: &\quad a(\boldsymbol{u}_{\text{ex}}, \boldsymbol{u}_{\text{ex}}) \approx 2.23858386360232 \cdot 10^1 \\ \alpha = 80: &\quad a(\boldsymbol{u}_{\text{ex}}, \boldsymbol{u}_{\text{ex}}) \approx 4.48424691179860 \cdot 10^1 \\ \alpha = 160: &\quad a(\boldsymbol{u}_{\text{ex}}, \boldsymbol{u}_{\text{ex}}) \approx 8.97291532487928 \cdot 10^1 \end{aligned}$$

For the numerical approximation of the solution, the domain is first discretized using $4 \times 4 \times 4$ elements. This base mesh is refined in the vicinity of the shock by superposing multiple levels of finer overlay elements. The refinement is steered geometrically towards the surface of the sphere with radius r_0. This results in the nested, three-dimensional mesh structure presented



(a) Convergence in log-log scale

(b) Convergence in semi-log scale with $\hat{\gamma} = \frac{\gamma}{\ln(10)}$

Figure 8.37: Convergence of the shock problem ($\alpha = 40$) under uniform p-elevation on a k-times graded mesh [Zander et al., 2016].

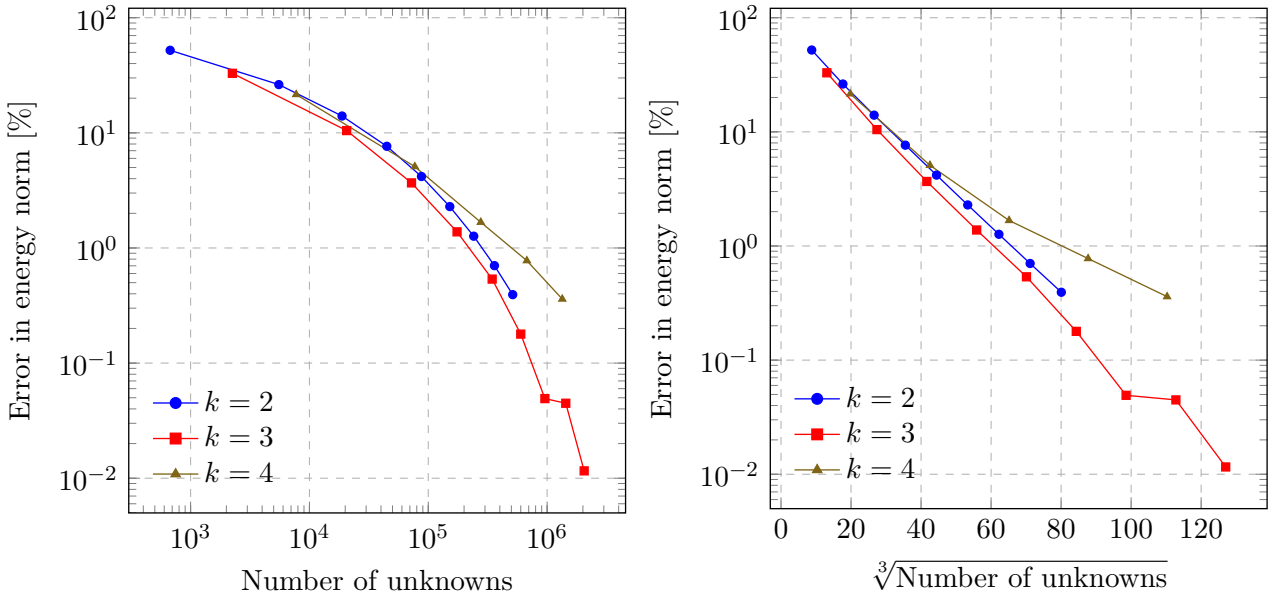
in Figure 8.36a. Figure 8.36b exemplarily depicts one numerical approximation computed on the locally refined discretization.

In the following, the convergence of the error in the energy norm is studied. To this end, the approximation accuracy is increased by elevating the polynomial degree of the shape functions uniformly while keeping the number of refinements fixed. In Figure 8.37, the convergence results are shown for a mild shock with $\alpha = 40$. In this setting, the pure p-extension on the non-refined base mesh already yields exponential convergence in the form

$$||e||_E \leq C \exp\left(\gamma N^{\theta}\right) \quad \text{with } \theta = 1/3, \tag{8.35}$$

which—according to e.g. [Yosibash, 2012]—is to be expected for the regular solution under consideration. However, the convergence rate can be further improved by using one- or two-mesh refinements. As shown in Figure 8.37b, the finer elements in the vicinity of the shock increase the exponential convergence rate γ by about 50%. When using three or more levels of refinement, the convergence rates decrease again because the finer elements introduce unnecessary unknowns, which over-resolve the mild shock.

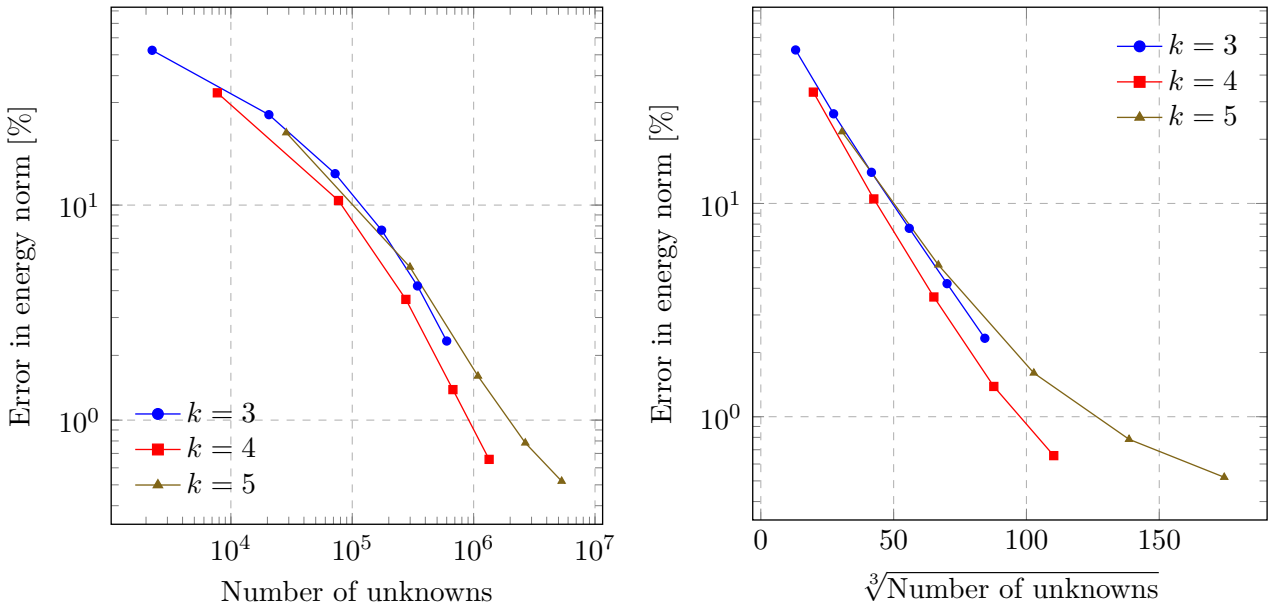
This observation is confirmed when increasing the sharpness of the transition to $\alpha = 80$ and 160, respectively, as shown in Figures 8.38 and 8.39. In these configurations, the width of the shock is significantly smaller. Thus, an accurate resolution demands for a small element size. Accordingly, the optimal number of refinement levels increases to three and four, respectively. Further h-refinement decreases the rates of convergence again. These findings show that the multi-level hp-method can also be used for regular but complex solutions to ensure and improve exponential convergence.



(a) Convergence in log-log scale

(b) Convergence in semi-log scale

Figure 8.38: Convergence of the shock problem ($\alpha = 80$) under uniform p-elevation on a k-times graded mesh [Zander et al., 2016].



(a) Convergence in log-log scale

(b) Convergence in semi-log scale

Figure 8.39: Convergence of the shock problem ($\alpha = 160$) under uniform p-elevation on a k-times graded mesh [Zander et al., 2016].

Chapter 9

Dynamic local mesh refinement and coarsening

The previous chapter analyzed the applicability of the suggested multi-level *hp*-refinement method to resolve singular solution features in the context of stationary examples. An additional challenge for any discretization scheme—which is the focus of the present chapter—is a sharp solution feature that changes its position during the runtime of the simulation. Possible scenarios of engineering relevance include e.g. problems with sliding contact, propagating wave and flame fronts, evolving cracks, phase changes, shocks, changing boundary conditions, or moving heat sources such as lasers. In all these cases, the discretization has to be updated dynamically throughout the runtime of the discretization to keep the refinement local to the respective solution characteristics.

In the context of conventional refine-by-replacement strategies, this implies a continuous creation, replacement, and deletion of elements. In particular, the coarsening of the mesh demands for a sophisticated discretization kernel since a set of fine elements has to be regrouped into one coarse element. To the author's knowledge, only very few code-frameworks can handle this complexity in the context of *hp*-finite elements, for instance Hermes used in e.g. [Korous and Šolín, 2012; Schnepp and Weiland, 2012; Šolín et al., 2010b; Šolín and Korous, 2012a,b].

When instead the suggested refine-by-superposition approach is followed, the refinement can be kept local to the discontinuity by simply "moving" the overlay mesh along with the traveling sharp solution feature while keeping the base mesh unchanged. Technically, this does not require a complex coarsening algorithm. Instead, only the depth of the element refinement tree needs to be adjusted in every time-step of the simulation.

The aim of this section is to study the dynamic mesh refinement capabilities of the refinement scheme. To this end, the sharp resolution of a propagating wave front is considered in the first example. In the second example, the discontinuous load benchmark considered in Section 8.3 is extended by continuously changing the position of the load during the runtime of the simulation. In the third example, the traveling load benchmark is extended to three dimensions.

9.1 Propagating wave[f]

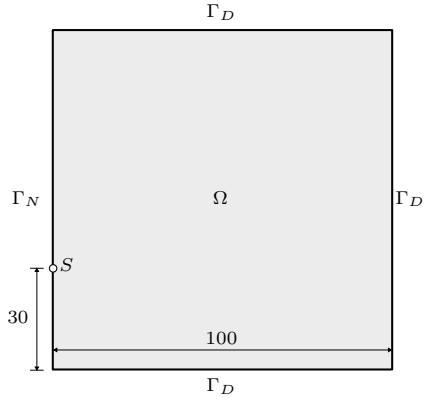


Figure 9.1: Setup of compression wave example [Zander et al., 2015].

The aim of this first example is to analyze the capabilities of the suggested refinement scheme to resolve a propagating wave front. To this end, the evolution of an acoustic wave is considered using the well-known wave equation [Sadd, 2009]

$$\frac{\partial^2 \phi}{\partial t^2} = \Delta\phi + \hat{s}(x,t) \qquad \forall\, x \in \Omega \text{ and } t \in [0,100] \tag{9.1a}$$

$$\phi = 0 \qquad \forall\, x \in \Gamma_D \text{ and } t \in [0,100] \tag{9.1b}$$

$$\nabla\phi \cdot n = 0 \qquad \forall\, x \in \Gamma_N \text{ and } t \in [0,100] \tag{9.1c}$$

$$\phi = 0 \qquad \forall\, x \in \Omega \text{ and } t = 0 \tag{9.1d}$$

$$\frac{\partial \phi}{\partial t} = 0 \qquad \forall\, x \in \Omega \text{ and } t = 0, \tag{9.1e}$$

with Ω being depicted in Figure 9.1. Centered on the point S, the following wavelet-shaped source term is applied to emit a wave front:

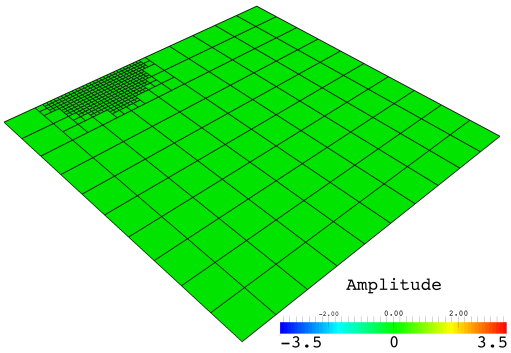
$$\hat{s}(x,t) = a_t(t) \cdot a_x(x) \tag{9.2a}$$

$$\text{with } a_x(x) = A_x \cdot e^{-\frac{(x-S)^2}{2\sigma_x^2}} \quad \text{and} \quad a_t(t) = -A_t \cdot (t-\mu) e^{-\frac{(t-\mu)^2}{2\sigma_t^2}} \tag{9.2b}$$

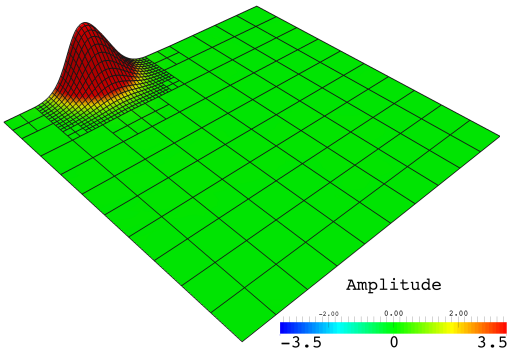
$$\text{and } A_x = 100,\ A_t = 1000,\ \sigma_x = 6,\ \sigma_t = 1,\ \mu = 10 \tag{9.2c}$$

The time range is discretized in 200 equidistant time steps using the Newmark time stepping scheme [Bathe, 2007; Hughes, 2000; Newmark, 1959]. The spatial domain is discretized with 10×10 elements on which shape functions with $p = 4$ are defined. During the simulation, this base discretization is refined by superposing up to three levels of overlay elements. To keep the refinement local to the wave front, the refinement tree depth of every element is adapted in every time step. The refinement is steered by the relative energy contribution of the respective element K in comparison to the total energy of the current solution:

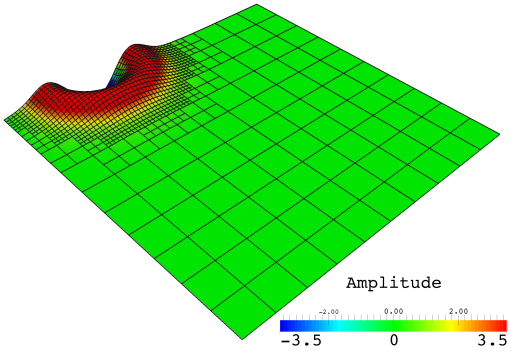
$$\frac{||u||_E^K}{||u||_E^{tot}} > tol,$$



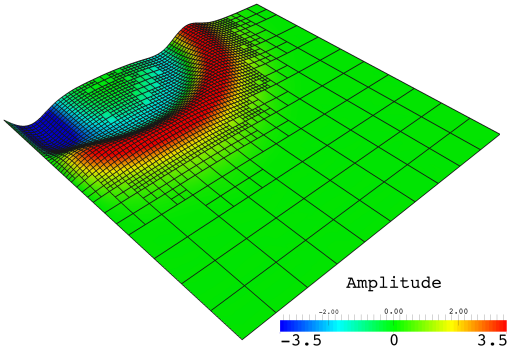
(a) $t = 5\%$



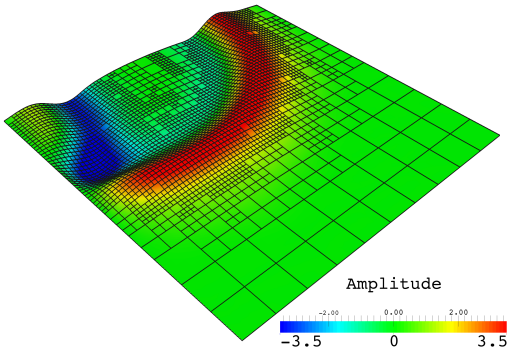
(b) $t = 12\%$



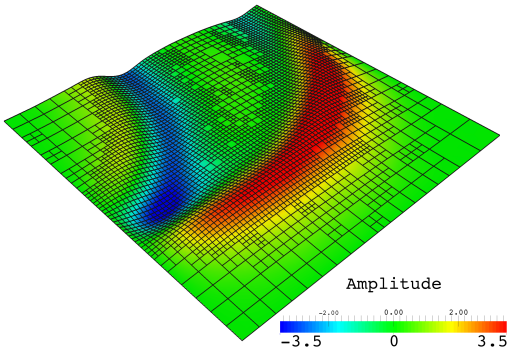
(c) $t = 22\%$



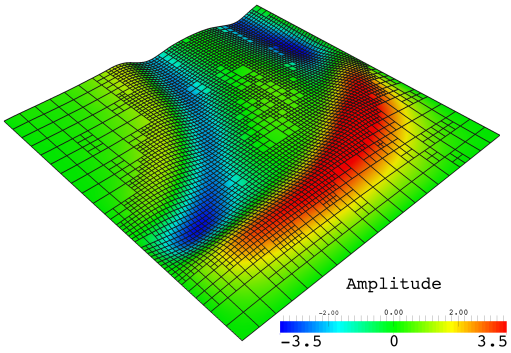
(d) $t = 45\%$



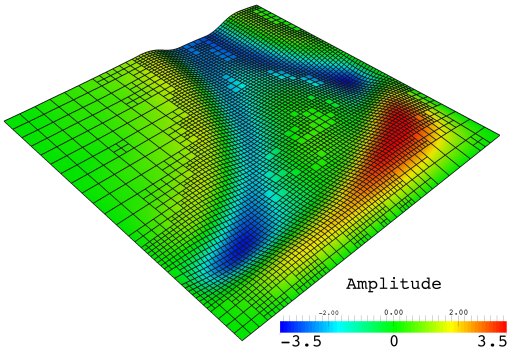
(e) $t = 57\%$



(f) $t = 78\%$



(g) $t = 88\%$



(h) $t = 100\%$

Figure 9.2: Propagation of wave front and discretization over time (max. three overlay meshes, $p = 4$) [Zander et al., 2015].

with *tol* being a user defined tolerance value.

Figure 9.2 depicts the results of this simulation, with the mesh being warped in the z-direction according to the current wave amplitude. The time series shows that the suggested refine-by-superposition approach allows the discretization to change dynamically during the runtime of the simulation. In this way, the refinement can stay local to the propagating wave.

9.2 Traveling load in two dimensions

The results presented in Section 8.3 suggest that the multi-level *hp*-method is well suited to resolve the jump in the stress state caused by a discontinuous traction boundary condition. The current section aims at extending this study to the transient regime. To this end, the *traveling*, discontinuous load illustrated in Figure 9.3 is considered.

As depicted, the elastic domain is defined by the rectangle $\Omega = (-2, 2) \times (0, 1)$. The structure is supported in the vertical direction from below and the horizontal displacement is constrained at the lower left corner. In analogy to Example 8.3, the domain is loaded by a discontinuous traction on the top edge, which—in this example—travels from left to right during the runtime of the simulation. On all other edges, homogeneous Neumann boundary conditions are applied.

The elastic response of the system is governed by the equations of linear elastodynamics reading

$$\rho \ddot{\boldsymbol{u}} = \nabla \cdot \boldsymbol{\sigma} \qquad \forall (\boldsymbol{x}, t) \in \Omega \times (0, T] \tag{9.3a}$$

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon} \qquad \forall (\boldsymbol{x}, t) \in \Omega \times (0, T] \tag{9.3b}$$

$$\boldsymbol{\varepsilon} = \frac{1}{2} \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^\top \right) \quad \forall (\boldsymbol{x}, t) \in \Omega \times (0, T]. \tag{9.3c}$$

As for the stationary case, $\boldsymbol{\sigma}$ denotes the stress tensor, $\mathbb{C}$ the elastic material tensor, $\boldsymbol{\varepsilon}$ the engineering strain, and $\boldsymbol{u}$ the displacement vector. Additionally, ρ denotes the density of the elastic material. The values of the material parameter are given in Figure 9.3. The problem definition is completed by the homogeneous initial conditions for the displacement and the

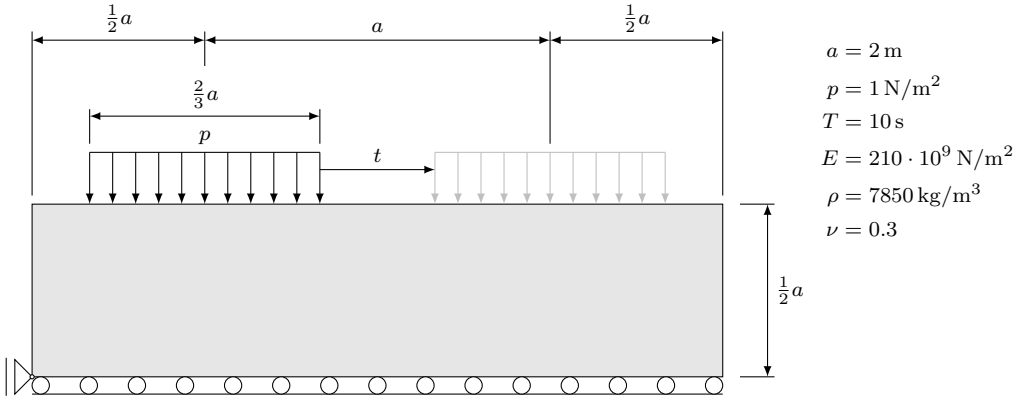
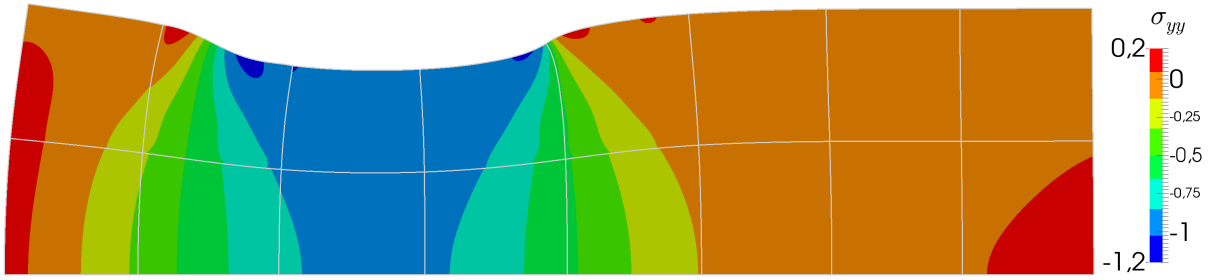
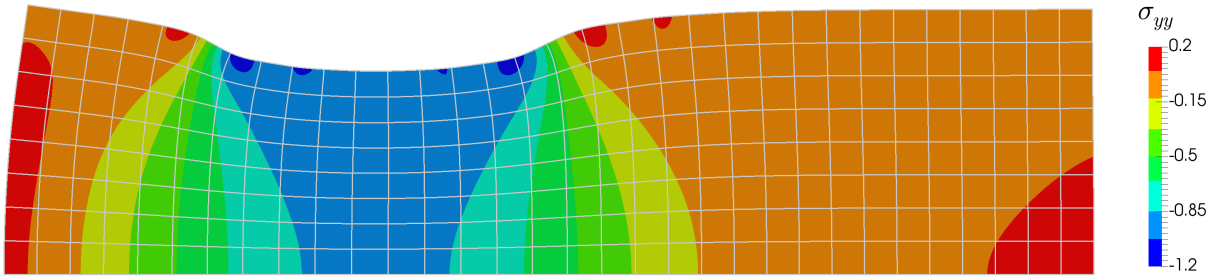


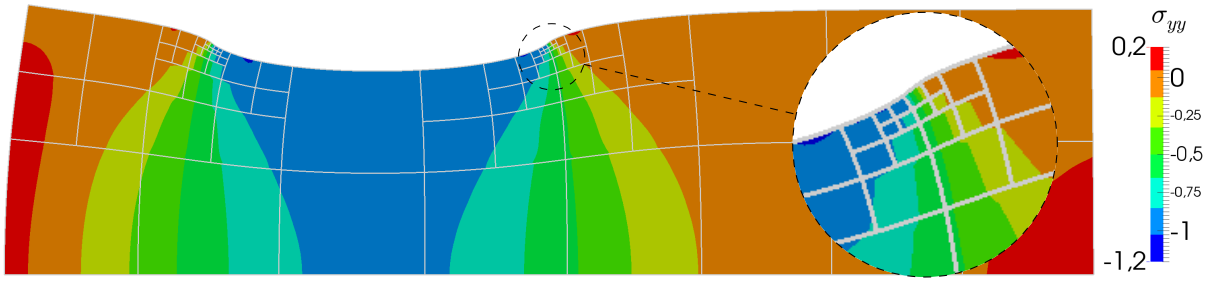
Figure 9.3: Setup of traveling load example (plane stress).



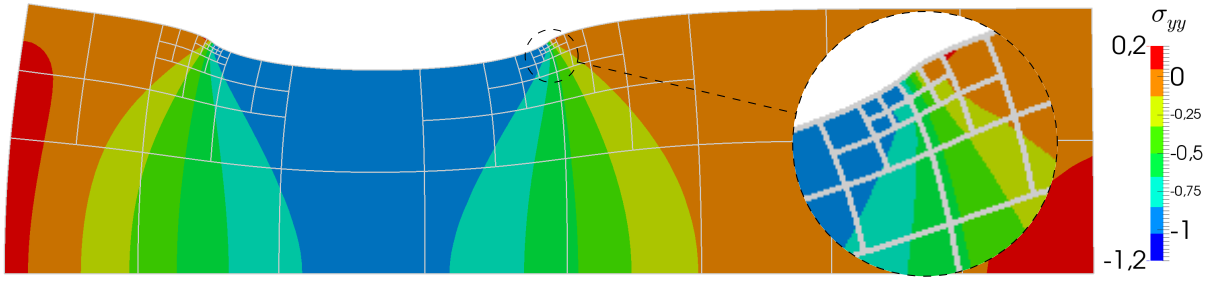
(a) p-FEM, 8×2 elements, $p = 5$, 902 degrees of freedom



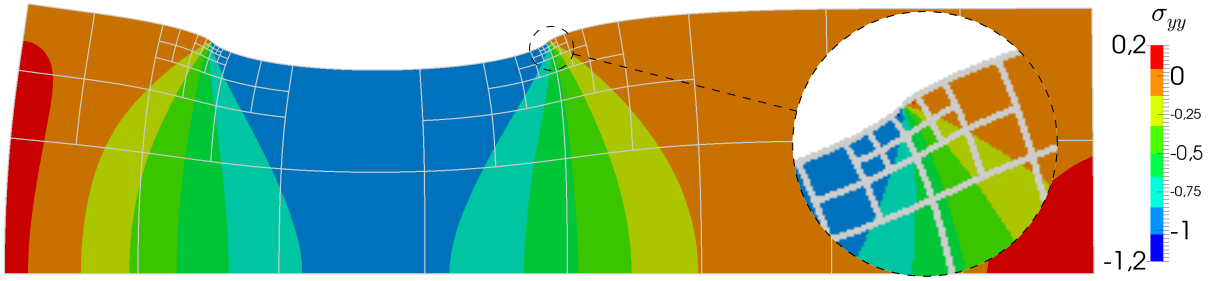
(b) B-spline basis, 32×8 knot-span elements, $p = 5$, 962 degrees of freedom



(c) hp-d-refinement, 8×2 elements, 5 refinement levels, $p = 5$, 1004 degrees of freedom



(d) Multi-level hp-refinement, 8×2 elements, $k = 5$, linear p-distribution, $p_{\max} = 5$, 1996 degrees of freedom



(e) Multi-level hp-refinement, 8×2 elements, $k = 5$, linear p-distribution, $p = 5$, 3812 degrees of freedom

Figure 9.4: Comparison of stress distribution σ_{yy} at $t = 24\%$ using different refinement strategies. For better visualization, the deformation is magnified by a factor of $5 \cdot 10^{10}$.

velocity

$$\boldsymbol{u} = \dot{\boldsymbol{u}} = \mathbf{0} \quad \forall \boldsymbol{x} \in \Omega,\, t = 0. \tag{9.4}$$

To approximate the resulting deformation, the above equations are considered in their weak form. For the temporal discretization, the Newmark time-stepping scheme with 96 equidistant time steps is used [Bathe, 2007; Hughes, 2000; Newmark, 1959]. For the spatial discretization, a finite element mesh consisting of 8×2 high-order elements is used.

The numerical approximation obtained with this discretization is depicted in Figure 9.4a exemplarily for an intermediate time step. The result demonstrates that the high-order base mesh can accurately capture the smooth part of the solution away from the discontinuous loading condition. However, the jump in the stress state cannot be resolved sharply by the coarse mesh. This is confirmed by the plot of the numerically computed vertical stress σ_{yy} along the top edge depicted in Figure 9.5. In analogy to the stationary case in Example 8.3, the coarse, high-order base mesh is not able to resolve the discontinuous stress state sharply. Instead, the numerical solution suffers from severe oscillations that spread over large parts of the domain and only decay after crossing multiple elements.

As discussed in the previous chapter, this phenomena is caused by the C^∞-continuity of the used polynomial shape functions, which does not allow for a separate representation of the two distinct stress states within one element. This limitation causes the oscillatory artifacts. Only the C^0-inter-element continuity of the polynomial basis functions partially reduces the coupling between the numerical solutions of adjacent elements. This limits the range of the oscillations to the elements in the vicinity of the stress discontinuity.

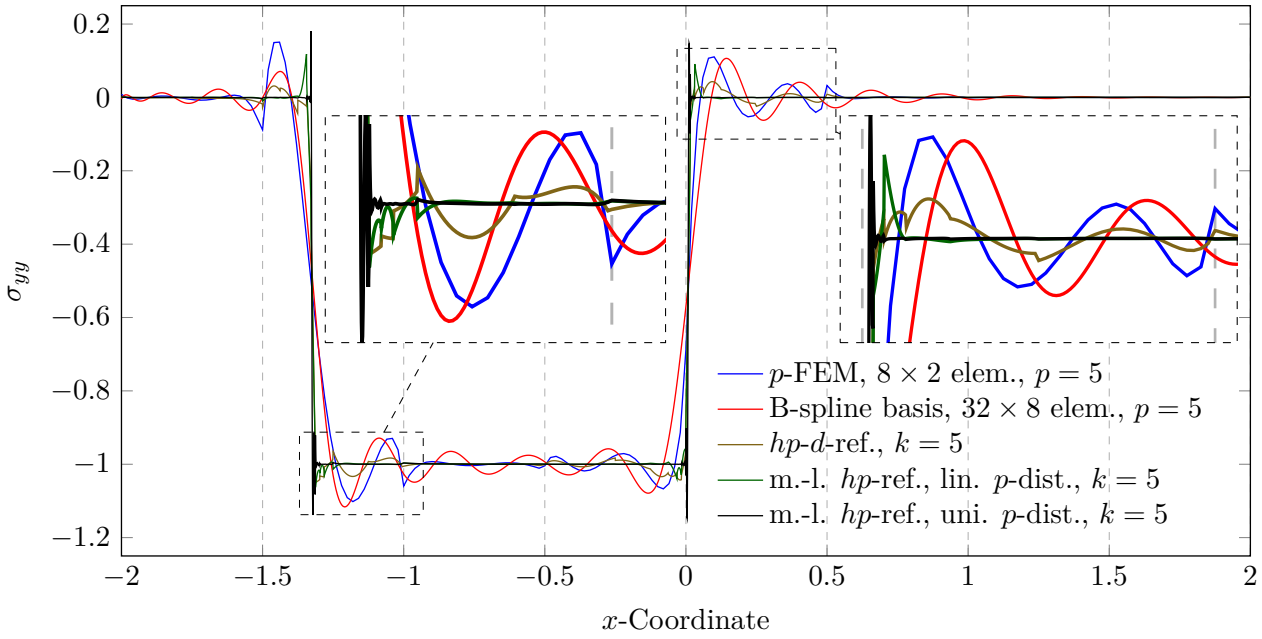
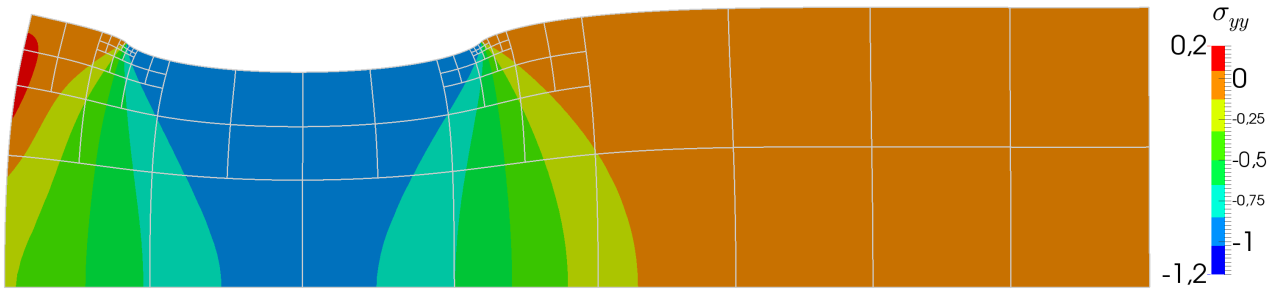
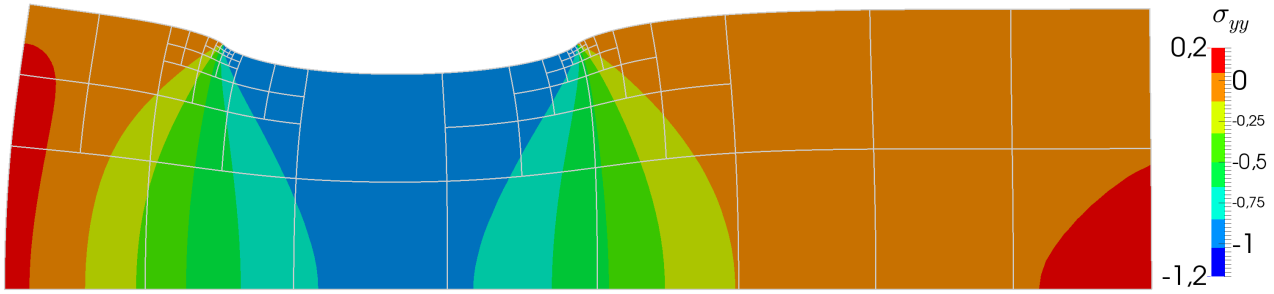


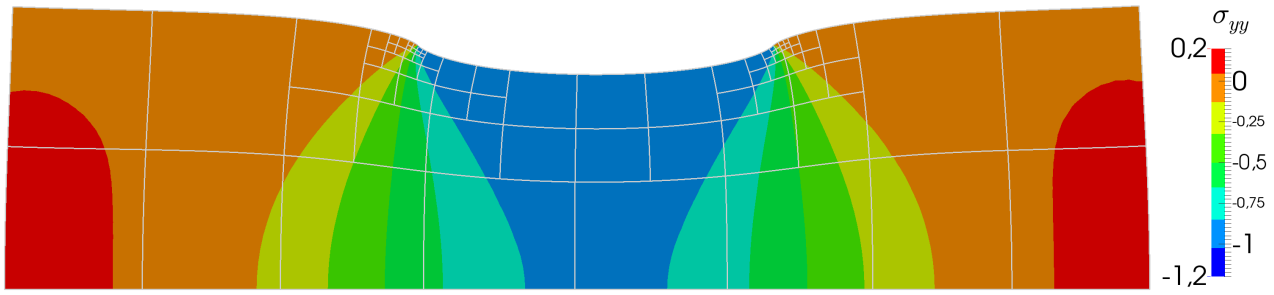
Figure 9.5: Surface traction σ_{yy} along the top of the block at $t = 24\%$. The dashed lines along the abscissa mark the boundaries of the base elements along the path. Used abbreviations: elem.: elements, ref.: refinement, m.-l. hp: multi-level hp, k: number of refinement levels, lin.: linear, uni.: uniform, dist.: distribution.



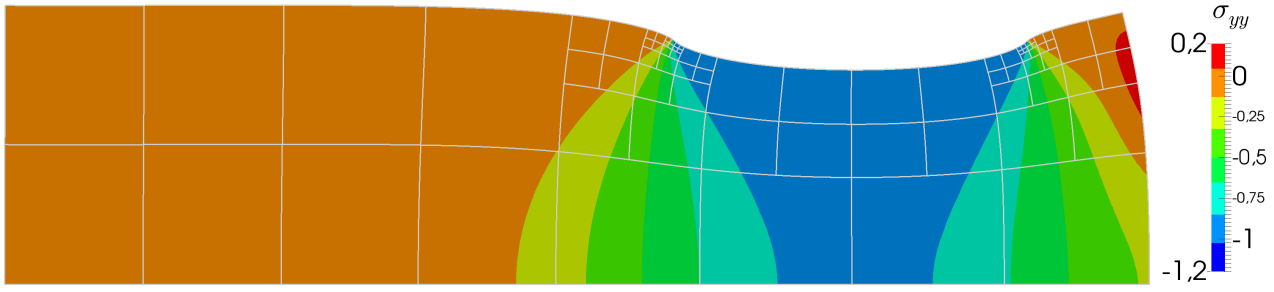
(a) $t = 8\%$



(b) $t = 24\%$



(c) $t = 57\%$



(d) $t = 100\%$

Figure 9.6: Stress distribution σ_{yy} at different time steps using the uniform multi-level *hp*-refinement strategy (5 levels, $p = 5$). For better visualization, the deformation is magnified by a factor of $5 \cdot 10^{10}$.

Accordingly, the oscillatory phenomena becomes more pronounced when using shape functions that provide a higher global continuity, such as e.g. B-splines (see e.g. [Cottrell et al., 2009; Hughes et al., 2005] on the use of B-spline basis functions). This is demonstrated in Figures 9.4b and 9.5 using 32×8 knot span elements of order five yielding a similar number of unknowns as the previously considered p-FEM basis. The depicted results demonstrate that the C^{p-1}-inter-element continuity of the B-spline basis functions prevents the effective separation of the two solution states. Hence, the oscillations spread over the complete length of the domain.

These observations underline the need for a local mesh refinement with a reduced solution continuity in the vicinity of the changing loading conditions. For this purpose, the simulation is repeated using the suggested overlay refinement technique. To keep the refinement local to the discontinuity, the refinement tree depth of every base-element is adjusted according to the current position of the moving load in every time-step of the simulation. As depicted in Figure 9.6, this allows the refinement zone to follow the moving discontinuity, which is traveling through the domain.

In a first step, the refinement is carried out following the *hp-d* approach. Here, the base mesh remains unchanged, and the accuracy is increased by superposing five levels of overlay-elements with linear shape functions only. The results depicted in Figure 9.4c demonstrate a significant improvement in the approximation quality. In particular, the boundary of the load surface is resolved more sharply. However, as shown in Figure 9.5, the oscillations in the numerical stress solution are *not* bounded to the finest element level but still spread over large parts of the domain—although with a reduced amplitude.[g]

In analogy to the stationary version of this benchmark, this poor result quality is due to the use of overlay meshes with only linear shape functions. Since the high-order modes remain active on the coarse base mesh, their support remains unchanged. Hence, the linear overlay mesh only breaks the continuity of the approximation within the base element but does not yield a high-order mesh refinement.

This is fundamentally different when using the graded or uniform multi-level *hp*-approach. Since the high-order shape functions are moved from the base to the leaf-elements, their support is minimized. Only this corresponds to a high-order mesh refinement and yields a substantial increase of the approximation quality as shown in Figures 9.4d and e. Both versions of the multi-level *hp*-method limit the spreading of the oscillations. In particular the uniform multi-level *hp*-approach renders the traction surface extremely sharp, whereas the use of a linearly increase p-distribution appears to be a good compromise between accuracy and efficiency.[g]

These results demonstrate that the high approximation quality observed for the stationary Benchmark 8.3 also carry over to situations that demand a dynamic change of the discretization during the runtime of the simulation.

9.3 Traveling load in three dimensions[b]

The aim of the last example of this section is to analyze whether the approximation properties observed for the previous two-dimensional example also carry over to higher dimensions. To this end, the three-dimensional counterpart of the last example, depicted in Figure 9.7, is considered.

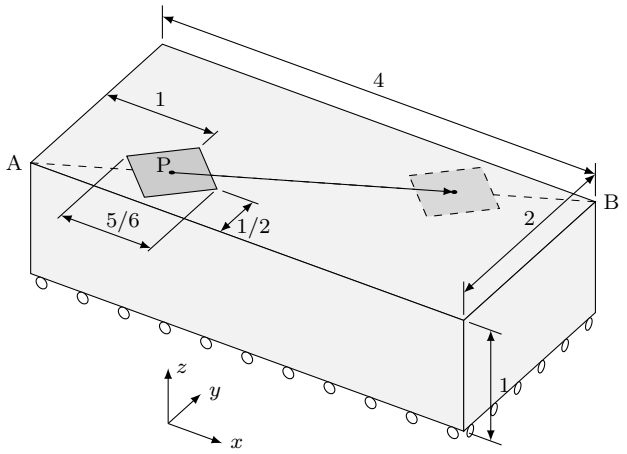


Figure 9.7: Setup of traveling load example [Zander et al., 2016].

Problem definition[g]

In analogy to the two-dimensional case, a linear elastodynamic problem is solved. The density is chosen as $\rho = 7850\,\mathrm{kg/m^3}$, the Young's modulus as $E = 210 \cdot 10^9\,\mathrm{N/m^2}$ and Poisson's ratio as $\nu = 0.3$. Furthermore, the end time T is set to 10 s.

As depicted in Figure 9.7, the domain is simply supported on the lower face:

$$u_z = 0 \quad \forall (\boldsymbol{x}, t) \in \Gamma_D \times (0, T] \quad \text{with } \Gamma_D = \{(x, y, z) \in \Omega \,:\, z = 0\}. \tag{9.5}$$

To prevent rigid body motions, the displacement is further fixed at two additional points as follows:

$$u_x = u_y = 0 \quad \text{on } \boldsymbol{x} = (0, 0, 0), t \in (0, T] \tag{9.6}$$
$$u_y = 0 \quad \text{on } \boldsymbol{x} = (2, 0, 0), t \in (0, T]. \tag{9.7}$$

A vertical traction

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = -\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \forall (\boldsymbol{x}, t) \in \Gamma_q(t) \times (0, T] \tag{9.8}$$

is applied onto the square load surface depicted in dark gray in Figure 9.7:

$$\Gamma_q(t) = \{(x, y, z) \in \Omega : |x - P_x(t)| + |y - P_y(t)| \leq w/2, z = 1\} \quad \text{with } w = \frac{5}{6}. \tag{9.9}$$

During the simulation runtime, the center point P—and with it the load surface—moves across the top surface of the domain as follows:

$$\begin{bmatrix} P_x(t) \\ P_y(t) \end{bmatrix} = -\begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix} + \frac{t}{T}\begin{bmatrix} 2 \\ 1 \end{bmatrix}. \tag{9.10}$$

On the remaining part of the boundary

$$\Gamma_0 = (\partial\Omega \setminus \Gamma_D) \setminus \Gamma_q, \tag{9.11}$$

homogeneous Neumann boundary conditions are applied

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \mathbf{0} \quad \forall (\boldsymbol{x}, t) \in \Gamma_0(t) \times (0, T]. \tag{9.12}$$

To complete the problem definition, the initial displacement and velocity are set to zero:

$$\boldsymbol{u} = \dot{\boldsymbol{u}} = \mathbf{0} \quad \forall \boldsymbol{x} \in \Omega,\ t = 0. \tag{9.13}$$

Numerical setup[g]

For the numerical approximation of the solution, the multi-level *hp*-approach is combined with the Newmark time stepping scheme using 48 time steps [Bathe, 2007; Hughes, 2000; Newmark, 1959]. In a first approach, the spatial domain Ω is discretized by $4 \times 2 \times 2$ elements of order $p = 4$. To avoid any side-effects resulting from the numerical quadrature of the discontinuous traction (9.8), a high-resolution integration mesh of 64×64 integration cells per element is used to accurately capture the load on the top surface. This integration mesh is used for all following mesh refinement configurations.

From the boundary condition (9.8) applied on the top surface, it is evident that the vertical stress σ_{zz} on the top surface should be -1 within the load surface Γ_q and zero outside. Any deviation from this expected result can thus be identified as a discretization error.

Numerical results

The stress distribution obtained using this numerical setup is depicted exemplarily for an intermediate time step in Figure 9.9a. As for the two-dimensional benchmark considered in

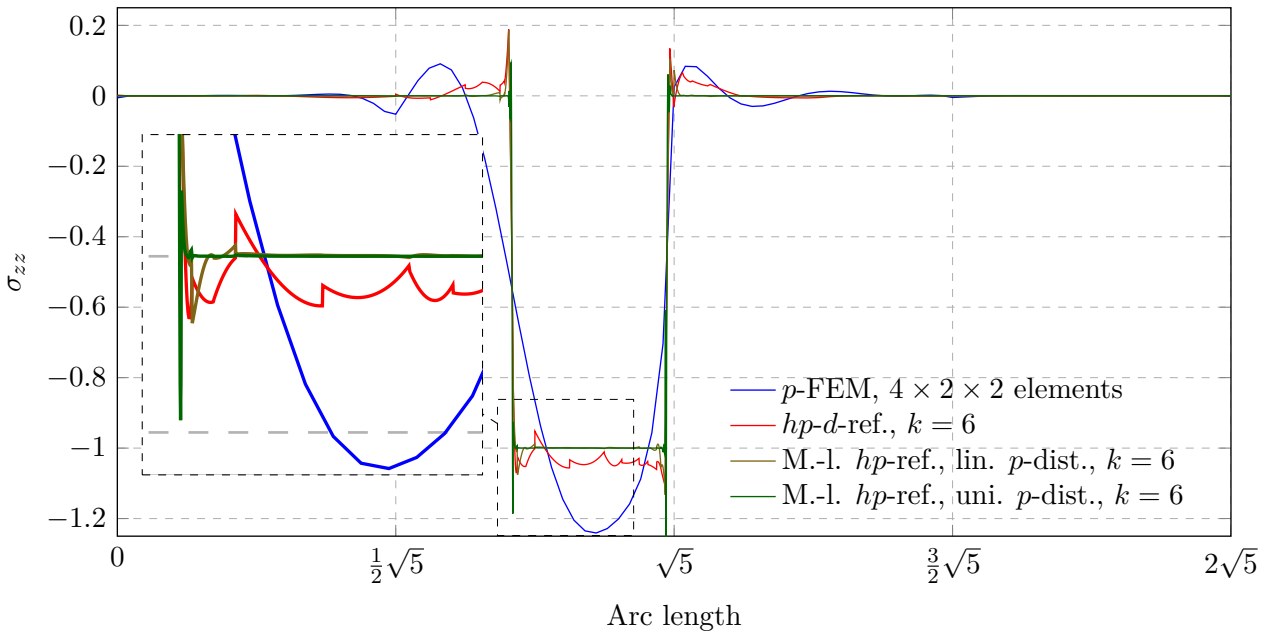
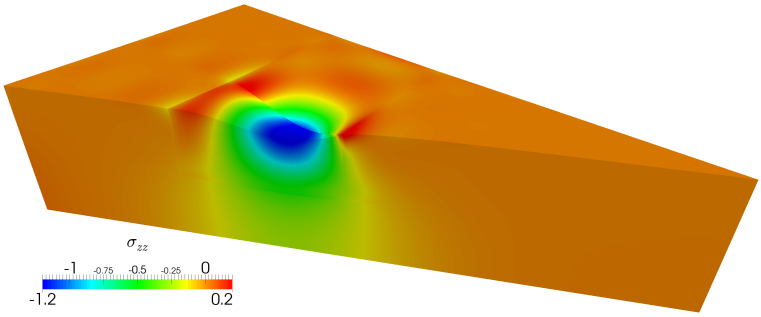
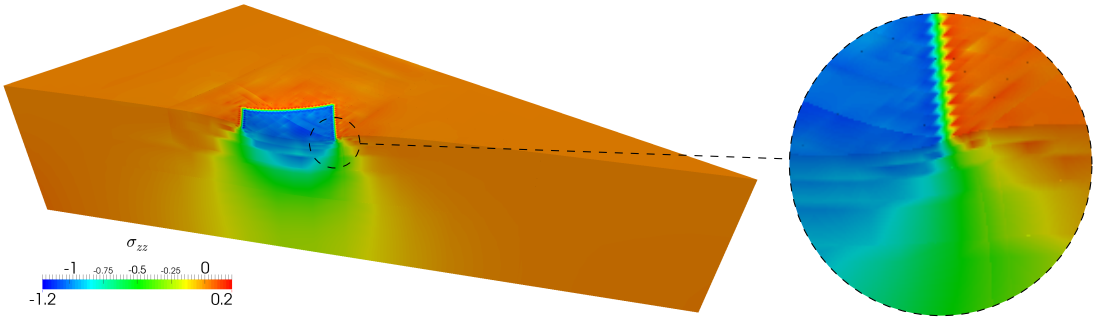


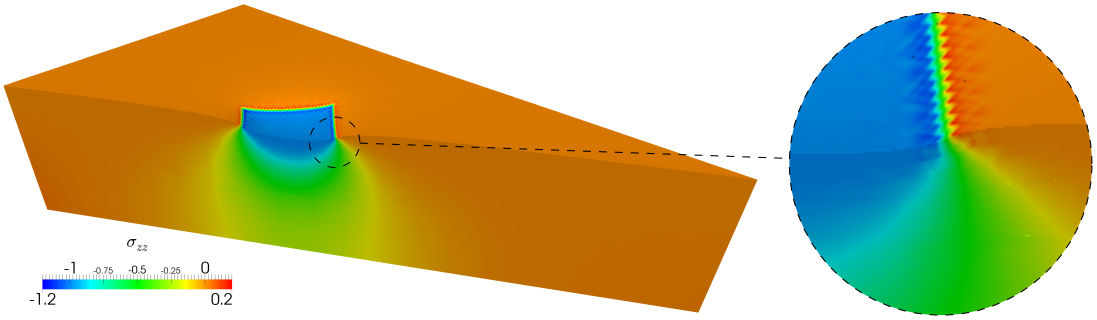
Figure 9.8: Vertical stress σ_{zz} along top of the block (cut $\overline{AB}$) at $t = 38\%$ [Zander et al., 2016]. Used abbreviations: ref.: refinement, m.-l. *hp*: multi-level *hp*, *k*: number of refinement levels, lin.: linear, uni.: uniform, dist.: distribution.



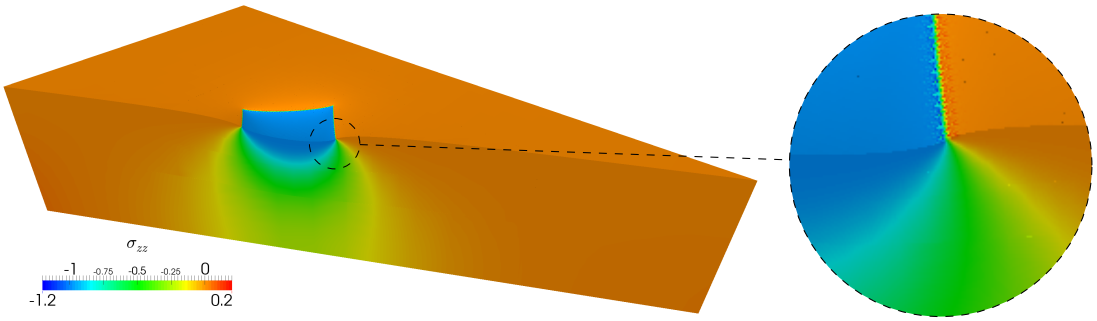
(a) No refinement ($p = 4$, 4,131 unknowns)



(b) *hp-d*-refinement [Rank, 1992; Schillinger et al., 2012b] ($p = 4$, $k = 6$, 10,257 unknowns)

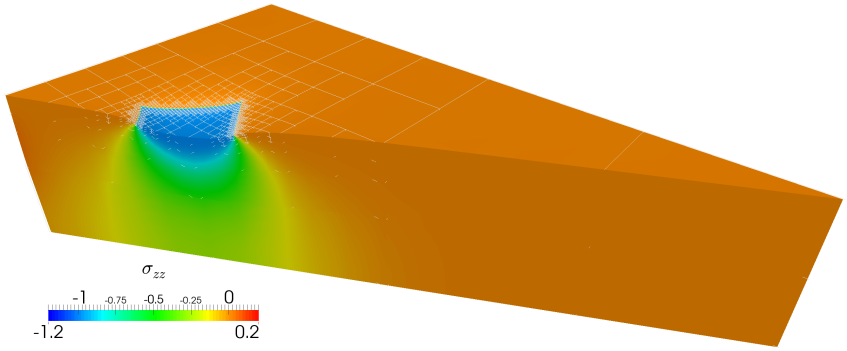


(c) Multi-level *hp*-refinement (linear p-distribution, $p_{\max} = 4$, $k = 6$, 51,165 unknowns)

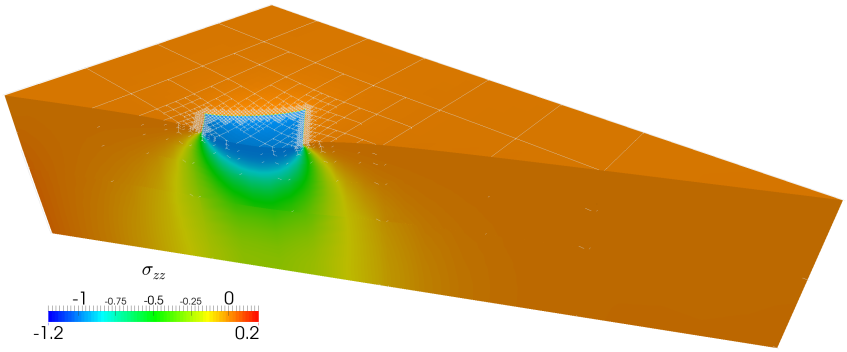


(d) Multi-level *hp*-refinement (uniform p-distribution, $p = 4$, $k = 6$, 472,215 unknowns)

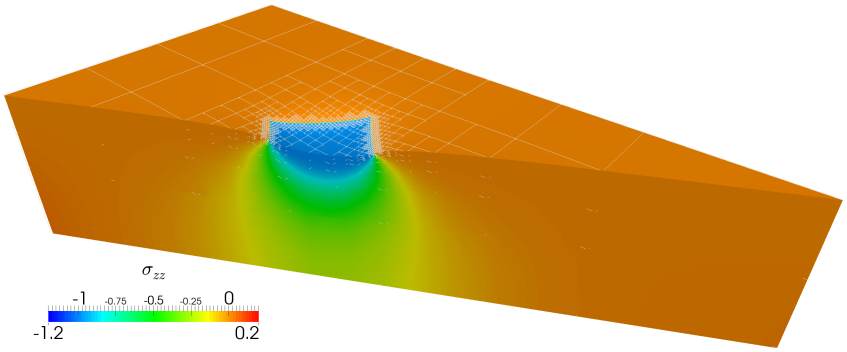
Figure 9.9: Comparison of vertical stress distribution σ_{zz} at $t = 38\%$ using different refinement strategies. For better visualization, the deformation is magnified by a factor of $5 \cdot 10^{10}$. Used abbreviations: p: polynomial degree of shape functions; k: number of refinement levels [Zander et al., 2016].



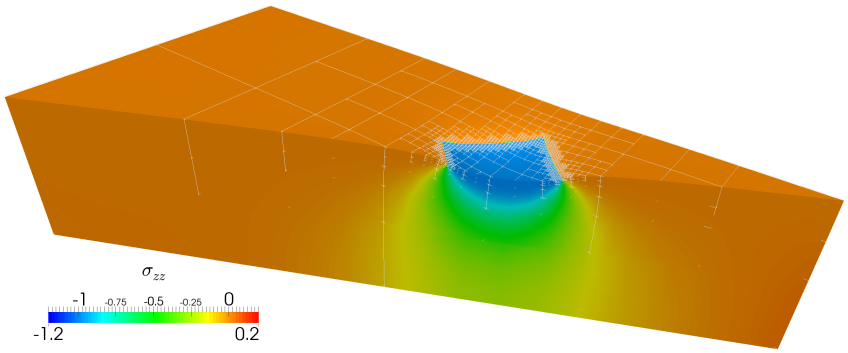
(a) $t = 4\%$



(b) $t = 24\%$



(c) $t = 38\%$



(d) $t = 100\%$

Figure 9.10: Vertical stress distribution σ_{zz} at different time steps using the uniform multi-level *hp*-refinement strategy ($p = 4$, $k = 6$). For better visualization, the deformation is magnified by a factor of $5 \cdot 10^{10}$ [Zander et al., 2016].

the previous section, the coarse, high-order base mesh can correctly capture the global solution characteristic of the problem. However, the boundary of the load surface cannot be resolved sharply. Instead, the solution suffers from significant oscillations that spread over large parts of the domain as depicted in Figure 9.8.

In analogy to the two-dimensional counterpart, the accurate resolution requires a non-uniform refinement of the mesh that stays local with respect to the traveling load. As shown in Figure 9.10, this dynamically changing discretization also works for the present three-dimensional setting by the applied refine-by-superposition approach. The discretization is refined as soon as the load surface Γ_q enters an element and coarsened again when the singularity has left the element.

As depicted in Figure 9.9, this non-uniform refinement yields a significant improvement of the approximation quality. Furthermore, the cut along the top surface depicted in Figure 9.8 shows that—also for this three-dimensional benchmark—the oscillations in the stress state are confined to the finest element level. These results demonstrate that the potential of the multi-level *hp*-refinement method to resolve moving sharp solution features carry over to three-dimensional problems.

Chapter 10

Application to fictitious domain problems

The previous two chapters analyzed the applicability of the suggested multi-level *hp*-refinement scheme to accurately resolve the complex solution state in the vicinity of singularities. As demonstrated, these points of singularities can be fixed in space or even change their position during the runtime of the simulation.

Another challenge for any discretization scheme—which is the focus of this chapter—arises when not only the singularity itself but also the complete object under consideration is changing its position or shape during the course of the simulation. Possible scenarios of engineering interest that face this problem include e.g. the motion of rigid objects in fluid flow, or moving non-conductors in an electric or thermal field. The common challenge of these applications is the continuous change of the computational domain Ω throughout the runtime of the simulation.

Classical discretization approaches resolve this problem by updating the mesh in every time-step. An alternative approach—followed in this work—is the use of a fixed background mesh. In this discretization technique, the geometry of the moving body is not resolved by the mesh explicitly. Instead, it is *immersed* or *embedded* in the discretization. One discretization approach that follows this idea and has gained increasing popularity in the past decade is the Finite Cell Method (FCM).

A second challenge rising in the context of moving inclusions is that these inclusions might introduce steep gradients, discontinuities or even singularities in the solution. As in the case of stationary problems, this calls for a local refinement of the discretization. However, since the inclusions are changing their position, this refinement has to be adapted dynamically during the course of the simulation.

As demonstrated in the previous chapter, the suggested superposition-based *hp*-refinement allows for this flexibility in the discretization. For this reason, the aim of this chapter is to analyze whether the multi-level *hp*-approach and the finite cell concept can be combined to capture the solution state around moving objects on a fixed, non-boundary conforming background mesh. To this end, the first section of this chapter reviews the core principles of the Finite Cell Method. In the second part, the combination of the two methods is tested for different two- and three-dimensional examples.

10.1 Essential idea of the Finite Cell Method[e]

In daily engineering practice, the mesh-generation process is often time consuming and sometimes even a limiting factor when using the Finite Element Method for industrial applications [Cottrell et al., 2009; Hughes et al., 2005]. This problem becomes even more severe for the p-version of the Finite Element Method: since the solution can be represented by coarser elements than in the h-version, the elements have to assume the possibly complex shape of the domain to yield accurate results [Szabó and Babuška, 1991]. The Finite Cell Method (FCM)—introduced in [Düster et al., 2008; Parvizian et al., 2007]—aims at reducing this problem by combing the benefits of higher-order Finite Elements with the ideas of fictitious domain methods.

This approach has proven its excellent applicability in various contexts of engineering relevance including linear elasticity [Düster et al., 2008; Parvizian et al., 2007], topology optimization [Cai et al., 2014; Parvizian et al., 2011], geometrically nonlinear continuum mechanics [Schillinger, 2012; Schillinger et al., 2012b,c], contact mechanics [Bog et al., 2015], computational steering [Yang et al., 2012a,b], bio-medical engineering [Ruess et al., 2012; Verhoosel et al., 2015; Wille et al., 2016], elastoplasticity [Abedian et al., 2013, 2014], wave propagation in heterogeneous materials [Duczek et al., 2014; Joulaian et al., 2014], implicit dynamics [Elhaddad et al., 2015], local enrichment for material interfaces [Joulaian and Düster, 2013], numerical homogenization [Heinze et al., 2015], porous media [Duczek et al., 2015; Joulaian et al., 2016], unstructured meshes [Duczek and Gabbert, 2016; Varduhn et al., 2016], convection diffusion problems [Cai, 2013], Karhunen-Loeve expansion [Betz et al., 2014], thin-walled structures [Guo and Ruess, 2015a,b; Rank et al., 2011], design-through analysis and isogeometric analysis [Kudela et al., 2015, 2016; Rank et al., 2012; Ruess et al., 2013, 2014; Schillinger et al., 2012a], thermoelasticity [Zander et al., 2012], cohesive delamination [Zander et al., 2017], and fluid-structure interaction [Kamensky et al., 2015; Xu et al., 2015]. Recently, the convergence properties of the method have been analyzed in detail in [Dauge et al., 2015]. A comprehensive review of the method and its recent extensions is given in [Schillinger and Ruess, 2014]. An open-source MATLAB-toolbox presented in [Zander et al., 2014] offers an easy start into this research field.[h]

The core principles of this approach are outlined in the first part of this section. The second part addresses the imposition of boundary conditions for non boundary-conforming discretizations.

10.1.1 Basic concept

The essential idea of the Finite Cell Method is to simplify the mesh-generation process by separating the approximation of the analytical solution from the geometry representation of the physical domain. For this purpose, two independent meshes are introduced: the *solution* and the *integration* mesh. The first mesh is used to approximate the analytical solution and, thus, spans the shape functions. However, it does not resolve the geometry of the domain. This is done by the second mesh, which is completely independent of the first one and does not introduce any additional degrees of freedom.

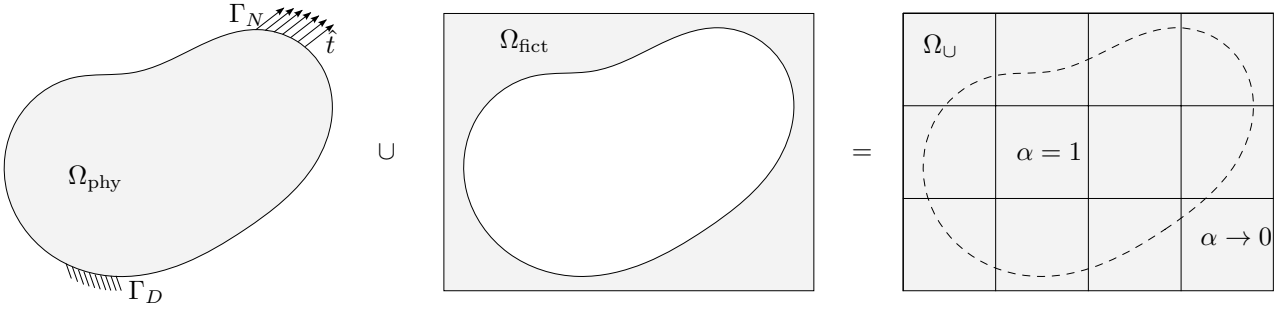


Figure 10.1: Principle idea of the finite cell method (FCM) following [Parvizian et al., 2007]: the complex physical domain Ω_{phy} is extended by a fictitious domain Ω_{fict} such that their union $\Omega_{\cup}$ yields a much simpler geometry. The original problem is recovered using the indicator function α (taken from [Zander et al., 2017]).

Solution mesh

The first step to separate the two meshes is to *embed* the actual physical domain Ω_{phy} in a fictitious domain Ω_{fict} such that their union $\Omega_{\cup}$ yields a simple shape. This idea is depicted in Figure 10.1. To recover the original boundary value problem on the new domain $\Omega_{\cup}$, an *indicator* function α is defined as

$$\alpha(x) = \begin{cases} 1 & \forall\, \boldsymbol{x} \in \Omega_{\text{phy}} \\ 0 & \forall\, \boldsymbol{x} \notin \Omega_{\text{phy}}{}^{1}. \end{cases} \tag{10.1}$$

Using this indicator, the bi-linear form $a(\cdot,\cdot)$ of, for example, a linear elastic problem can then be rewritten as

$$a\left(\boldsymbol{u},\boldsymbol{v}\right) = \int\limits_{\Omega_{\text{phy}}} \boldsymbol{\varepsilon}\left(\boldsymbol{v}\right) : \mathbb{C} : \boldsymbol{\varepsilon}\left(\boldsymbol{u}\right) \,\mathrm{d}\Omega \tag{10.2a}$$

$$= \int\limits_{\Omega_{\text{phy}}} 1 \cdot \boldsymbol{\varepsilon}\left(\boldsymbol{v}\right) : \mathbb{C} : \boldsymbol{\varepsilon}\left(\boldsymbol{u}\right) \,\mathrm{d}\Omega + \int\limits_{\Omega_{\text{fict}}} 0 \cdot \boldsymbol{\varepsilon}\left(\boldsymbol{v}\right) : \mathbb{C} : \boldsymbol{\varepsilon}\left(\boldsymbol{u}\right) \,\mathrm{d}\Omega \tag{10.2b}$$

$$\approx \int\limits_{\Omega_{\cup}} \alpha \cdot \boldsymbol{\varepsilon}\left(\boldsymbol{v}\right) : \mathbb{C} : \boldsymbol{\varepsilon}\left(\boldsymbol{u}\right) \,\mathrm{d}\Omega. \tag{10.2c}$$

The right hand side $f(\cdot)$ is treated in analogy. In this way, the meshing of the complex physical domain can be simplified, as the simply-shaped domain $\Omega_{\cup}$ can be discretized instead using a Cartesian-shaped solution mesh. This renders the mesh-generation process trivial. To avoid a confusion of names, the resulting non boundary-conforming, high-order elements are denoted as *finite cells* giving the method its name.

[1] Choosing α as zero outside of the physical domain has a detrimental effect on the condition number of the stiffness matrix. To control the conditioning, the value of α is typically set to a small value that allows solving the system of equations without numerical problems. Experience shows that direct solvers can handle values of $\alpha \approx 10^{-10}$ without major difficulties.

Integration mesh

The second step is to recover the domain geometry using a separate integration mesh. In principle, this second mesh can be generated following various strategies.

The classical approach is to use a space-tree as integration mesh. Starting from the non-conforming solution mesh, cut cells are recursively refined towards the domain boundary yielding a quad- or octree structure. This approach has the advantage that it requires only a simple inside-outside test. It can therefore also be applied in conjunction with simple geometry modeling techniques such as voxel-models.

A more advanced approach is to use conforming integration simplices by triangulating the physical domain. XFEM- and Level Set-based methods commonly apply this strategy [Dréau et al., 2010; Legrain et al., 2012]. Typically, this approach is used with linear simplices. However, the method can also be extended to higher-orders as shown in [Cheng and Fries, 2009; Sevilla et al., 2008, 2011a,b,c]. The applicability of these approaches in the context of fictitious domain methods is demonstrated in [Fries and Omerović, 2015; Kudela, 2013; Kudela et al., 2015, 2016; Legrain, 2013; Stavrev, 2012; Stavrev et al., 2016].

10.1.2 Boundary conditions

One challenge in the context of embedded domain methods is the imposition of boundary conditions. Since the boundary is *not* resolved by edges of the finite cells, a classical imposition over the element edges is not possible. Accordingly, alternative approaches have to be applied. The following two sub-sections summarize the most prevalent techniques.

10.1.2.1 Neumann boundary conditions

Like in the classical FEM, homogeneous Neumann boundary conditions need no special treatment in the FCM. In the case of in-homogeneous Neumann boundary conditions, the respective boundary integral term on the right hand side of the weak form has to be computed. Since the mesh does not resolve the boundary, this integration cannot be performed over the facets of the elements as in conventional finite elements. Instead, an explicit surface discretization is introduced on which the traction integral can be evaluated. Just as the previously described integration mesh, this *surface integration mesh* is independent of the actual solution mesh and does not introduce additional degrees of freedom. A convenient way to generate this mesh is to provide the surface of interest in the Surface Tesselation Language (STL). The STL format is supported by most CAD systems and represents surfaces as a collection of linear triangles over which the integration can be performed.

10.1.2.2 Dirichlet boundary conditions

In contrast to the Neumann case, the incorporation of Dirichlet boundary conditions is more challenging. As discussed in Section 2.2, these are classically imposed by explicitly choosing the shape functions from kinematically admissible spaces, such that the prescribed values are met on the boundary by definition [Szabó and Babuška, 1991]. This imposes the boundary conditions in the strong sense.

A similar approach can be followed in the case of non-boundary conforming discretizations by adapting the admissible function spaces for the particular geometry under consideration.

Popular methods following this idea are for example web- and i-spline based methods (see e.g. [Höllig et al., 2001; Sanches et al., 2011]).

An alternative approach followed in this work is to extend the weak formulation such that it directly incorporates the Dirichlet boundary conditions. Possible strategies for this purpose are for example the Lagrange Multiplier or Penalty Method [Gerstenberger and Wall, 2010; Schillinger et al., 2012c; Zhuang, 2010; Zienkiewicz et al., 2005a]. Alternatively, an approach originally introduced by Nitsche [1971] for the Laplace problem can be used, which was later extended to other applications [Bazilevs and Hughes, 2007; Embar et al., 2010; Fernández-Méndez and Huerta, 2004; Hansbo et al., 2005]. The idea of this method is to extend the original weak form by additional constraining expressions as follows:

$$\tilde{a}\left(\boldsymbol{u},\boldsymbol{v}\right) = a\left(\boldsymbol{u},\boldsymbol{v}\right) - \int_{\Gamma_D} \left(\boldsymbol{\sigma}\left(\boldsymbol{u}\right)\cdot\boldsymbol{n}\right)\cdot\boldsymbol{v}\,\mathrm{d}\Gamma - \int_{\Gamma_D} \left(\boldsymbol{\sigma}\left(\boldsymbol{v}\right)\cdot\boldsymbol{n}\right)\cdot\boldsymbol{u}\,\mathrm{d}\Gamma + \beta\int_{\Gamma_D}\boldsymbol{v}\cdot\boldsymbol{u}\,\mathrm{d}\Gamma \tag{10.3}$$

$$\text{and}\quad \tilde{f}\left(\boldsymbol{v}\right) = f\left(\boldsymbol{v}\right) - \int_{\Gamma_D} \left(\boldsymbol{\sigma}\left(\boldsymbol{v}\right)\cdot\boldsymbol{n}\right)\cdot\hat{\boldsymbol{u}}\,\mathrm{d}\Gamma + \beta\int_{\Gamma_D}\boldsymbol{v}\cdot\hat{\boldsymbol{u}}\,\mathrm{d}\Gamma. \tag{10.4}$$

The first additional integral of the bi-linear form follows naturally from the divergence theorem when the test functions are not restricted to be homogeneous on the boundary. Therefore, it ensures the method's consistency. The second additional term of $\tilde{a}$ and the first additional term of $\tilde{f}$ are introduced to yield a symmetric stiffness matrix. The two remaining integrals are introduced for numerical stabilization, since coercivity might be lost when subtracting the consistency terms from the bi-linear form. With this extension, the discrete problem can be reformulated incorporating the Dirichlet boundary conditions in the weak sense as follows:

$$\text{Find } \boldsymbol{u_h} \in \mathcal{V}\left(\Omega\right) \quad \text{such that} \quad \tilde{a}\left(\boldsymbol{u_h},\boldsymbol{v_h}\right) = \tilde{f}\left(\boldsymbol{v_h}\right) \quad \forall\, \boldsymbol{v_h} \in \mathcal{V}\left(\Omega\right). \tag{10.5}$$

In analogy to Neumann boundary conditions, the constraining expressions are evaluated using a separate surface integration mesh.

The major advantage of Nitsche-like methods is their inherent consistency, which ensures that the solution of the original strong problem also solves the modified weak problem [Arnold et al., 2001; Embar et al., 2010; Fernández-Méndez and Huerta, 2004]. However, the major challenge of this approach is to select the stability parameter β large enough to guarantee the coercivity of the modified weak form. As shown in [Bazilevs and Hughes, 2007; Embar et al., 2010; Fernández-Méndez and Huerta, 2004], the correct value of β is dependent on the mesh-size h, the polynomial degree p, and the material parameters. In [Embar et al., 2010; Fernández-Méndez and Huerta, 2004; Hansbo et al., 2005] it is suggested to solve an auxiliary eigenvalue-problem to estimate β.

10.2 Numerical examples

10.2.1 Rotating ellipse example[a]

The aim of this first example is to assess whether the multi-level *hp*-approach and the Finite Cell Method can be combined to capture the temperature distribution around moving objects

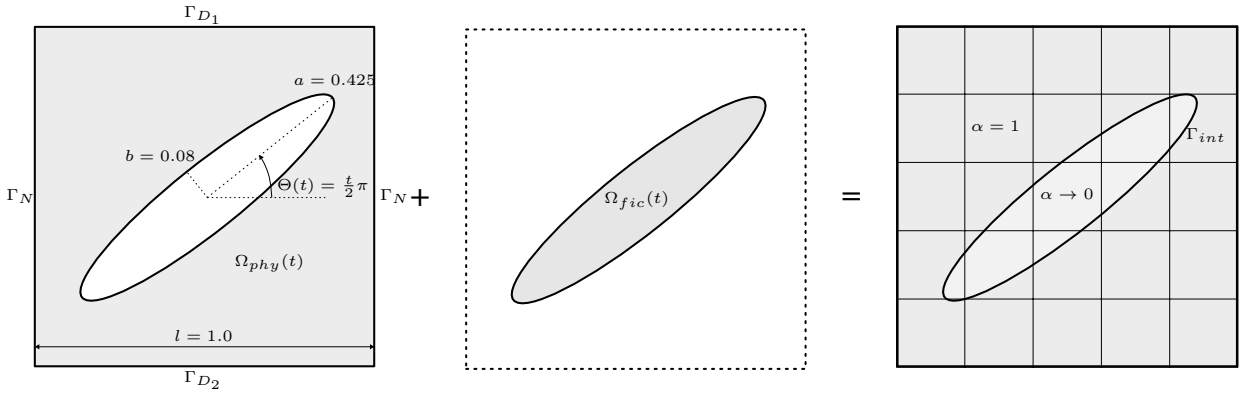


Figure 10.2: Setup of rotating ellipse example in the context of the Finite Cell Method [Zander et al., 2015].

on a fixed background mesh in a two-dimensional setting. For this purpose, the setup depicted in Figure 10.2 is considered. As illustrated, the physical domain has a rectangular shape in which a non-conducting inclusion in the form of an ellipsoid is embedded. With time, the inclusion rotates about the center point of the outer square. To simulate the heat flux around the rotating ellipse, the following model problem is considered

$$\frac{\partial \phi}{\partial t} = \Delta \phi \qquad \forall\, \boldsymbol{x} \in \Omega \text{ and } t \in [0, 2] \tag{10.6a}$$

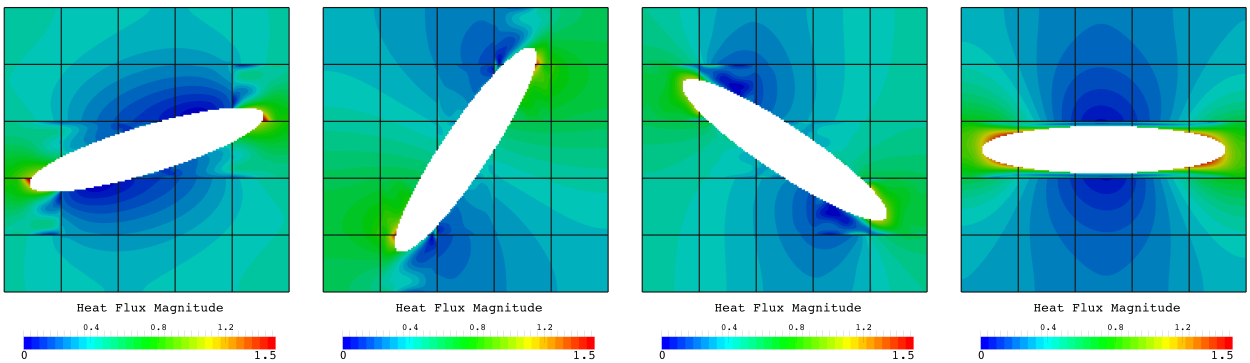
$$\phi = 1 \qquad \forall\, \boldsymbol{x} \in \Gamma_{D_1} \text{ and } t \in (0, 2] \tag{10.6b}$$

$$\phi = 0 \qquad \forall\, \boldsymbol{x} \in \Gamma_{D_2} \text{ and } t \in (0, 2] \tag{10.6c}$$

$$\nabla \phi \cdot n = 0 \qquad \forall\, \boldsymbol{x} \in \Gamma_N \text{ and } t \in (0, 2] \tag{10.6d}$$

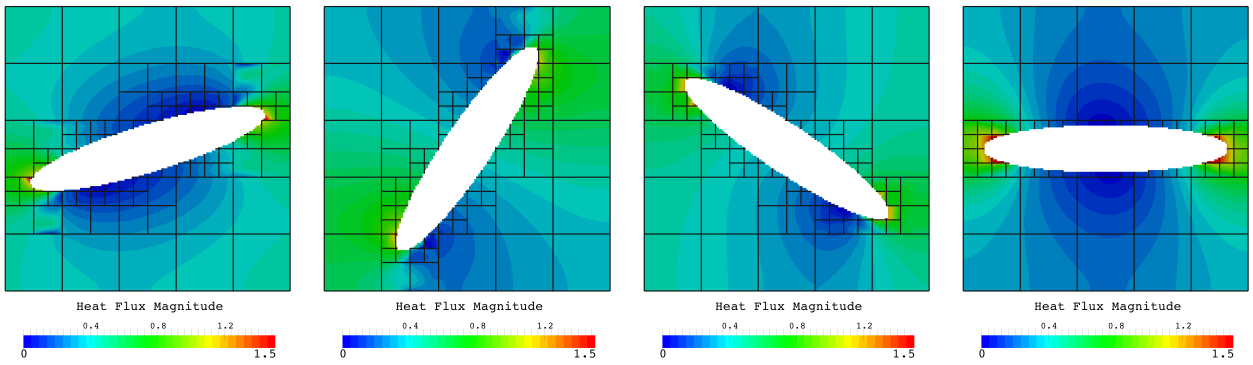
$$\phi = 0 \qquad \forall\, \boldsymbol{x} \in \Omega \text{ and } t = 0. \tag{10.6e}$$

For the temporal discretization, a backward Euler approach with 32 equidistant time steps is used. The spatial discretization follows the concept of the Finite Cell Method. As illustrated in Figure 10.2, a coarse mesh of 5×5 elements partitions the square domain without resolving



(a) $t = 10\%$ (b) $t = 30\%$ (c) $t = 80\%$ (d) $t = 100\%$

Figure 10.3: Heat flow around the rotating ellipse, no refinement, $p = 4$ [Zander et al., 2015].



(a) $t = 10\%$ (b) $t = 30\%$ (c) $t = 80\%$ (d) $t = 100\%$

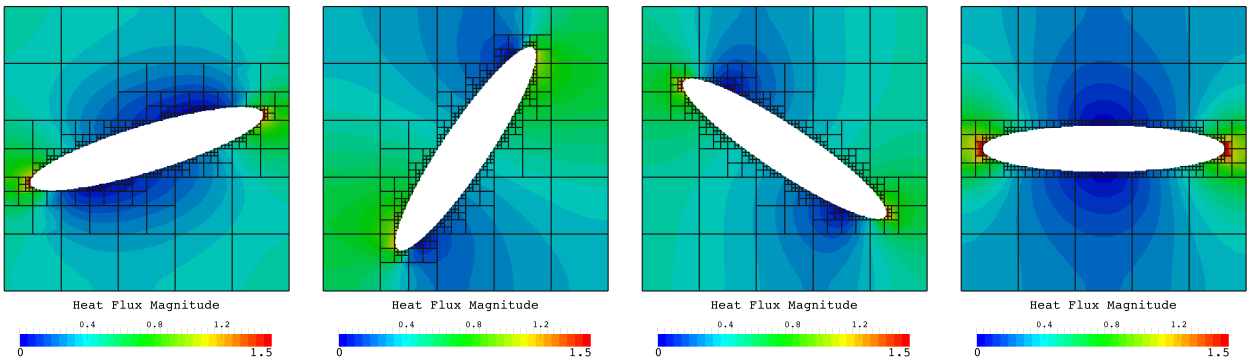
Figure 10.4: Heat flow around the rotating ellipse with $p = 4$ and 2 levels of *hp-d*-refinements [Zander et al., 2015].

the inclusion. On this non-boundary confirming mesh, the FCM indicator function α recovers the original problem. Within the embedded ellipse, the value of α is set to 10^{-10} to account for the non-conductive nature of the inclusion. Outside of the ellipse, α is set to one. For the numerical integration of the physical domain, a quad-tree of depth five is used. The resulting integration mesh is updated in every time step to resolve the motion of the ellipse.

The numerical approximation obtained with this discretization is depicted in Figure 10.3. The results show that the coarse finite cell mesh can capture the global solution state correctly. However, the approximation suffers from oscillations in the elements cut by the ellipse. The reason for this problem is that the shape of the ellipse is slim compared to the size of the elements. Due to this design of the benchmark, most coarse elements intersect the physical domain on both sides of the ellipse. Hence, the high-order shape functions of the coarse elements have to approximate two independent solution states that might differ significantly. This complexity yields the oscillations. Accordingly, the discretization has to be refined locally and dynamically around the embedded boundary of the rotation ellipse. For this purpose, the suggested superposition-based refinement scheme is used. To account for the motion of the ellipse, the depth of the element refinement trees is updated in every time-step of the simulation.

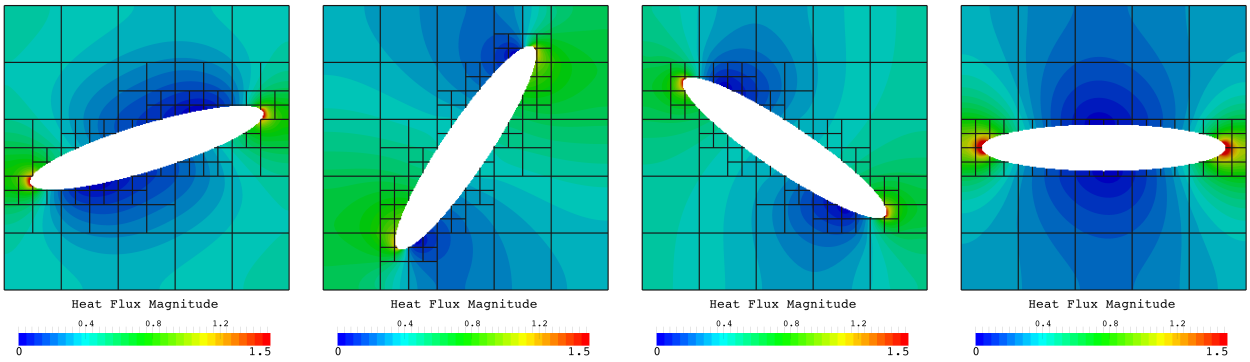
In a first step, the *hp-d*-method with two levels of refinement is used. As shown in Figure 10.4, the refinement has a beneficial effect on the result quality, but the oscillations remain present. In analogy to the two previous chapters, the reason for this is the piece-wise linear nature of the overlay solution. Since the high-order shape functions remain active on the coarse base mesh, their support is not decreased. Accordingly, the original problem remains present as the high-order shape functions still have support in the physical domain on both sides of the ellipse. Even increasing the number of overlay meshes yields only a minor improvement of the result quality as illustrated in Figure 10.5.

The situation is fundamentally different when using the multi-level *hp*-version of the overlay refinement as depicted in Figure 10.6. Since the high-order shape functions are moved to the overlay mesh, their support is decreased. Therefore, the high-order shape function have no longer support on both sides of the ellipse. This solves the original problem. Accordingly, the



(a) $t = 10\%$ (b) $t = 30\%$ (c) $t = 80\%$ (d) $t = 100\%$

Figure 10.5: Heat flow around the rotating ellipse with $p = 4$ and 4 levels of *hp*-*d*-refinements [Zander et al., 2015].



(a) $t = 10\%$ (b) $t = 30\%$ (c) $t = 80\%$ (d) $t = 100\%$

Figure 10.6: Heat flow around the rotating ellipse with $p = 4$ and 2 multi-level *hp*-refinements [Zander et al., 2015].

artificial oscillations vanish even when using only two levels of refinement.

10.2.2 Embedded moving ellipsoid

The objective of this second example is to study whether the combination of the multi-level *hp*- and the Finite Cell Method is able to capture the solution state around a moving object also in the context of a three-dimensional setting. To this end, the heat conducting cube depicted in Figure 10.7 is considered. Within the cube, an insulator in the shape of an ellipsoid is embedded. As illustrated in Figure 10.7, the inclusion moves from bottom to top with time.

As in the previous two-dimensional example, the transient heat flux around this inclusions is analyzed by solving the transient, linear heat equation on this domain

$$\frac{\partial \phi}{\partial t} = \Delta \phi \qquad \forall\, \boldsymbol{x} \in \Omega \text{ and } t \in (0, T]. \tag{10.7}$$

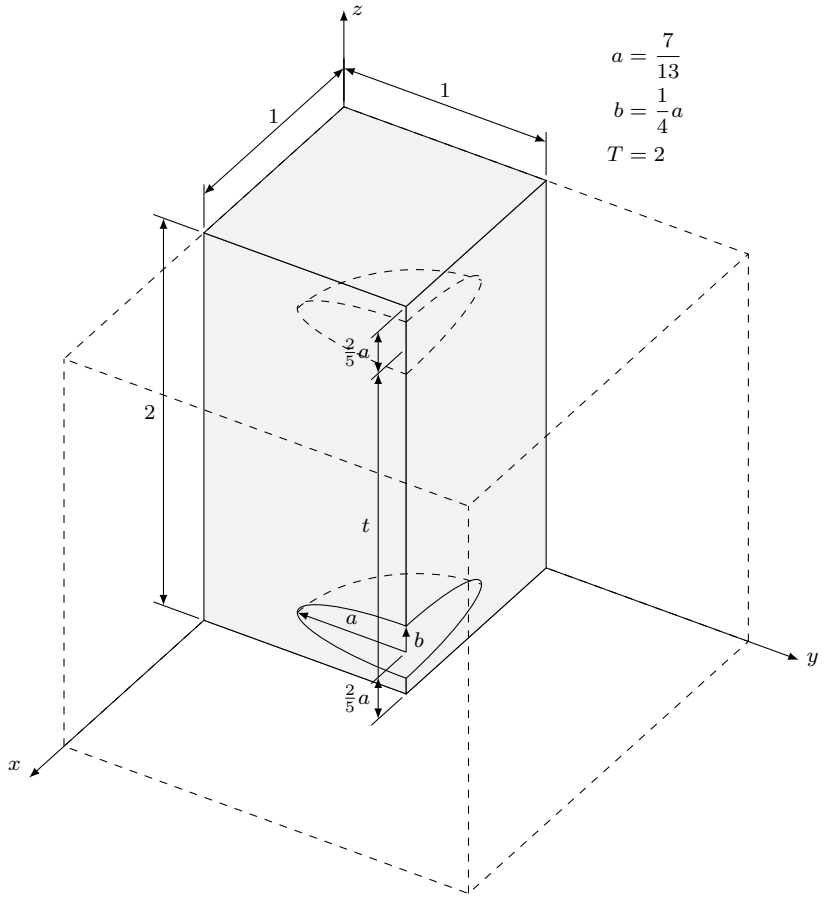


Figure 10.7: Setup of embedded moving ellipsoid: a heat-conducting cube of size $2 \times 2 \times 2$ is considered. On the bottom and top of the cube, the temperature value is fixed to zero and an one, respectively. Within the cube, an insulator in the shape of an ellipsoid is embedded. The inclusion moves from bottom to top with time. Because of the symmetry of the problem, only the shaded quadrant of the cube is considered.

On the bottom and top of the cube, the temperature value is fixed to zero and one, respectively

$$\phi = 0 \quad \forall\, \boldsymbol{x} \in [0,2] \times [0,2] \times \{0\} \text{ and } t \in (0,T] \tag{10.8a}$$

$$\phi = 1 \quad \forall\, \boldsymbol{x} \in [0,2] \times [0,2] \times \{1\} \text{ and } t \in (0,T] \tag{10.8b}$$

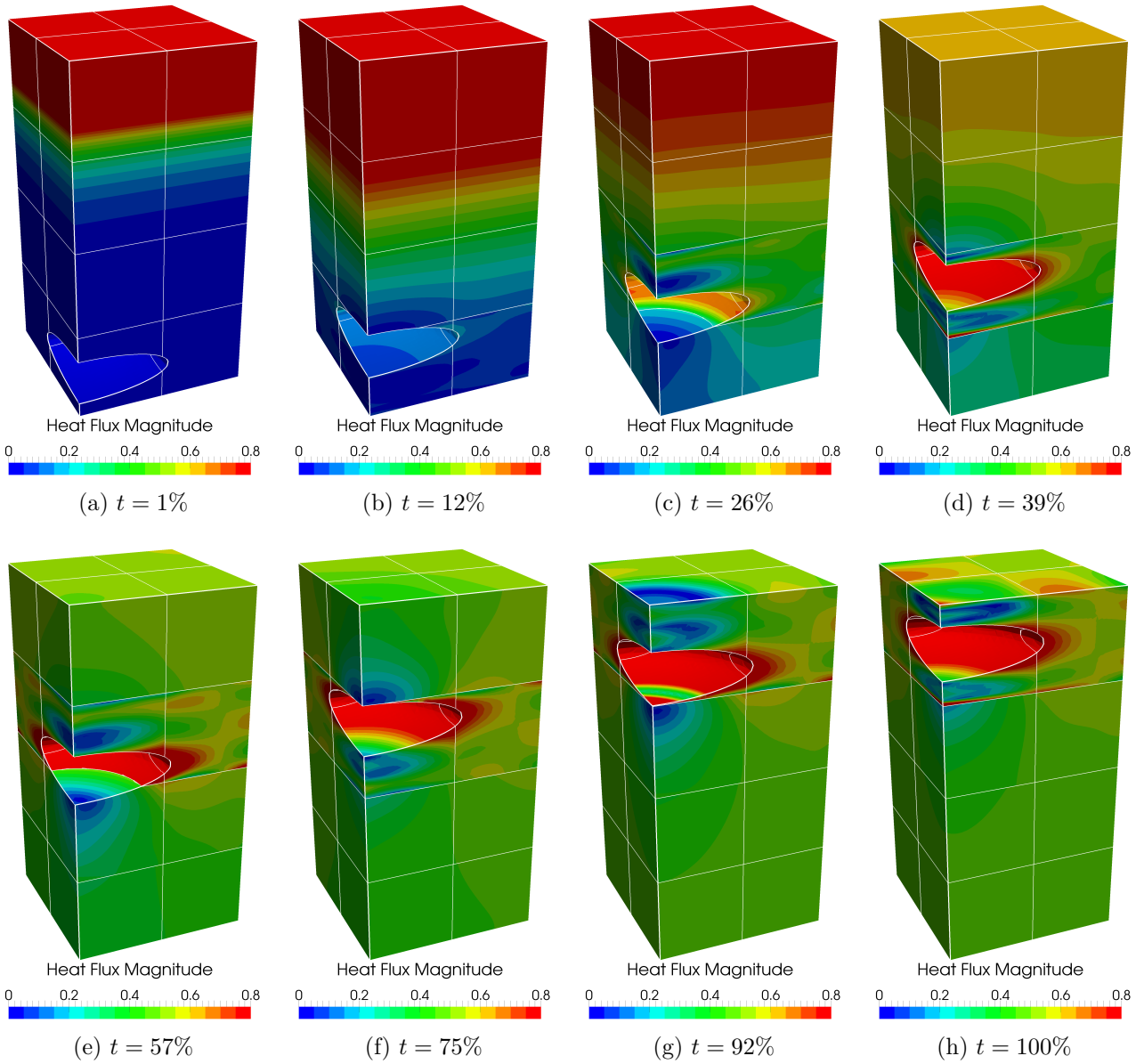
On the remaining boundaries, homogeneous Neumann boundary conditions are applied. The problem description is completed by a homogeneous initial condition

$$\phi = 0 \qquad \forall\, \boldsymbol{x} \in \Omega \text{ and } t = 0. \tag{10.9}$$

As in the previous example, the analytical solution is approximated using the backward Euler time-stepping scheme for the temporal and the Finite Cell Method for the spatial discretization. The time range is sub-divided in 64 equal time steps. The spatial discretization takes advantage of the problem's symmetry by only considering the quadrant shaded in Figure 10.7. In a first step, this reduced domain is partitioned using $2 \times 2 \times 4$ finite elements.

Following the concept of the Finite Cell Method, these elements do *not* resolve the embedded inclusion but only the surrounding cuboid. The original problem is recovered using the indicator function α. The non-conducting nature of the inclusion is modeled numerically by choosing $\alpha = 10^{-3}$ within the moving ellipsoid. In the remaining domain, α is set to one. For the numerical quadrature of the stiffness matrix, an octree integration scheme of depth three is used in the finite cells cut by the inclusion. To account for the motion of the ellipsoid, the integration tree is updated according to the current position of the inclusion in every time step.

The numerical approximation of the heat flux obtained with this discretization is depicted for different time states in Figure 10.8. The results confirm the findings of the previously



(a) $t = 1\%$ (b) $t = 12\%$ (c) $t = 26\%$ (d) $t = 39\%$

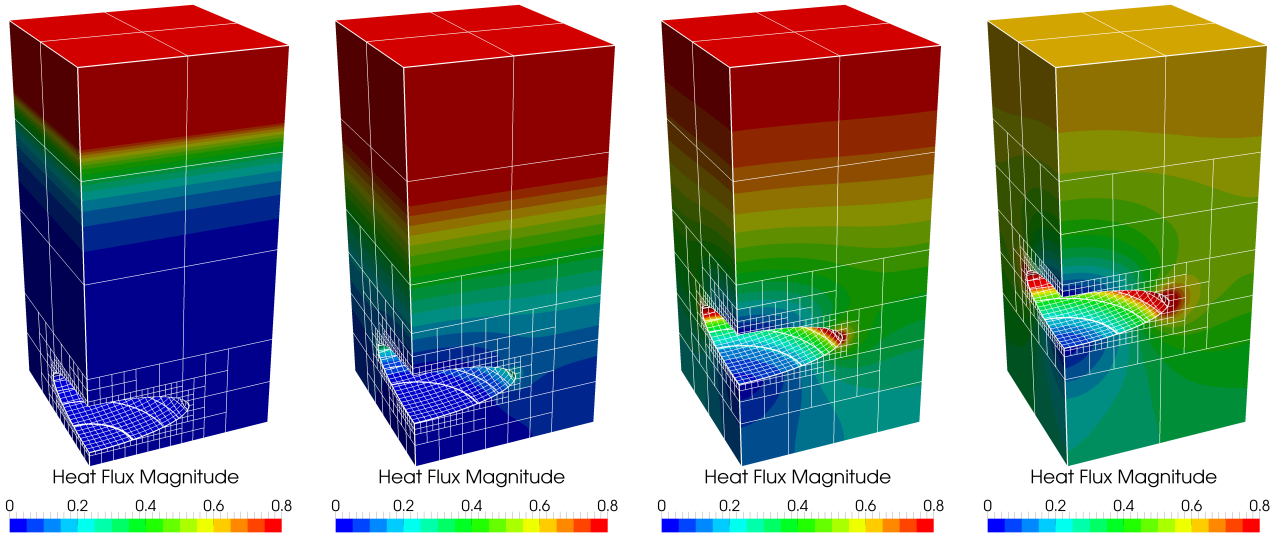
(e) $t = 57\%$ (f) $t = 75\%$ (g) $t = 92\%$ (h) $t = 100\%$

Figure 10.8: Heat flux around moving inclusion, no refinement, $p = 5$.

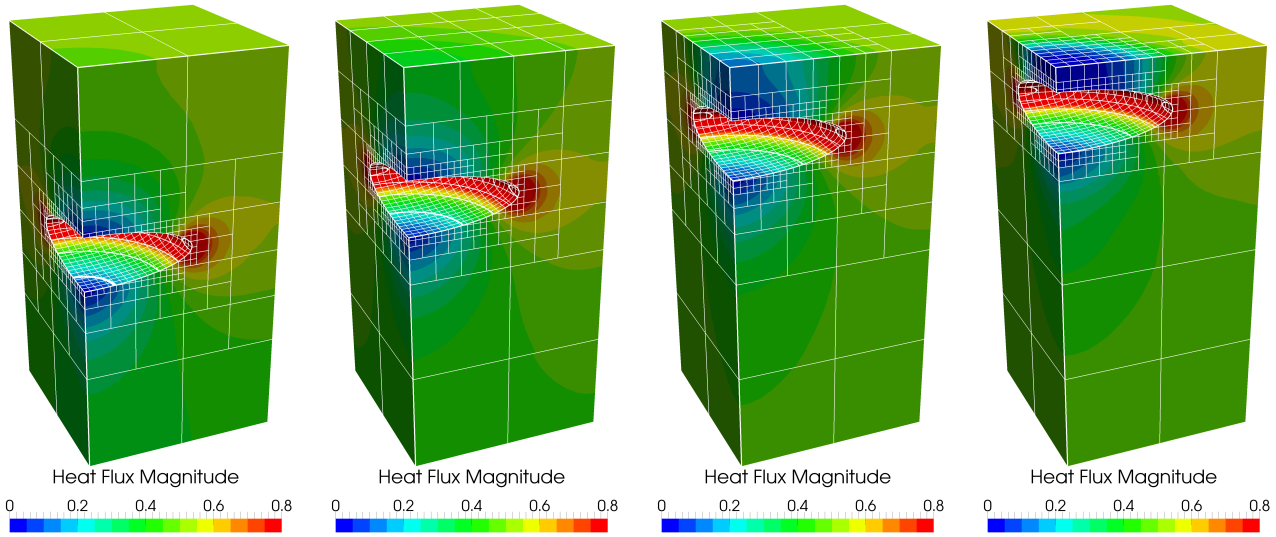
considered two-dimensional benchmark: the coarse finite cell mesh is able to capture the global solution state, but the numerical approximation is oscillatory in cells cut by the inclusion. As discussed in the context of the two-dimensional example, the cause of this problem is that some of the coarse elements intersect the physical domain on the top and bottom sides of the ellipsoid. Therefore, the used high-order shape functions have to account for two solution states that potentially differ significantly. Accordingly, the discretization has to be refined locally around the embedded boundary of the inclusion.

As observed in the context of the two-dimensional counterpart of this example, the use of the *hp-d*-method for the refinement of the discretization has only minor beneficial effects on the result quality since the support of the high-order shape functions remains unchanged. For this reason, the multi-level *hp*-technique is used. Accordingly, the aforementioned coarse base discretization is refined by superposing four levels of overlay meshes in the vicinity of the embedded boundary. The refinement is guided geometrically by recursively sub-dividing every element that is cut by the boundary of the embedded ellipse. To account for the motion of the inclusion, the refinement tree of every element is updated dynamically in every time step of the simulation according to the current position of the ellipsoid.

The approximation to the analytical heat flux obtained with this discretization is depicted for different time states in Figure 10.9. The results demonstrate that the decrease of the shape function's support achieved by the multi-level *hp*-refinement effectively prevents oscillations. This confirms that the high approximation quality of the multi-level *hp*-method also carries over to the simulation of moving inclusions embedded in three-dimensional structures.



(a) $t = 1\%$ (b) $t = 12\%$ (c) $t = 26\%$ (d) $t = 39\%$



(e) $t = 57\%$ (f) $t = 75\%$ (g) $t = 92\%$ (h) $t = 100\%$

Figure 10.9: Heat flux around moving inclusion, 4 levels of multi-level *hp*-refinement, $p = 5$.

Chapter 11

Application to cohesive delamination[h]

In many engineering and industrial applications, the accurate prediction of crack initiation and propagation is of major importance to prevent catastrophic failure. As demonstrated in Section 8.2.2, the suggested multi-level *hp*-refinement method requires only a small number of unknowns to capture the complex stress state ahead of a crack tip with high accuracy. The aim of this chapter is to assess whether this high-approximation capabilities can also be used to capture the initiation and the propagation of a cracks.

For this purpose, the numerical scheme has to be extended by a failure model. Depending on the properties of the used material, the fracture mechanism might be brittle along a sharp interface or occur within a fracture process zone, in which the material gradually loses its strength. The current work focuses on this second fracture mechanism.

A common approach to capture this kind of failure is the use of cohesive zone models first introduced in [Dugdale, 1960] and [Barenblatt, 1962]. The basic idea of these models is to describe the cohesive forces in the fracture process zone with respect to the crack opening using a traction-separation law. In [Hillerborg et al., 1976], this modeling strategy was combined with the finite element method to simulate crack initiation and propagation numerically. Over the decades, this approach has proven its potential in diverse fields of application such as void nucleation [Needleman, 1987], quasi-static crack growth [Needleman, 1990a,b; Tvergaard and Hutchinson, 1992], and e.g. dynamic crack growth [Pandolfi et al., 2000; Ruiz et al., 2001; Xu and Needleman, 1994]. In particular, in cases where the potential crack path is known in advance as in e.g. laminated structures, cohesive zone models yield their full potential since they can be incorporated easily in the finite element formulation [de Borst and Remmers, 2006; de Borst et al., 2004; Geubelle and Baylor, 1998; Schellekens and de Borst, 1993a]. A review about the basic principles of the method is given e.g. in [Planas et al., 2003].

Despite their advantages, the use of cohesive zone models is numerically challenging due to their "intrinsic discretization sensitivity" [Samimi et al., 2011a]. In particular, the cohesive zone ahead of the crack requires sufficiently small elements to prevent artificial, discretization-induced oscillations in the load-displacement curve as reported in numerous works [Alfano and Crisfield, 2001; Crisfield and Alfano, 2002; Dimitri et al., 2014; Guiamatsia et al., 2009; Harper and Hallett, 2008; Hermes, 2010; Mi et al., 1998; Nguyen and Nguyen-Xuan, 2013; Samimi et al., 2009, 2011a,b; Tomar et al., 2004; Turon et al., 2007; Wang and Waisman, 2015].

Studies presented in [Harper and Hallett, 2008; Hermes, 2010; Tomar et al., 2004; Turon et al., 2007; Wang and Waisman, 2015] show that the cohesive zone must be resolved with

at least two to five elements when using linear shape functions to achieve reasonable results. Resolving the whole computational domain with such a fine resolution is prohibitively expensive. Hence, the mesh is typically refined locally along the crack path, given that it is known *a priori*.

The task of further improving the discretization performance is still a topic of active research. Recently, B-spline and NURBS basis functions have been used to resolve the progressing crack with high accuracy [Hosseini, 2014; Hosseini et al., 2015; Nguyen et al., 2014; Verhoosel et al., 2011]. The positive effect of higher-order shape functions on the artificial, discretization-induced oscillations is studied in [Nguyen and Nguyen-Xuan, 2013]. In [Dimitri et al., 2014], the approach has been further extended by using T-splines to refine the discretization non-uniformly along the crack path.

However, the possible crack paths can become large and spoil an efficient analysis. Laminated wing structures, for example, have dimensions of several meters whereas a high approximation accuracy is needed only in the vicinity of the cohesive failure zone of a few millimeters. Consequently, an *a priori* refinement along the full crack path adds a large number of unnecessary additional degrees of freedom to the discrete system.

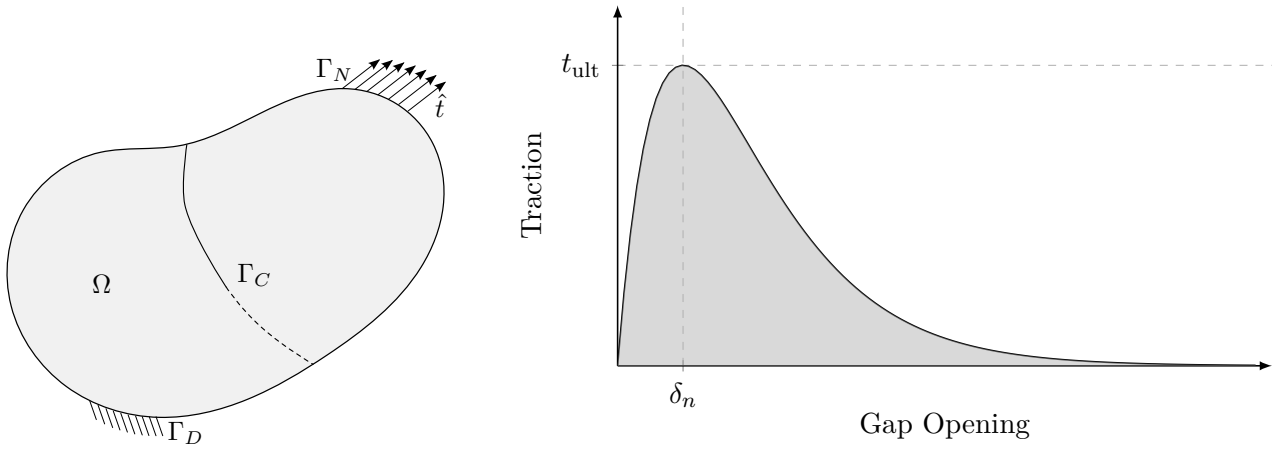
To avoid this problem, alternative approaches have been developed allowing for a locally bounded refinement zone that follows the crack tip. Common strategies use adaptive hierarchical enrichment [Crisfield and Alfano, 2002; Guiamatsia et al., 2009], self-adaptive finite elements [Samimi et al., 2011a,b], adaptive h-refinement [Park et al., 2012], or error-driven adaptive re-meshing [Khoei et al., 2012]. All these approaches significantly increase the accuracy of the approximation at the crack tip. However, they are restricted to an approximation with linear shape functions. This demands for a fine base mesh to capture the global deformation accurately.

The high approximation quality of *hp*-finite elements seem to be a natural choice to overcome efficiently the problem of artificial, discretization-induced oscillations in cohesive fracture propagation. However, the moving crack front requires a continuously adaptive change of the discretization throughout the simulation to ensure a sufficient locally refined cohesive process zone. Following the conventional *hp*-idea, where coarse elements are replaced during runtime by a finer mesh, is challenging and demands for a very sophisticated discretization kernel [Šolín et al., 2010a; Šolín and Korous, 2012a,b].

As demonstrated in the previous two chapters, the suggested multi-level version of *hp*-FEM is well capable of adapting the mesh throughout the runtime of the simulation. The aim of this chapter is to analyze whether these dynamic discretization features can be used to resolve the cohesive zone ahead of the crack tip by a local mesh refinement that follows the propagating crack. To this end, the first section provides a review of the essential ideas of the cohesive zone modeling. In the second section, the capability of the suggested multi-level *hp*-refinement approach is tested in the context of different two- and three-dimensional benchmarks.

11.1 Numerical simulation of cohesive crack growth

Within the cohesive zone model, the crack is modeled as a sharp interface immersed in the physical domain Ω. As illustrated in Figure 11.1a, the path Γ_C along which the crack potentially evolves is assumed to be known in advance. This is reasonable when considering e.g. the delamination of composite structures or cracks along material interfaces. Across the



(a) Within the cohesive zone model, the crack is modeled as an interface Γ_C that is embedded in the elastic domain Ω.

(b) Traction-separation curve of exponential Xu-Needleman cohesive zone law introduced in Xu and Needleman [1993].

Figure 11.1: Cohesive crack growth [Zander et al., 2017].

cracked interface, the displacement $\boldsymbol{u}$ may be discontinuous yielding a crack opening $[\![\boldsymbol{u}]\!]$. The resulting gap can be split into a normal and tangential part $[\![u_n]\!]$ and $[\![\boldsymbol{u}_s]\!]$, respectively. Pure normal separation is referred to as mode I fracture, whereas the shear separations represent sliding fracture (mode II) and tearing fracture (mode III). In the following, the equations governing the cohesive crack growth and the use of the finite element method for the numerical simulation of this process are discussed.

11.1.1 Governing equations

Assuming linear elasticity, the deformation of the continuum is described by the following set of partial differential equations:

$$\nabla \cdot \boldsymbol{\sigma} + \hat{\boldsymbol{b}} = \boldsymbol{0} \qquad \text{on } \Omega \tag{11.1a}$$

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon} \qquad \text{on } \Omega \tag{11.1b}$$

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\top}\right) \qquad \text{on } \Omega \tag{11.1c}$$

$$\boldsymbol{u} = \hat{\boldsymbol{u}} \qquad \text{on } \Gamma_D \tag{11.1d}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \hat{\boldsymbol{t}} \qquad \text{on } \Gamma_N \tag{11.1e}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t}([\![\boldsymbol{u}]\!]) \qquad \text{on } \Gamma_C. \tag{11.1f}$$

In analogy to the pure elastic model discussed in Section 2.1, $\boldsymbol{\sigma}$ describes the stress tensor, $\hat{\boldsymbol{b}}$ the prescribed body force, $\mathbb{C}$ the elasticity tensor, $\boldsymbol{\varepsilon}$ the linear strain tensor, $\boldsymbol{u}$ the displacement vector, $\boldsymbol{n}$ the surface normal vector, $\hat{\boldsymbol{u}}$ the prescribed displacement on the Dirichlet boundary Γ_D, and $\hat{\boldsymbol{t}}$ the prescribed surface traction on the Neumann boundary Γ_N.

The traction-separation law $\boldsymbol{t}([\![\boldsymbol{u}]\!])$ in (11.1f) extends the conventional elasticity problem to account for a cohesive crack analysis. This second constitutive equation models the load-

carrying capabilities of the partially opened crack with respect to the crack opening $[\![\boldsymbol{u}]\!]$. A number of different traction-separation laws are discussed in the literature that are tuned to the respective application under consideration. A comprehensive review is given in e.g. [Turon et al., 2007; van den Bosch et al., 2006].

This work considers the so-called exponential cohesive zone law introduced in [Xu and Needleman, 1993] and later improved in [van den Bosch et al., 2006]. As in most models, the interface traction is split into its normal and shear components t_n, t_{s_1} and t_{s_2}, respectively. The studies presented in [Harper and Hallett, 2008; Yang and Cox, 2005] show that the cohesive zone of a mode I fracture typically is significantly smaller than for a mode II or a mode III failure. Therefore, the analysis presented in this work focuses on pure mode I failure since it is the computationally more challenging fracture type and thus serves as a severe test case for the proposed refinement scheme. Accordingly, the tangential gap opening and the tangential traction in the interface are assumed to be zero. Following e.g. [van den Bosch et al., 2006], the interface traction can be expressed as

$$t_n = \frac{G_c}{\delta_n}\frac{[\![u_n]\!]}{\delta_n}\exp\left(-\frac{[\![u_n]\!]}{\delta_n}\right) \quad \text{with } \delta_n = \frac{G_c}{t_{\text{ult}}\exp(1)}. \tag{11.2}$$

The depiction of this traction-separation law in Figure 11.1b reveals a gradual increase of the traction between the surfaces of the crack interface until the limit strength t_{ult} of the material is reached. Beyond this point, the fracture strength gradually decays to zero modeling the cohesive behavior of the fracture. The area enclosed by the traction-separation curve represents the energy release per unit of cracked surface and is denoted as fracture toughness G_c.

11.1.2 Finite element discretization

The numerical simulation of the crack propagation is based on the weak form of the governing partial differential equation (11.1):

$$\begin{aligned}&\text{Find } \boldsymbol{u} \in \mathcal{V}, \text{ such that}\\ &\int_\Omega \delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma}\,\mathrm{d}\Omega + \int_{\Gamma_C} \delta[\![\boldsymbol{u}]\!]\cdot\boldsymbol{t}([\![\boldsymbol{u}]\!])\,\mathrm{d}\Gamma = \int_\Omega \delta\boldsymbol{u}\cdot\hat{\boldsymbol{b}}\,\mathrm{d}\Omega + \int_{\Gamma_N}\delta\boldsymbol{u}\cdot\hat{\boldsymbol{t}}\,\mathrm{d}\Gamma \quad \forall\,\delta\boldsymbol{u}\in\mathcal{V}_0,\end{aligned} \tag{11.3}$$

with the functions spaces being defined as

$$\mathcal{V} = \left\{\boldsymbol{v} : v_i \in H^1(\Omega), v_i|_{\Gamma_D} = \hat{u}_i\right\} \quad \text{and} \quad \mathcal{V}_0 = \left\{\boldsymbol{v} : v_i \in H^1(\Omega), v_i|_{\Gamma_D} = 0\right\}, \tag{11.4}$$

and H^1 denoting the Sobolev space of degree one.

The prevalent approach to solve the above problem numerically is the finite element method. Within this work, it is assumed that the (potentially closed) crack interface Γ_C separates the original domain Ω into two parts. Each of these sub-domains is discretized using a separate finite element mesh. For simplicity, these meshes are designed to be matching geometrically along Γ_C. Using this discretization, the displacement field $\boldsymbol{u}$ of the two sub-domains is given by the product of the shape function matrix $\boldsymbol{N}$ and the degree of freedom vector $\hat{\boldsymbol{u}}$:

$$\boldsymbol{u}_i = \boldsymbol{N}_i \cdot \hat{\boldsymbol{u}}_i, \tag{11.5}$$

where the sub-index i of the above terms labels the two different domains. In analogy, the strain resulting from the displacement is given by:

$$\boldsymbol{\varepsilon}_i = \boldsymbol{B}_i \cdot \hat{\boldsymbol{u}}_i, \tag{11.6}$$

where $\boldsymbol{B}$ denotes the strain-displacement matrix [Bathe, 2007]. The current gap value $[\![\boldsymbol{u}]\!]$ is computed as the difference between the displacements of the two sub-domains:

$$[\![\boldsymbol{u}]\!] = \boldsymbol{u}_1 - \boldsymbol{u}_2 = \boldsymbol{N}_1 \cdot \hat{\boldsymbol{u}}_1 - \boldsymbol{N}_2 \cdot \hat{\boldsymbol{u}}_2 = \begin{bmatrix} \boldsymbol{N}_1 & -\boldsymbol{N}_2 \end{bmatrix} \cdot \begin{bmatrix} \hat{\boldsymbol{u}}_1 \\ \hat{\boldsymbol{u}}_2 \end{bmatrix} = \boldsymbol{H} \cdot \hat{\boldsymbol{u}}, \tag{11.7}$$

$$\text{with} \quad \boldsymbol{H} = \begin{bmatrix} \boldsymbol{N}_1 & -\boldsymbol{N}_2 \end{bmatrix} \quad \text{and} \quad \hat{\boldsymbol{u}} = \begin{bmatrix} \hat{\boldsymbol{u}}_1 \\ \hat{\boldsymbol{u}}_2 \end{bmatrix}. \tag{11.8}$$

With this kinematic relations, the weak form (11.3) can be translated into the following system of non-linear equations

$$\boldsymbol{f}_{\text{int}}(\hat{\boldsymbol{u}}) = \boldsymbol{f}_{\text{ext}}, \quad \text{with} \quad \boldsymbol{f}_{\text{int}}(\hat{\boldsymbol{u}}) = \int_{\Omega} \boldsymbol{B}^{\top} \boldsymbol{\sigma} \, \mathrm{d}\Omega + \int_{\Gamma_C} \boldsymbol{H}^{\top} \boldsymbol{t}([\![\boldsymbol{u}]\!]) \, \mathrm{d}\Gamma \tag{11.9}$$

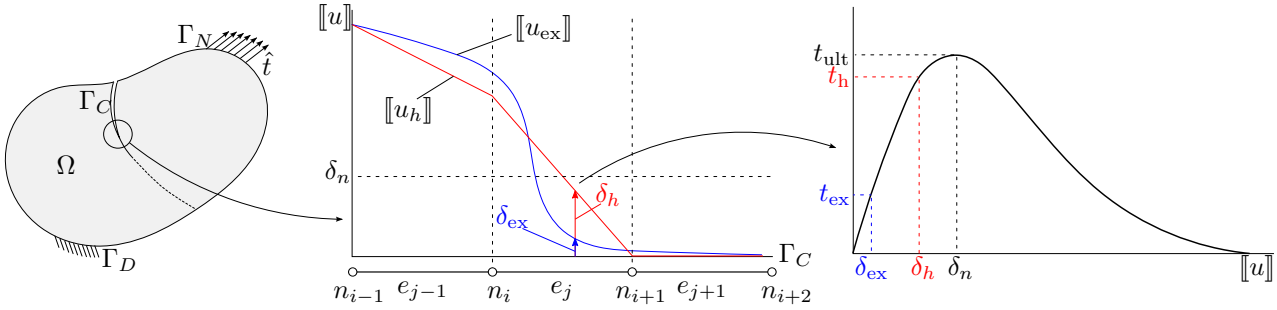
$$\text{and} \quad \boldsymbol{f}_{\text{ext}} = \int_{\Omega} \boldsymbol{N}^{\top} \hat{\boldsymbol{b}} \, \mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{N}^{\top} \hat{\boldsymbol{t}} \, \mathrm{d}\Gamma. \tag{11.10}$$

This non-linear equation system is solved using a Newton-Raphson iteration scheme.

A common approach to account for the contribution of the progressing crack is to introduce additional, zero-thickness cohesive interface elements along Γ_C, whose kinematic is given by $\boldsymbol{H}$, *cf.* [Schellekens and de Borst, 1993b]. In this work, a different approach is applied. Instead of defining a new element type with kinematics specific to the gap opening, the virtual work of the crack opening is computed by numerically integrating the interface contribution along Γ_C. To this end, the crack interface is partitioned into a set of integration domains that conform to the boundaries of the finite elements along Γ_C. On each of these integration segments, an appropriate number of Gauss-Legendre quadrature points is used to compute numerically the contribution of the progressing crack to the tangential stiffness matrix and the residual vector. This approach originates from the idea of weak mesh coupling [Guo and Ruess, 2015b; Ruess et al., 2014]. It is also similar to the Gauss-point-to-surface contact algorithm, which was used recently in [Dimitri et al., 2014] to simulate interface de-bonding. Mathematically, the used approach is equivalent to the conventional idea of interface elements. However, it has the advantage that it separates the computation of the interface contributions from the actual discretization. This simplifies the refinement process as no kinematics of refined interface elements has to be formulated.

11.1.3 Discretization induced oscillations

The solution procedure of the non-linear system of equations is numerically challenging since the cohesive interface approach suffers from an intrinsic discretization sensitivity as reported in e.g. [Alfano and Crisfield, 2001; Crisfield and Alfano, 2002; Dimitri et al., 2014; Guiamatsia et al., 2009; Harper and Hallett, 2008; Hermes, 2010; Mi et al., 1998; Samimi et al., 2009, 2011a,b; Tomar et al., 2004; Turon et al., 2007; Wang and Waisman, 2015]. This sensitivity is characterized by unphysical oscillations in the global load-displacement curve, which are caused by a cyclic stiffening phenomenon as sketched in the following, *cf.* Figure 11.2:



(a) cracked domain (b) interface crack opening (c) traction vs. crack opening

Figure 11.2: Discrete cohesive crack opening: linear approximation error [Zander et al., 2017].

- assume that in load step k the crack arrived at a node n_i of an undamaged element e_j, *cf.* Figure 11.2b. In the following load step increment $k+1$, the node n_i will be displaced such that the crack propagates through e_j.
- using a linear displacement approximation, the crack spreads over the complete element domain. Thus, in large parts of element e_j, the numerically derived gap opening δ_h is significantly larger than the analytical opening δ_{ex} as depicted in Figure 11.2b.
- due to the higher stiffness of the interface material in the pre-cracked regime, *cf.* Figure 11.2c, the opening over the complete finite element requires a higher force than a local propagation of the crack, i.e. $t_h \gg t_{\text{ex}}$. This renders the discrete interface stiffer than in reality. Therefore, the element remains closed until the load is further increased, followed by an abrupt opening and a corresponding energy release. Thus the crack advances in discrete steps governed by the element size.

In between these steps, the high force required to break the interface causes a correspondingly large deformation of the elastic structure. This unrealistic deflection decreases again once the interface element has failed. The unphysical deformation cycles yield oscillations in the global load-displacement curve and may cause a series of snap-through and snap-back effects. These effects render the numerical solution of the non-linear system significantly more difficult and typically demand for arc-length based path-following techniques.

The artificial oscillations become more pronounced with increasing element size. Thus, the accurate simulation of the crack propagation demands for a local refinement of the discretization in the vicinity of the cohesive zone.

11.1.4 Multi-level *hp*-refinement for cohesive crack growth

As demonstrated in the previous chapters, the refine-by-superposition concept applied by the multi-level *hp*-method allows for a dynamic change of the discretization during the simulation runtime. Within this chapter, this high flexibility shall be used to keep the refinement zone local to the crack tip, while it is propagating through the domain. To this end, the refinement tree of every element is updated according to the current position of the crack in every load step of the non-linear solution scheme.

The normal gap opening $[\![u_n]\!]$ across the interface Γ_C serves as a natural refinement criterion. To mark the elements for refinement, equidistant sample points are distributed in every element

edge along Γ_C. The number of sampling points is chosen in dependence of the polynomial degree of the adjacent elements. On every sampling point, the normal gap value $[\![u_n]\!]$ is computed according to (11.8). If the computed gap opening is within a range $[g_1, g_2]$, the leaf-elements adjacent to the current point are marked for refinement given that a pre-defined refinement depth is not exceeded. If the gap opening is larger than g_2 for all sampling points of an element edge, the respective element is marked for being coarsened. In this way, the refinement tree builds up recursively and refines the mesh only around the crack tip.

The values of g_1 and g_2 are chosen empirically such that $g_1 < \delta_n < g_2$, where δ_n is the gap opening of maximal cohesive traction *cf.* (11.2). The examples presented in the next section demonstrate that the refinement region resolves the fracture zone sharply when choosing g_1 about two orders of magnitude smaller and g_2 about one order of magnitude larger than δ_n. The specific values of g_1 and g_2 used for the different examples are given in the description of the respective example setup.

In addition to the refinement of the element size h, also the choice of the approximation order p is important for the result quality. As discussed before, the overlay refinement approach allows to place the high-order modes on different levels of the refinement tree. For this reason, the first question to be analyzed in the example section of this chapter is which p-distribution yields the highest approximation quality for the simulation of cohesive crack propagation.

It is to be expected that the cohesive zone is resolved best when the high-order modes are placed on the leaf-elements. This can be seen by extending the simple residual problem discussed in Section 8.1.1 for the present case of delamination. This is done in Figure 11.3a. Here, a standard quadratic base discretization is refined by adding an additional linear overlay mode. Hence, the final numerical solution is composed of a linear contribution u_1 and a quadratic contribution u_2:

$$u_h = u_1 + u_2, \tag{11.11}$$

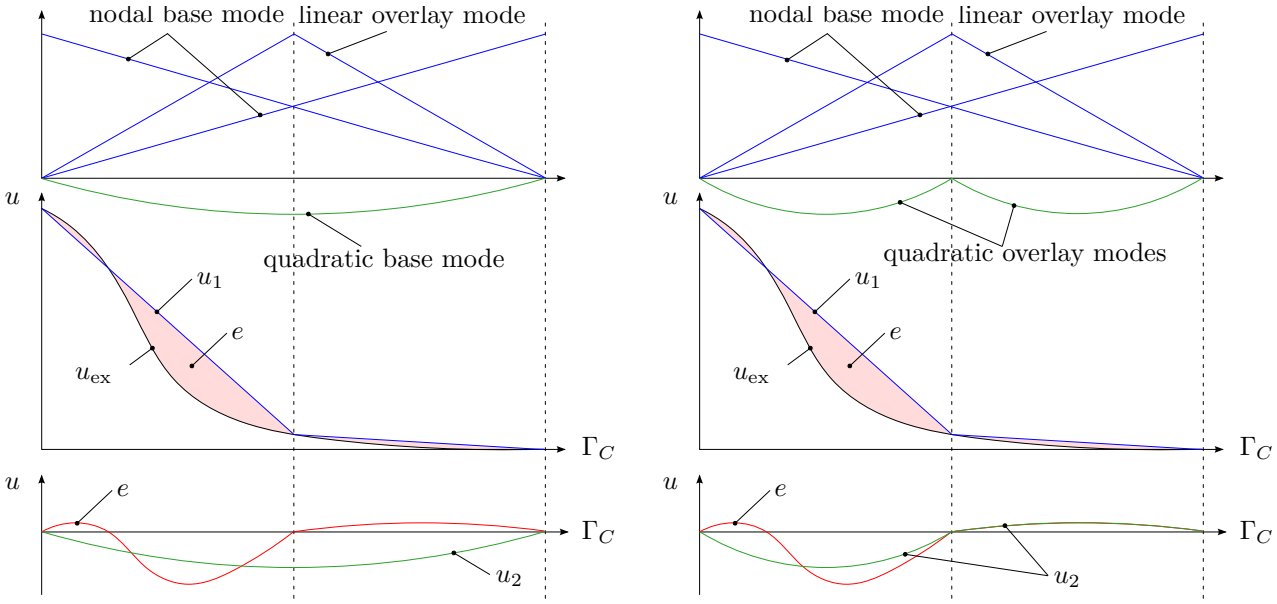
both having support on the full base element.

When the crack arrives at the considered element, these two contributions approximate the complex gap function $[\![u_{\text{ex}}]\!]$ that forms the transition from the closed to the fully opened crack. Using only the linear contribution for approximation results in an error

$$e = u_{\text{ex}} - u_1. \tag{11.12}$$

In the considered scenario, the residual is close to zero on the right half-interval but non-zero on the left. The remaining deviation cannot be compensated by the quadratic mode u_2 as this shape function spans the complete *coarse* element. Therefore, the left and the right part of the solution cannot be approximated independently. Instead, the solution scheme has to find the best-fit compromise.

As illustrated in Figure 11.3b, the situation is fundamentally different if the base-element is refined by placing the high-order modes on the leaf-elements. In this case, the discretization features two quadratic overlay modes, whose support is restricted to the left and right part of the base element, respectively. Therefore, these two modes can approximate the two parts of the solution at a much higher accuracy. For the present case, this allows to capture the right-hand-side residual with high accuracy, which confines the approximation error to the left element.



(a) Refinement by linear overlay: the high-order shape functions remains on the base element.

(b) Refinement by high-order overlay: the high-order shape functions are shifted to the leaf elements.

Figure 11.3: Approximation of solution by overlay-refinement with $p = 2$ [Zander et al., 2017].

To verify this assumption, the influence of the element size h, the approximation order p and the overlay configuration on the result quality is systematically studied in the following examples.

11.2 Numerical examples

In the following, the performance of the proposed multi-level hp-refinement method is studied with several examples. In particular, the potential of h-, p-, and hp-refinement for preventing discretization-induced oscillations is of interest. To this end, the first benchmark analyzes the approximation quality of the different discretizations using the well known double cantilever beam benchmark. In the second example, the proposed refinement scheme is used in a three-dimensional simulation of a two-ply laminate. The section closes by analyzing the potential of the finite cell method to model cohesive crack propagation within complex domains. All examples concentrate on pure mode I delamination as it is the numerically most challenging type of failure [Harper and Hallett, 2008; Yang and Cox, 2005].

11.2.1 Double cantilever beam benchmark

The objective of this fist example is to demonstrate that the proposed refinement scheme efficiently prevents an unphysical snap-through and snap-back behavior of the load-displacement curve. To this end, the double cantilever beam (DCB) benchmark problem is chosen, which has been studied before in e.g. [Alfano and Crisfield, 2001; Crisfield and Alfano, 2002; Dimitri et al., 2014; Guiamatsia et al., 2009; Harper and Hallett, 2008; Samimi et al., 2011b; Turon

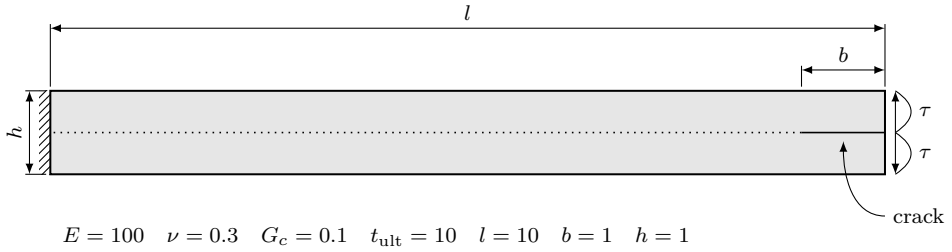


Figure 11.4: Setup of a double cantilever beam benchmark problem [Zander et al., 2017].

et al., 2007; Wang and Waisman, 2015]. The problem is reduced to two dimensions by applying plane strain conditions. The geometry and the material properties are depicted in Figure 11.4. The two beams are loaded by shear tractions applied to the front face of the two beams as illustrated in Figure 11.4. The applied shear traction is distributed quadratically over the height of the two beams such that the load is zero at the upper and lower boundaries of the two beams and thus consistent.

The cohesive layer between the two beams is modeled using the exponential cohesive zone law introduced in [Xu and Needleman, 1993] (see Equation (11.2) and Figure 11.1b). As the aim of this example is to demonstrate that the proposed refinement scheme works reliably under extreme conditions, the material parameters are chosen to be extremely brittle. To follow the expected snap-through and snap-back phenomena, which are commonly present for this type of analysis, a dissipation-based arc-length method as proposed in [Verhoosel et al., 2009] is used.

A first analysis model is composed of 640 linear elements over the beam length and 10 elements over the height of each beam. In Figure 11.5, the load-displacement curve of the beam tip obtained with this discretization is depicted. The results show that using a sufficiently fine discretization, the curve is smooth and free of numerical artifacts. However, already doubling the element size along the horizontal axis introduces a series of minor snap-through and snap-back effects. Reducing the resolution even further to 160 linear elements results in severe discretization-induced oscillations, which render the result unphysical. Even more, as the crack propagates and the stiffness of the beam decreases, the oscillation cycles tilt more and more horizontal. As a consequence, the number of load steps significantly increases compared to the initially fine discretization model.

These results demonstrate the intrinsic dependency of the cohesive zone model on the discretization. In the following, the effects of global p-refinement and local hp-refinement are studied. To this end, the load-displacement curves obtained from different discretization models are compared in Figure 11.6.

The study begins with considering a uniform p-refinement of the model. For this purpose, the discretization is coarsened from 320 to 160 elements along the beam length. To compensate the larger element size, the polynomial order of the shape functions is increased from $p = 1$ to $p = 2$. Thus, the total number of unknowns remains unchanged. However, the numerical stiffness of the system changes as the higher-order elements can better represent the quadratic shear stress. This results in a parallel shift of the load-displacement curve, *cf.* Figure 11.6a, which in the mean is closer to the sufficiently converged reference solution shown in Figure 11.6f. Increasing the polynomial degree further to $p = 3$ reduces the oscillation effect noticeably. Furthermore,

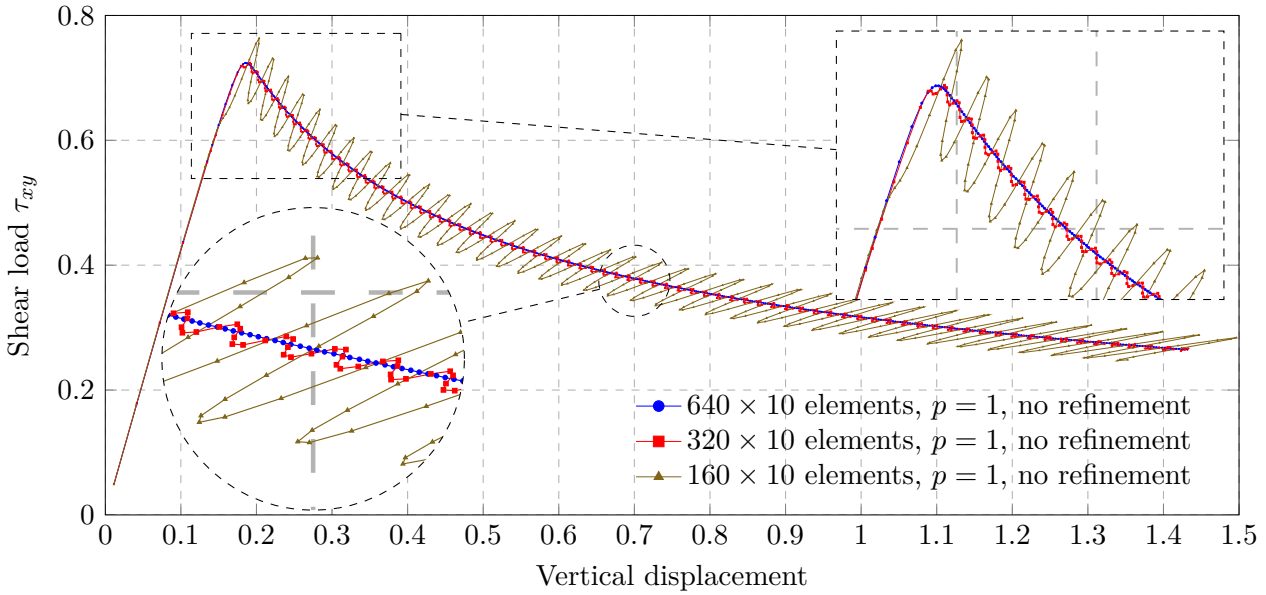


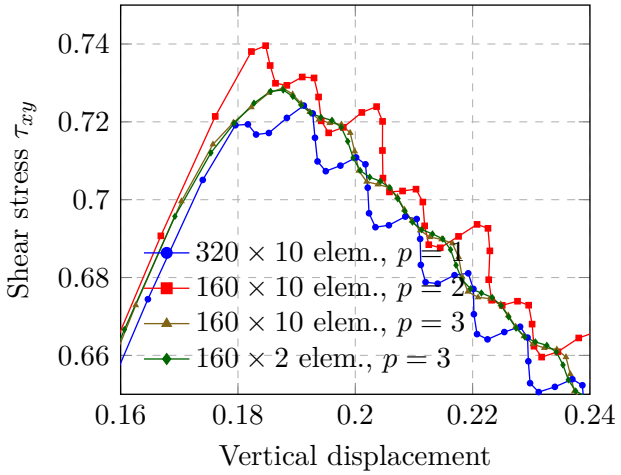
Figure 11.5: Global load-displacement curve of beam tip [Zander et al., 2017].

the higher order shape functions allow a reduction of the number of elements over the beam height to two elements without compromising the accuracy of the computation. In this way, the number of unknowns significantly reduces with immediate effect on the computational performance. For this reason, the following higher-order computations are performed with only two elements over the height of the beam.

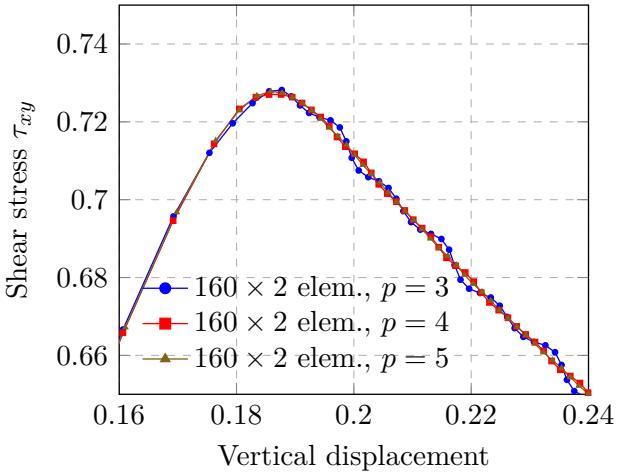
The results depicted in Figure 11.6b show that a further order elevation significantly reduces the amplitude of the unphysical oscillations. Similar positive effects of higher order discretizations on the stability of the solution scheme have been reported in the context of isogeometric analysis in e.g. [Dimitri et al., 2014]. However, this positive effect has its limits as shown in Figure 11.6c: further coarsening of the mesh from 160 to 80 elements reveals the dependency of the solution behavior on the element size even for a higher approximation order. This demonstrates that an accurate and efficient representation of the crack propagation needs both, a small element size and high-order shape functions. However, this refined discretization is only needed in the direct vicinity of the failure zone, i.e. the crack tip. All other parts of the analysis domain may fully profit from larger high-order elements providing an accurate stress representation.

For this reason, the proposed *hp*-adaptive scheme is applied to refine the mesh locally in the vicinity of the crack tip. As outlined in Section 6.2, the gap opening computed in the last load step serves as a refinement criterion. All elements along the crack interface whose deformation results in a normal gap opening that is larger than $g_1 = 10^{-4}$ and smaller than $g_2 = 5 \cdot 10^{-2}$ are refined. The refinement is performed up to a pre-defined refinement depth. If the gap opening is larger than g_2 along the complete edge of an element, the element is coarsened again until the original base mesh is reached.

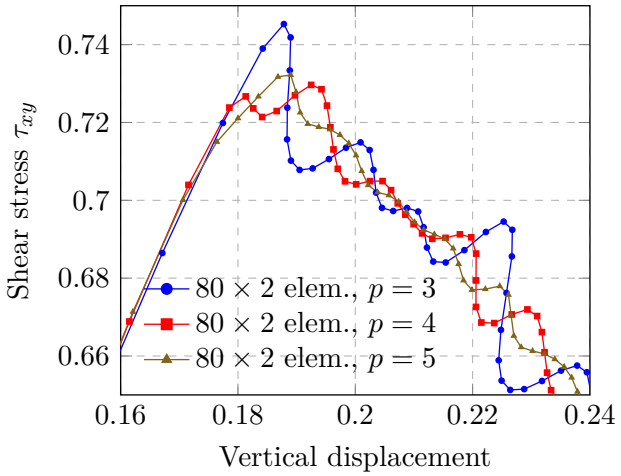
In the study, depicted in Figure 11.6d, a base discretization of 160×2 quadratic elements is used. In combination with one level of refinement, the element size at the crack tip thus corresponds to a discretization using 320×2 quadratic elements. However, using the *hp-d-*



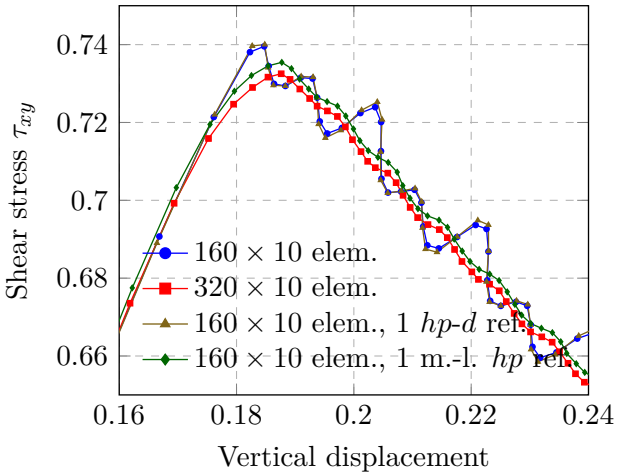
(a) Global p-refinement on fine discretization (part 1)



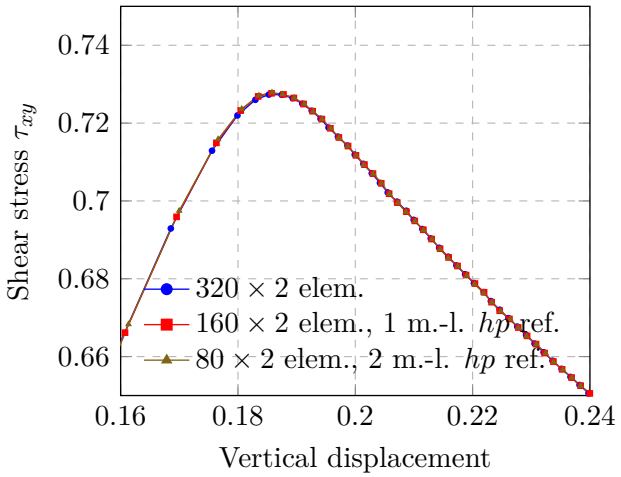
(b) Global p-refinement on fine discretization (part 2)



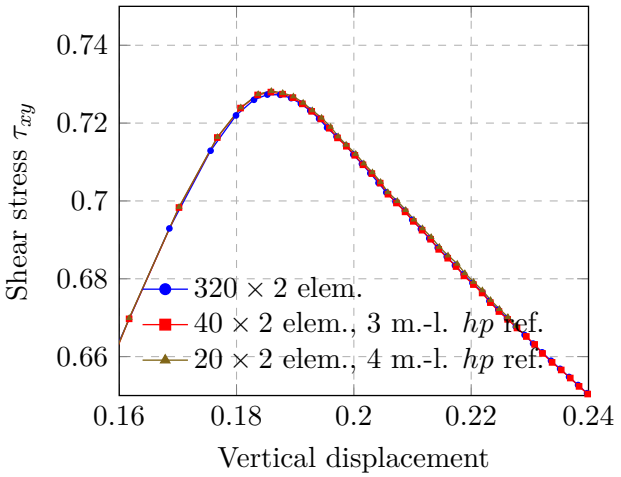
(c) Global p-refinement coarse discretization



(d) Local hp-d and multi-level hp-refinement for $p = 2$

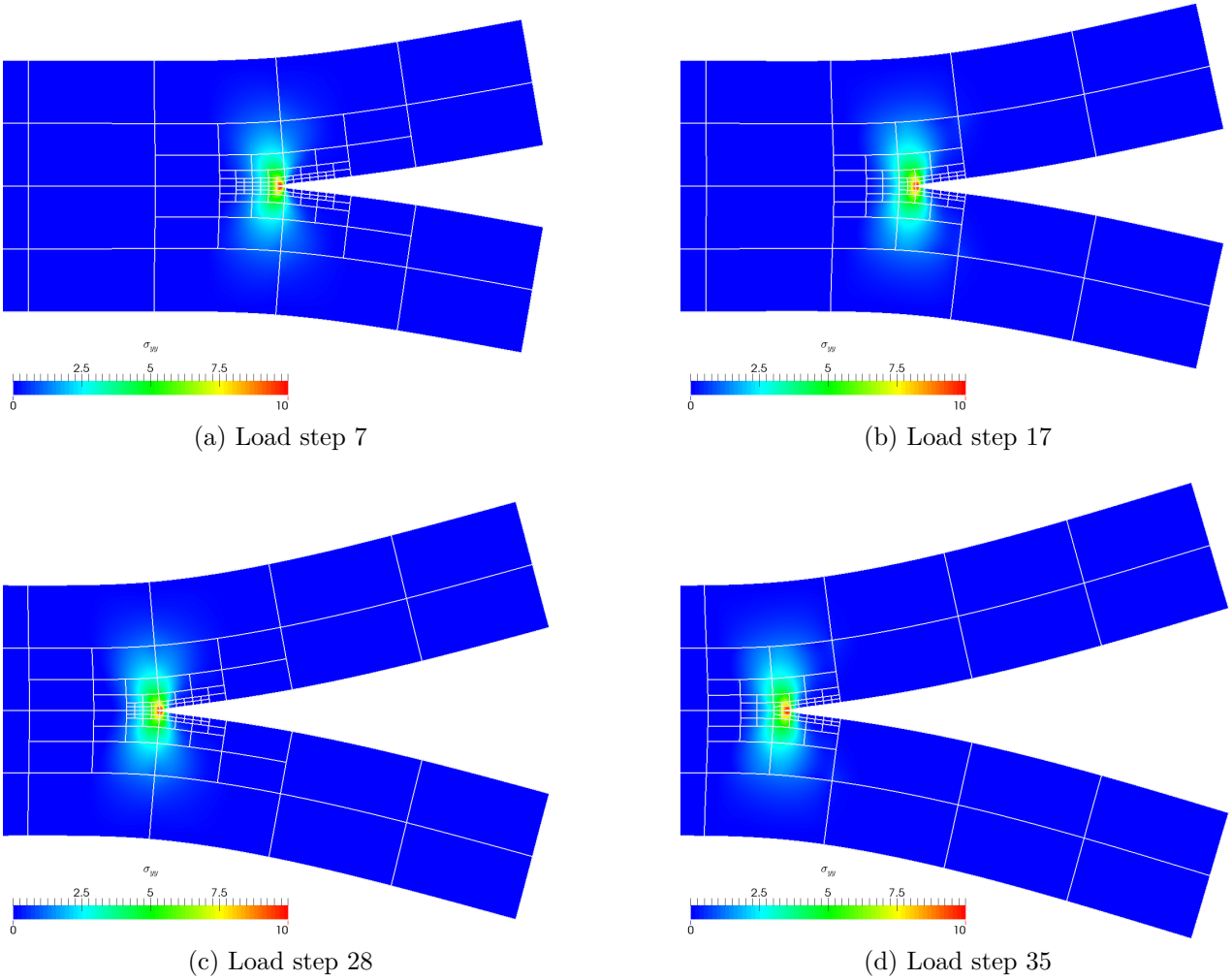


(e) Local multi-level hp-refinement for $p = 3$ (part 1)



(f) Local multi-level hp-refinement for $p = 3$ (part 2)

Figure 11.6: Comparison of load curves at limit point for different discretizations (used abbreviations: elem. = elements, ref. = refinement(s), m.-l. hp = multi-level hp) [Zander et al., 2017].



(a) Load step 7

(b) Load step 17

(c) Load step 28

(d) Load step 35

Figure 11.7: Stress distribution in deformed beam at different load steps (20×2 elements, $p = 3$, 4 levels of multi-level *hp*-refinement) [Zander et al., 2017].

refinement approach shows no visible improvement compared to the results obtained from the coarse, unrefined discretization. If instead the base mesh is refined by the multi-level *hp*-approach, the magnitude of the oscillation reduces to the same size obtained by the globally fine discretization of 320×2 quadratic elements. This analysis shows that the correct capturing of the propagating crack front requires high-order shape functions on the *smallest* support, which can be achieved using the multi-level *hp*-formulation. This confirms the expectations of Section 11.1.4.

The results obtained by the global and the local multi-level *hp*-refinement only differ in a slight parallel shift of the curves. This is again due to the different stiffness of the numerical systems caused by the inability of the quadratic elements to correctly capture the shear stress in the beam. For this reason, the study is repeated using cubic shape functions. The results depicted in Figure 11.6e demonstrate an excellent agreement with the reference solution based on a globally refined mesh.

No. of elements	p	No. of refinements	Max. no. of dofs	Relative runtime [%]
640 × 10	1	0	28,204	122.1
320 × 2	3	0	26,908	100.0
160 × 2	3	1	14,428	41.1
80 × 2	3	2	8,236	24.5
40 × 2	3	3	5,188	20.1
20 × 2	3	4	3,388	19.4

Table 11.1: Comparison of the number of unknowns and the computational time using the adaptive refinement. The runtime measurements were performed on a desktop workstation using a single core of an Intel® Core™ i7-4790 CPU and the same finite element code framework. k: number of recursive refinements [Zander et al., 2017].

The further results shown in Figures 11.6e and f demonstrate that the same quality of results can be achieved by using even coarser base meshes when increasing the refinement depth accordingly. Even using only 20 elements over the beam length and four levels of refinements shows a virtually identical load curve.

Figure 11.7 exemplarily depicts the stress concentration ahead of the propagating crack tip for four different load steps. The computation is performed on a base mesh of 20×2 elements, which is refined four times towards the crack tip. The transition of fracture process zone through the elements takes place without oscillations in the stress state. As soon as the crack has traveled completely through an element domain, this element is coarsened again by simply reducing the depth of the refinement tree.

In Table 11.1, the maximum number of unknowns are listed for various discretizations with comparable accuracy. The comparison shows that already the use of higher-order bulk elements is superior to a fine, low-order mesh. In the present case, the dominating bending mode results in a quadratic distribution of the shear stress over the beam height. Thus, cubic elements are optimal in this example. Other applications, however, might benefit from even higher order shape functions. Furthermore, the results demonstrate that—compared to a refinement along the full crack path—the use of a dynamic local mesh refinement yields a significant reduction of the necessary degrees of freedom and the computational time without degrading the result quality. For the present case, using the suggested local refinement approach decreases the runtime by a factor of five. The expected gain in performance can even be higher when increasing the length of the beams while keeping the length of the cohesive zone unchanged.

11.2.2 Delamination in three-dimensional models

Based on the previous model, the following example considers an extension to three dimensions. The purpose of this example is to study the three-dimensional stress state along the crack front. To this end, the laminated plates depicted in Figure 11.8 are considered.

Again, the Xu-Needleman traction separation law is used to model the cohesive crack propa-

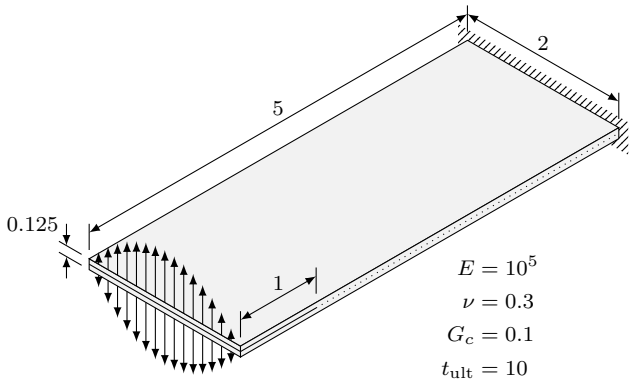


Figure 11.8: Double cantilever plate model [Zander et al., 2017].

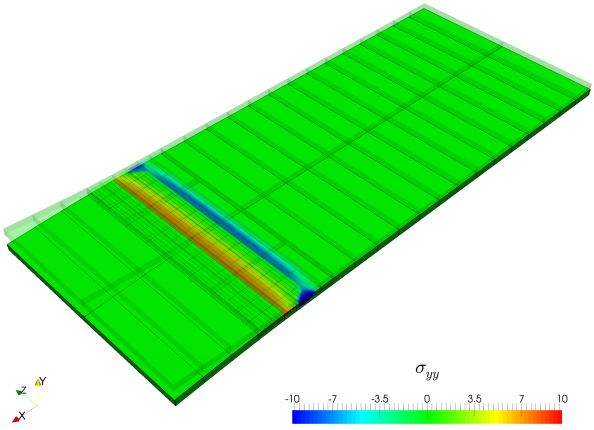
gation. In analogy to the previously discussed two-dimensional example, the loading is applied as a shear traction on the front faces of the two plates. To ensure consistent loading conditions, the load is applied in a bi-quadratic distribution that is zero on all four boundaries of the front faces.

Common discretization-approaches for the considered thin plates employ dimensionally reduced shell-elements, which are based on assumptions on the through-thickness displacement. A comprehensive review of this technology is given in e.g. [Bischoff et al., 2004]. In the present case, however, the crack front yields a complex stress state that cannot be incorporated easily in such models. For this reason, this element technology is not used here. Instead, the geometry is discretized with thin, volumetric, three-dimensional hexahedral elements of higher order. In [Rank et al., 2005], it was shown that such hierarchic high-order shape functions qualify well for the analysis of thin shell models. In particular, the high-order shape functions alleviate locking phenomena, which is a common issue for standard low-order shell elements. This allows for coarse, thin elements with a high aspect ratio.

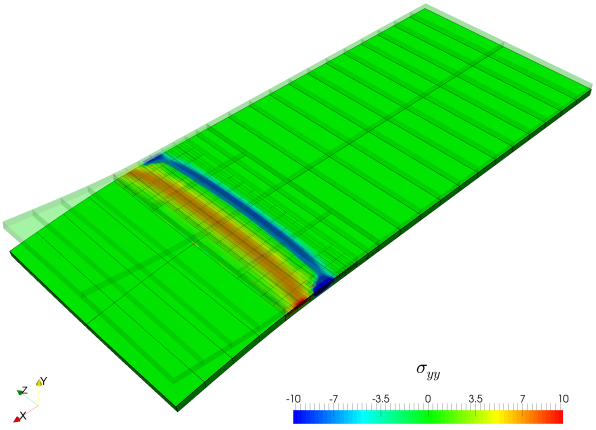
Following this approach, each plate is discretized using $1 \times 2 \times 20$ elements of order $p = 4$. This base mesh is refined locally around the crack front to resolve the cohesive zone accurately. As in the two-dimensional case, the gap opening serves as the criterion to decide about multi-level *hp*-refinement. The range where to refine is selected as $g_1 = 5 \cdot 10^{-6}$ and $g_2 = 10^{-2}$. Using three levels of adaptive refinement, this discretization yields up to 340,000 degrees of freedom. In contrast, a global refinement with a comparable element size would result in more than 8,200,000 unknowns, which demonstrates the significant reduction of computational effort gained by the local refinement.

As depicted in Figure 11.9, the *hp*-approach allows to resolve the cohesive zone ahead of the crack front while it is moving through the domain. In particular, the curvature of the crack front is captured automatically by the refinement. Furthermore, the refinement stays local to the crack as the scheme allows for arbitrary hanging nodes. It sharply resolves both the cohesive tension forces as well as the pressure reaction force. A zoom-in on the stress state is shown in Figure 11.9c. It demonstrates that at the crack front, the stress varies rapidly in the in-plane direction and through the thickness of the plates. The complexity of this stress state justifies the use of refined volumetric elements for the computation of the delamination process.

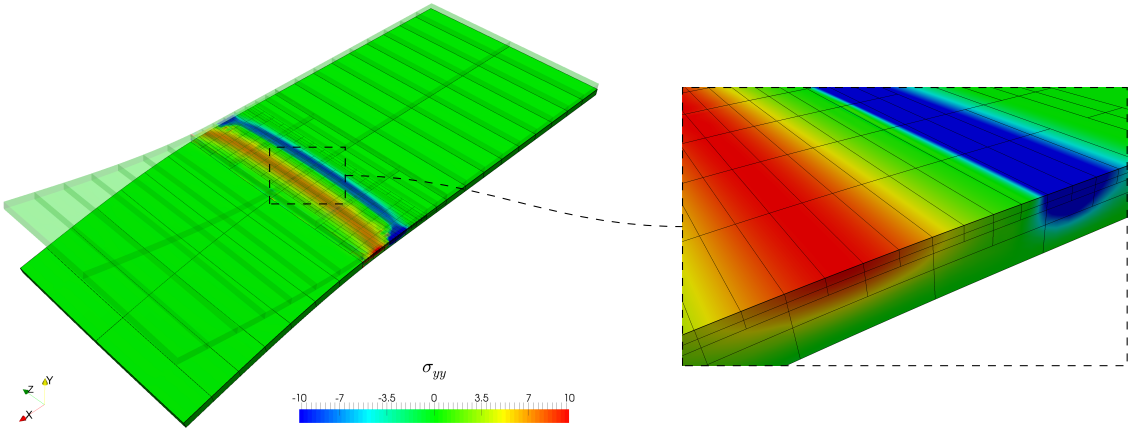
The depicted results also indicate a concentration of compressive stress at the boundary. This



(a) Load step 4 (249,630 degrees of freedom)



(b) Load step 15 (307,950 degrees of freedom)



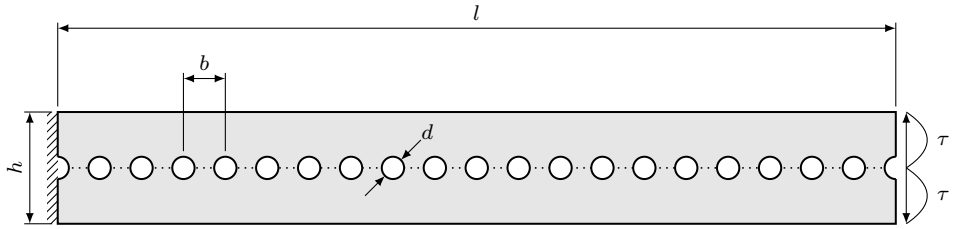
(c) Load step 31 (312,702 degrees of freedom) with zoom-in on the crack front. To visualize the internal stress state, the domain is clipped along the central axis.

Figure 11.9: Stress distribution in deflected beam for different load steps ($1 \times 2 \times 20$ elements, $p = 4$, three levels of refinement) [Zander et al., 2017].

phenomena is caused by the tilting of the cross-sections at these points yielding a localized contact of the two plates on the edges. Even the refined mesh is too coarse to capture the resulting singular-like stress-state. An automated refinement allowing the accurate resolution of this feature would demand for an error-based adaptivity scheme. The formulation of such an automated approach is subject to current research.

11.2.3 Combination of multi-level *hp*-adaptivity and the finite cell method

This final example demonstrates the potential of the chosen approach in the context of complex geometric domains. To this end, the perforated double cantilever beam introduced in



$E = 100$ $G_c = 2.5 \cdot 10^{-3}$ $l = 7.5$ $h = 1$
$\nu = 0.3$ $t_{\text{ult}} = 1$ $b = 0.375$ $d = 0.2$

Figure 11.10: Setup of the perforated double cantilever beam benchmark following [Verhoosel et al., 2009] (taken from [Zander et al., 2017]).

[Verhoosel et al., 2009] is considered. The geometry and material properties are depicted in Figure 11.10. As in Example 11.2.1, the Xu-Needleman traction-separation law is used and the structure is loaded by a quadratically distributed shear force applied to the front faces of the two beams.

Due to the perforation, the delamination of the considered structure yields a complex load-displacement curve characterized by *physically-correct* snap-back effects. Thus, the simulation demands for a sophisticated path-following technique. As shown in [Verhoosel et al., 2009], a dissipation-based criterion is well suited for this need and will thus be used here as well.

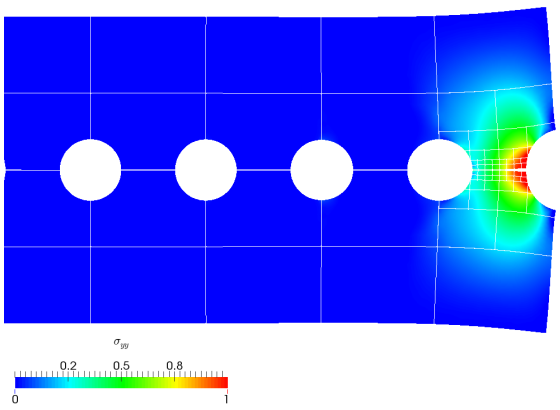
In the original work of Verhoosel et al. [2009], the geometry was discretized using triangular elements. The resolution of the fine-scale geometrical features demands for a small element size, which leads to a high number of unknowns.

An alternative circumventing this problem is the use Finite Cell Method discussed in Chapter 10. To apply this discretization concept to the present case, the weak form introduced in Section 11.1 is reformulated following the finite cell idea

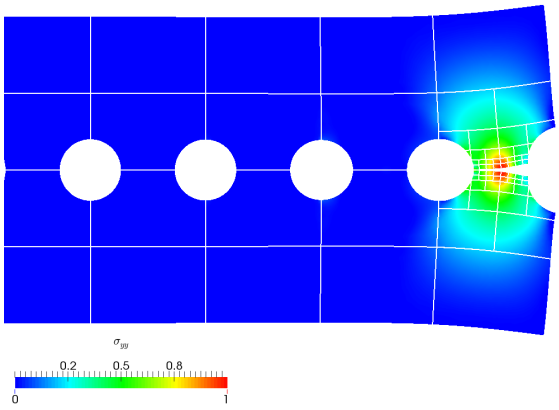
$$\text{Find } \boldsymbol{u} \in \mathcal{V}, \text{ such that}$$
$$\int_{\Omega_\cup} \delta\boldsymbol{\varepsilon} : \alpha\boldsymbol{\sigma}\,\mathrm{d}\Omega + \int_{\Gamma_C} \delta[\![\boldsymbol{u}]\!] \cdot \alpha\boldsymbol{t}([\![\boldsymbol{u}]\!])\,\mathrm{d}\Gamma = \int_{\Omega_\cup} \delta\boldsymbol{u} \cdot \alpha\hat{\boldsymbol{b}}\,\mathrm{d}\Omega + \int_{\Gamma_N} \delta\boldsymbol{u} \cdot \hat{\boldsymbol{t}}\,\mathrm{d}\Gamma \quad \forall\, \delta\boldsymbol{u} \in \mathcal{V}_0, \tag{11.13}$$

with $\mathcal{V}$ and $\mathcal{V}_0$ again denoting the function spaces defined in (11.1.2). Note that also the left hand side integral over Γ_C is augmented by the indicator function α. This allows for a traction-free delamination within the fictitious domain.

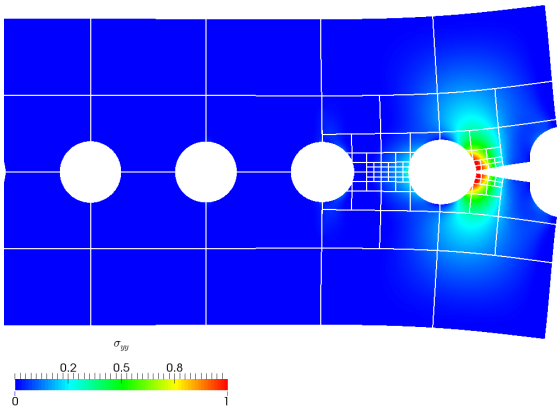
With these changes in the problem, the two beams depicted in Figure 11.10 are discretized using 20×2 hexahedral elements, in which the perforation is treated as fictitious domain. The geometry of the perforation is resolved on the integration level using a quadtree-based sub-cell integration scheme. This simple discretization allows the number of unknowns to be reduced significantly compared to a boundary-conforming finite element approach. However, the coarse base mesh is not sufficient to capture the complex stress patterns emerging around the holes. Following the previous examples, the discretization is, therefore, refined locally around the moving crack tip using four levels of multi-level *hp*-refinement. The gap range in which elements are selected for refinement is set to $g_1 = 2.5 \cdot 10^{-6}$ and $g_2 = 2.5 \cdot 10^{-3}$.



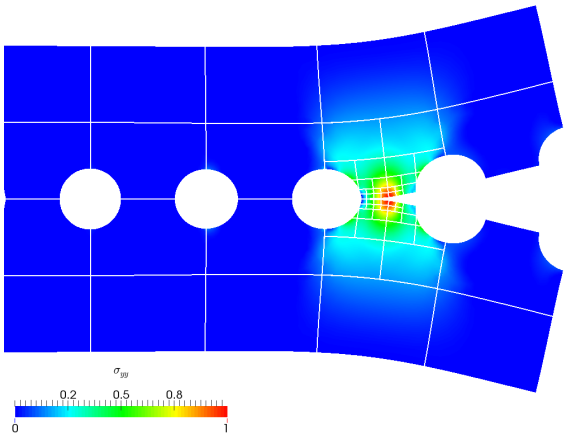
(a) Load step 22, 7104 degrees of freedom



(b) Load step 32, 7104 degrees of freedom



(c) Load step 50, 8244 degrees of freedom



(d) Load step 82, 6764 degrees of freedom

Figure 11.11: Stress distribution on the deformed structure at different load steps, 20×2 elements, $p = 5$, 4 levels of multi-level *hp*-refinement, max. 8564 degrees of freedom throughout the analysis [Zander et al., 2017].

Figure 11.11 depicts the stress concentration moving ahead of the crack-opening, which is computed using the dynamically changing multi-level *hp*-mesh. In Figure 11.12, the load-displacement curve of the problem is depicted, which is characterized by a number of complex snap-back effects. It is worth to notice that the fictitious domain extended problem was solved stably and reliably using the dissipation-based arc-length method presented in [Verhoosel et al., 2009]. A comparison of the multi-level *hp*-derived load-displacement curve with the results obtained using a uniformly refined mesh shows excellent agreement at one order of magnitude less degrees of freedom. The quality of the results presented for this problem clearly indicate the potential of the multi-level *hp*-refinement in combination with the finite cell method for propagation problems with complex domain geometry.

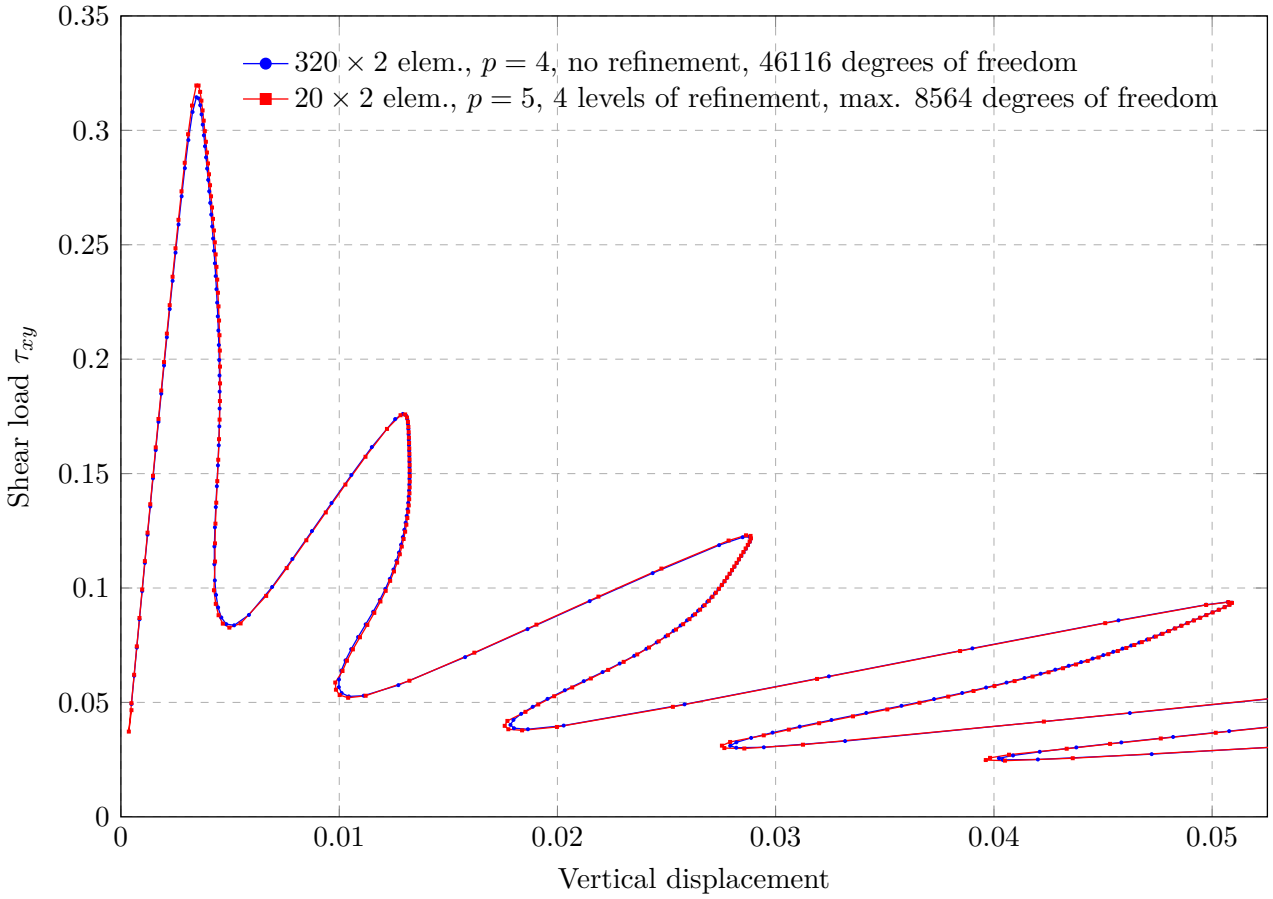


Figure 11.12: Global load-displacement curve of beam tip [Zander et al., 2017].

Chapter 12

Towards fluid mechanics: application to convection diffusion problems

The aim of this chapter is to extend the application range of the multi-level *hp*-method in the direction of fluid mechanics. An essential challenge for any discretization scheme in this application field is the possibility of convection-dominated flows. For this reason, the present chapter focuses on this aspect and assesses the approximation quality of the suggested refinement scheme for this kind of problems. To this end, the first section of this chapter briefly reviews the mathematical and numerical modeling of this application field. On this basis, the second part of the chapter analyzes the approximation accuracy of the multi-level *hp*-refinement scheme by means of one- and two-dimensional benchmark examples.

12.1 Numerical modeling of convection diffusion problems

As discussed in Section 2.1, a cornerstone of any model in the field of continuum mechanics is the conservation or balance of certain physical quantities. For the sake of simplicity, the analysis in this chapter focuses on the conservation of a scalar quantity denoted by Φ, which represents e.g. the mass or the energy contained in a control volume Ω. Further, it is assumed that Φ is given by the integral of a density or concentration u, i.e.

$$\Phi = \int_{\Omega} u \, \mathrm{d}\Omega. \tag{12.1}$$

Following equation (2.1), the conservation equation is expressed mathematically as

$$\frac{d\Phi}{dt} = \frac{d}{dt} \int_{\Omega} u \, \mathrm{d}\Omega = \int_{\Gamma} f_n(u) \, \mathrm{d}\Gamma + \int_{\Omega} \hat{s} \, \mathrm{d}\Omega, \tag{12.2}$$

with f_n denoting the flux over the domain boundary, and $\hat{s}$ denoting the source term internal to the domain.

In Section 2.1, the above equation was simplified based on the assumption of a stationary regime. If this assumption is not valid, it has to be considered that the value of u itself can

change with time, and that u might be *convected* through the control volume Ω by an external flow field $\boldsymbol{a}$. Following the Reynolds transport theorem *cf.* [Holzapfel, 2000, Section 4.2], the left-hand side integral of (12.2) can thus be written as

$$\frac{d}{dt}\int_{\Omega} u \,\mathrm{d}\Omega = \int_{\Gamma} u\,\boldsymbol{a}\cdot\boldsymbol{n}\,\mathrm{d}\Gamma + \int_{\Omega}\frac{\partial u}{\partial t}\,\mathrm{d}\Omega. \tag{12.3}$$

Here, the first integral on the right-hand side models the transport of u over the domain boundary Γ due to the external flow field $\boldsymbol{a}$. The second integral models the change of u within the control volume Ω. Applying the Gauss' divergence theorem, the change with time can be expressed as

$$\frac{d}{dt}\int_{\Omega} u \,\mathrm{d}\Omega = \int_{\Omega}\frac{\partial u}{\partial t} + \nabla\cdot(u\,\boldsymbol{a})\;\mathrm{d}\Omega. \tag{12.4}$$

Using the results presented in Section 2.1, the balance equation (12.2) can thus be given in its integral form as

$$\int_{\Omega}\frac{\partial u}{\partial t} + \nabla\cdot(u\,\boldsymbol{a})\;\mathrm{d}\Omega = \int_{\Omega}\nabla\cdot(\boldsymbol{\kappa}\cdot\nabla u) + \hat{s}\,\mathrm{d}\Omega, \tag{12.5}$$

with $\boldsymbol{\kappa}$ denoting the diffusivity of material under consideration.

Since the control volume Ω is arbitrary, the above equation has to be fulfilled point-wise yielding

$$\frac{\partial u}{\partial t} + \nabla\cdot(u\,\boldsymbol{a}) - \nabla\cdot(\boldsymbol{\kappa}\cdot\nabla u) = \hat{s} \quad \forall\,\boldsymbol{x}\in\Omega \tag{12.6a}$$

$$\Leftrightarrow \frac{\partial u}{\partial t} + u\nabla\cdot\boldsymbol{a} + \nabla u\cdot\boldsymbol{a} - \nabla\cdot(\boldsymbol{\kappa}\cdot\nabla u) = \hat{s} \quad \forall\,\boldsymbol{x}\in\Omega. \tag{12.6b}$$

Under the assumption of a divergence free external flow field $\boldsymbol{a}$, the above equation simplifies to

$$\frac{\partial u}{\partial t} + \nabla u\cdot\boldsymbol{a} - \nabla\cdot(\boldsymbol{\kappa}\cdot\nabla u) = \hat{s} \quad \forall\,\boldsymbol{x}\in\Omega, \tag{12.7}$$

which is supplemented by the following boundary conditions

$$u = \hat{u} \quad \forall\,\boldsymbol{x}\in\Gamma_D \tag{12.8a}$$

$$\boldsymbol{\kappa}\cdot\nabla u\cdot\boldsymbol{n} = \hat{q} \quad \forall\,\boldsymbol{x}\in\Gamma_N. \tag{12.8b}$$

In analogy to the previous chapters, $\hat{u}$ denotes the value of u prescribed on the Dirichlet boundary and $\hat{q}$ the applied flux over the Neumann boundary.

For the use of the Finite Element Method, the model problem has to be transferred from its differential into its variational form. To this end, the differential equation (12.7) is multiplied by a function $v\in H_0^1(\Omega)$ and integrated over the domain Ω yielding

$$\int_{\Omega} v\,\frac{\partial u}{\partial t}\,\mathrm{d}\Omega + \int_{\Omega} v\,\nabla u\cdot\boldsymbol{a}\,\mathrm{d}\Omega - \int_{\Omega} v\,\nabla\cdot(\boldsymbol{\kappa}\cdot\nabla u)\;\mathrm{d}\Omega = \int_{\Omega} v\,\hat{s}\,\mathrm{d}\Omega. \tag{12.9}$$

As outlined in Section 2.1, the third integral on the left-hand side can be written as

$$\int_\Omega v\,\nabla\cdot(\boldsymbol{\kappa}\cdot\nabla u)\,\mathrm{d}\Omega = -\int_\Omega \nabla v\cdot\boldsymbol{\kappa}\cdot\nabla u\,\mathrm{d}\Omega + \int_{\Gamma_N} v\hat{q}\,\mathrm{d}\Gamma, \tag{12.10}$$

Therefore, the weak or variational form of the convention-diffusion problem reads

$$\int_\Omega v\,\frac{\partial u}{\partial t}\,\mathrm{d}\Omega + \int_\Omega v\,\nabla u\cdot\boldsymbol{a}\,\mathrm{d}\Omega + \int_\Omega \nabla v\cdot\boldsymbol{\kappa}\cdot\nabla u\,\mathrm{d}\Omega = \int_\Omega v\,\hat{s}\,\mathrm{d}\Omega + \int_{\Gamma_N} v\hat{q}\,\mathrm{d}\Gamma_N. \tag{12.11}$$

To simplify the notation, the above equation is re-written as

$$\left(v,\frac{\partial u}{\partial t}\right)_\Omega + (v,\nabla u\cdot\boldsymbol{a})_\Omega + (\nabla v,\boldsymbol{\kappa}\cdot\nabla u)_\Omega = (v,\hat{s})_\Omega + (v,\hat{q})_{\Gamma_N}, \tag{12.12}$$

with $(\cdot,\cdot)$ denoting the L_2-inner product.

In the following, it is assumed that the change $d\Phi$ is solely due to the applied external flow field $\boldsymbol{a}$, whereas the local time derivative $\frac{\partial u}{\partial t}$ vanishes. With this assumption, the model problem can be written as

$$\text{Find } u\in H^1_{\hat{u}} \text{ such that } a(v,u) = f(v) \quad \forall\, v\in H^1_0(\Omega), \tag{12.13a}$$

$$\text{with} \quad a(v,u) = (v,\nabla u\cdot\boldsymbol{a})_\Omega + (\nabla v,\boldsymbol{\kappa}\cdot\nabla u)_\Omega \tag{12.13b}$$

$$\text{and} \quad f(v) = (v,\hat{s})_\Omega + (v,\hat{q})_{\Gamma_N}. \tag{12.13c}$$

The above equation has some similarities to the problems considered in the previous chapter. In fact, the functional $f(\cdot)$ is identical to the right-hand side of the weak heat equation given in (2.19). Similarly, the diffusive term $(\nabla v,\boldsymbol{\kappa}\cdot\nabla u)_\Omega$ is present in the weak form of the heat equation. The only, but essential, difference is the convective term $(v,\nabla u\cdot\boldsymbol{a})_\Omega$.

As noted in e.g. [Bartels, 2016, Proposition 7.8] or [Quarteroni and Valli, 1994, Section 8.1], the bi-linear form $a(\cdot,\cdot)$ is coercive and continuous. Hence, the Lax-Milgram Lemma 2.1.2 implies that a unique (weak) solution exists. Furthermore, the problem is suited for being discretized by a Galerkin-based method as Cea's Lemma 2.3.1 ensures the convergence of the numerical approximation towards the analytical solution.

However, the additional convective term renders the bi-linear form non-symmetric. Accordingly, $a(\cdot,\cdot)$ does not define an inner product, which implies that the numerical solution cannot be understood as the a-orthogonal projection of the analytical solution on a finite dimensional sub-space $\mathcal{V}_h$. Hence, the best-approximation property of the Galerkin scheme is lost for convection-diffusion problems, *cf.* [Zienkiewicz et al., 2005b, Section 1.1] or [Donéa and Huerta, 2003, Section 1.1]. This renders the numerical solution of problems where convective effects are non-negligible significantly more challenging than in the context of pure diffusion.

This difficulty can also be interpreted physically: a diffusive transport is symmetric in all spatial directions. In contrast, a convective transport is directed along the streamlines of the external flow field. Accordingly, the additional convective term adds a bias to the information transport by shifting the range of physical influence in the up-stream direction.

This bias has to be reflected by the discretization scheme. However, standard finite elements (as well as central finite difference approaches) weight up- and down-stream information equally. Therefore, these "symmetric" schemes are not optimal for convection-diffusion

problems. A detailed analysis of the truncation error of the discretization scheme reveals that neglecting this up-stream bias introduces a *negative* diffusion as a pure numerical artifact, *cf.* [Zienkiewicz et al., 2005b, Section 2.2.3] or [Donéa and Huerta, 2003, Remark 2.6]. Due to this under-diffusivity, the approximation error manifests itself in spurious oscillations, which render the numerical solution unphysical.

The amount of negative diffusion introduced by the discretization depends on the ratio of convective and diffusive transport and the approximation quality of the discretization. When using linear shape functions in a one-dimensional problem, this relation is expressed by the element Péclet number

$$P_e = \frac{a}{\kappa}\frac{h}{2}. \tag{12.14}$$

For large element Péclet numbers, the negative diffusion introduced by the truncation error of the discretization scheme is larger than the physical diffusion, which causes the approximation to be oscillatory *cf.* [Donéa and Huerta, 2003, Remark 2.6]. Therefore, the discretization has to have enough approximation power to resolve the convective nature of the problem. The applicability of the suggested superposition-based *hp*-refinement approach for this purpose is analyzed in the following.

12.2 Numerical examples[1]

For the analysis of the multi-level *hp*-discretization scheme in the context of convection-dominated problems, the following one- and two-dimensional benchmarks are considered.

12.2.1 One-dimensional convection diffusion benchmark

In this first numerical example, the approximation quality of the suggested *hp*-refinement scheme is tested for a one-dimensional convection diffusion problem. To this end, the following benchmark suggested in [Zienkiewicz et al., 2005b, Example 2.1] is considered

$$a \cdot u' - \kappa \cdot u'' = 0 \quad \forall\, x \in (0,1) \quad \text{with} \quad u(0) = 1 \quad \text{and} \quad u(1) = 0, \tag{12.15}$$

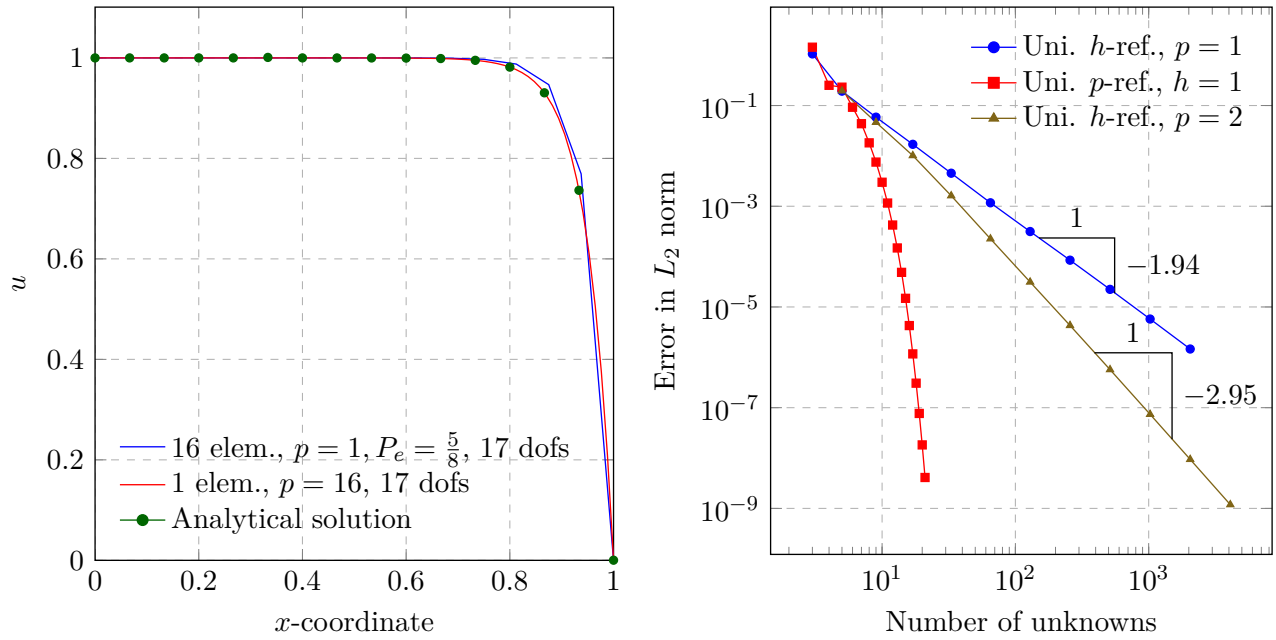
where a and κ are positive constants. The analytical solution to this differential equation reads

$$u = \frac{e^{\frac{a}{\kappa}x} - e^{\frac{a}{\kappa}}}{1 - e^{\frac{a}{\kappa}}}. \tag{12.16}$$

This solution is characterized by a boundary layer on the right boundary, which becomes more pronounced with a decreasing diffusivity κ.

For the numerical approximation of this solution, the governing equation is transformed into its variational form and then discretized using the h- and the p-version of the Finite Element Method, respectively. The resulting approximations for a moderate value of the diffusivity constant $\kappa = 5 \cdot 10^{-2}$ are compared in Figure 12.1a. The results show that using 16 linear

[1] This section is based on the Bachelor's and Master's thesis [Kopp, 2014, 2017] that was supervised by the author of this work.



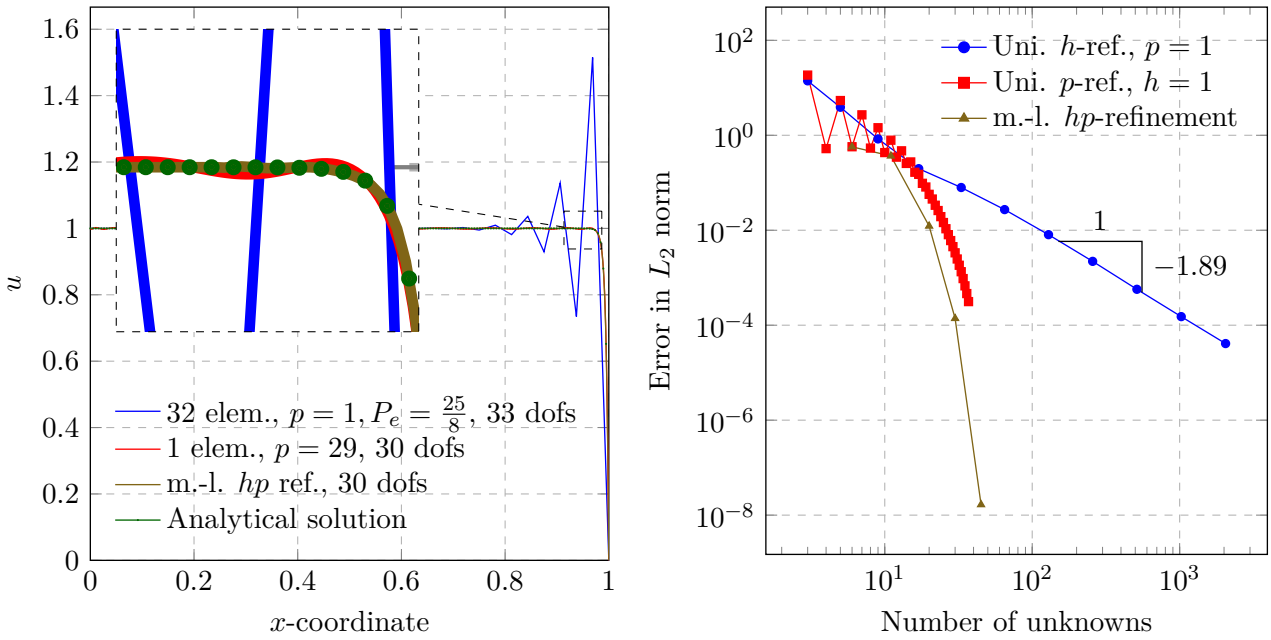
(a) Numerical approximation (b) Convergence

Figure 12.1: One-dimensional convection diffusion problem with $a = 1$ and $\kappa = 5 \cdot 10^{-2}$.

elements is sufficient to yield a numerical approximation that is free of spurious oscillations. This is in agreement with the discussion outlined in the previous section, as the element Péclet number is $P_e = 0.625$ for this case. Accordingly, the discretization is fine enough for the applied convection.

However, the results demonstrate that this does not imply a high quality of the approximation. As depicted in Figure 12.1a, the numerical solution obtained by the 16 linear elements significantly deviates from the analytical solution in the boundary layer. In contrast, the approximation obtained using one element of order 16 is visually not distinguishable from the analytical solution, while using the same number of unknowns as the discretization with linear shape functions. This difference in the result quality is to be expected since the analytical solution is smooth. Accordingly, the p-version yields an exponential decay of the approximation error under a uniform order elevation. In contrast, the error decays only algebraically when uniformly decreasing the element sizes. These *a priori* estimates are confirmed by the convergence results depicted in Figure 12.1b.

Interestingly, the convergence characteristic changes when the diffusivity of the problem is further decreased as shown in Figure 12.2b for $\kappa = 5 \cdot 10^{-3}$. While the convergence of the h-version remains algebraic, the convergence of the p-version is oscillatory in the pre-asymptotic range. A similar odd-even pattern has been reported by Cai [2013]. For higher values of p, the magnitude of the oscillations decays and the convergence eventually turns monotonically exponential. In this regime, the p-version of the Finite Element Method yields a clearly higher approximation quality than the h-version for the same number of unknowns as shown exemplarily in Figure 12.2a. However, when using only one element for the current configuration, this advantage is only significant for polynomial degrees $p > 15$. The applicability of such



(a) Numerical approximation

(b) Convergence

Figure 12.2: One-dimensional convection diffusion problem with $a = 1$ and $\kappa = 5 \cdot 10^{-3}$.

high-order polynomials for higher-dimensional problems of engineering relevance is however limited.

Accordingly, the discretization may benefit from an additional h-refinement that shifts the beginning of the exponential convergence regime to a more applicable polynomial order. To this end, the mesh is refined in a geometric progression towards the right boundary using the suggested multi-level hp-approach. Simultaneously, the approximation order is elevated linearly from left to right with

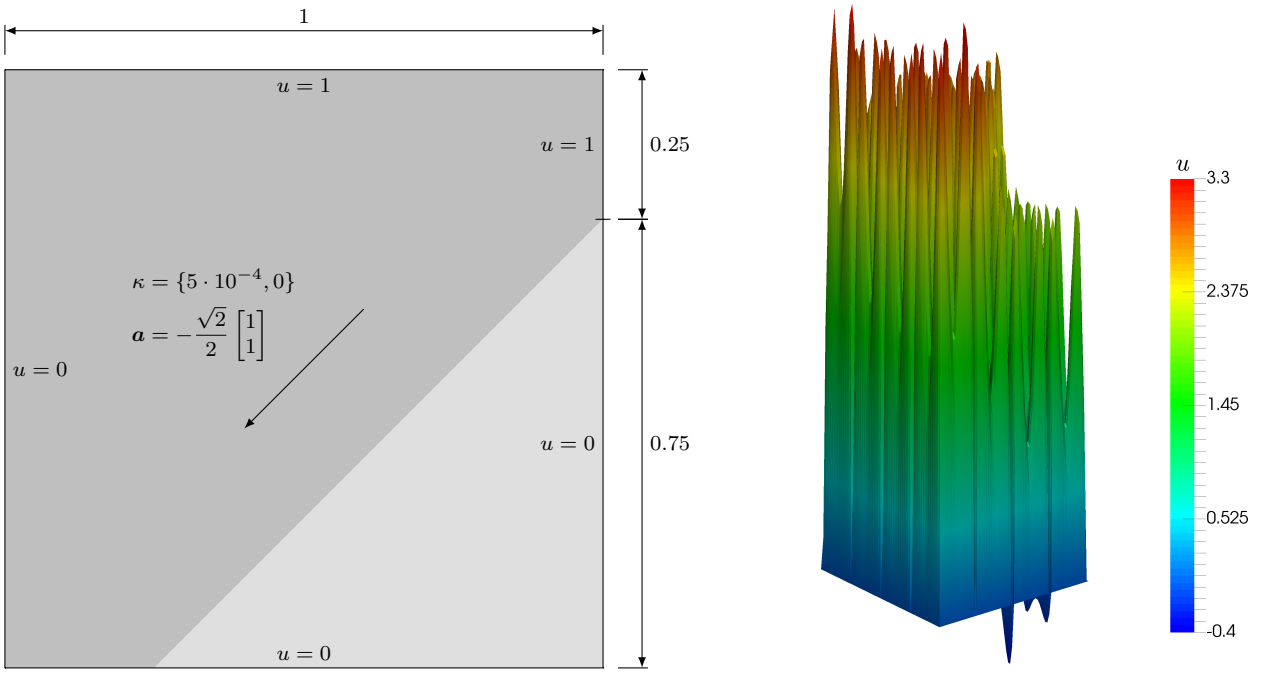
$$p_{\text{left}} = \left\lfloor \frac{m}{2} \right\rfloor + 1 \quad \text{and} \quad p_{\text{right}} = 2m + 1 \tag{12.17}$$

and m denoting the refinement level.

The results depicted in Figure 12.2b demonstrate that this refinement yields an exponential decay of the approximation error that is free of oscillations and even slightly better than the pure p-refinement. Furthermore, Figure 12.2b shows exemplarily that the combination of h- and p-refinement resolves the sharp boundary layer without spurious oscillations.

12.2.2 Convection skew to mesh benchmark

The aim of this second numerical example is to compare the approximation quality of the hp-d- and the multi-level hp-refinement scheme in the context of two-dimensional convection diffusion problems. To this end, the convection skew to mesh benchmark introduced by Brooks and Hughes [1982] is considered. As depicted in Figure 12.3a, the problem is defined on a unit square supplemented by Dirichlet boundary conditions. Within the domain, a convection velocity $\boldsymbol{a}$ is oriented in diagonal direction. This velocity transports the discontinuous boundary



(a) Problem setup

(b) Numerical approximation using 4×4 elements of order $p = 4$, $\kappa = 5 \cdot 10^{-4}$, no stabilization

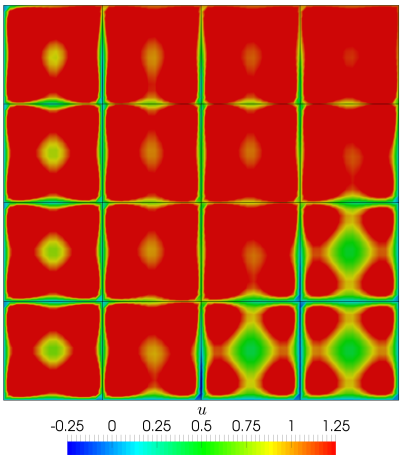
Figure 12.3: Convection-diffusion skew to the mesh benchmark following [Brooks and Hughes, 1982].

value prescribed on the right boundary along the diagonal of the domain. Hence, the analytical solution is characterized by an internal layer along the diagonal and boundary layers at the left and bottom boundaries. The sharpness of these layers depends on the diffusivity κ. In the following, two different cases are considered.

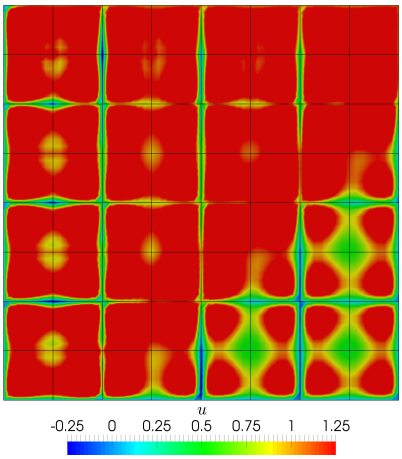
In both cases, the domain is first discretized with 4×4 elements of order six. The boundary conditions on the left and bottom edges are applied in the strong sense. The boundary conditions on the top and right edges are applied in the weak sense using the penalty method [Babuska, 1973]. To this end, the discontinuities are regularized using a spline of order p over the width of the finest elements adjacent to the jump location. In this way, the boundary values are free of oscillations and converge against the exact jump under h- and p-refinement.

12.2.2.1 Convection-dominated transport, no stabilization

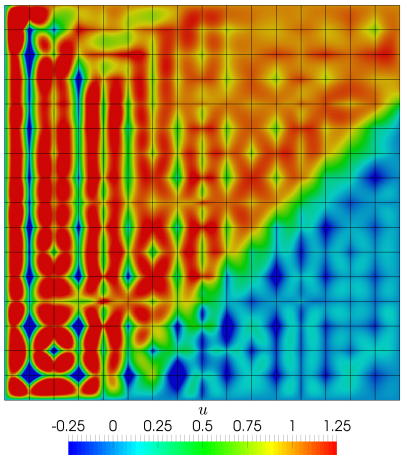
In the first case, a diffusivity of $\kappa = 5 \cdot 10^{-4}$ is assumed. In this configuration, the internal layer widens such that the resolution of the boundary layers is the major numerical challenges. The numerical approximation obtained by the base discretization of 4×4 elements is depicted in Figure 12.3b. The result demonstrates that the numerical solution suffers from severe spurious oscillations, which render the approximation nonphysical. For this reason, the mesh has to be refined in the vicinity of the internal and boundary layers. To this end, a gradient-based refinement indicator is formulated, which samples the numerical solution at an appropriated



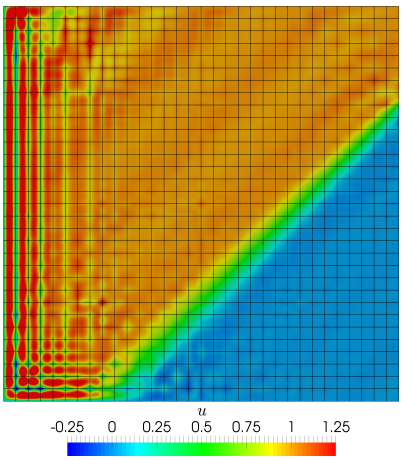
(a) $k = 0$, 289 dofs, 16 integration domains



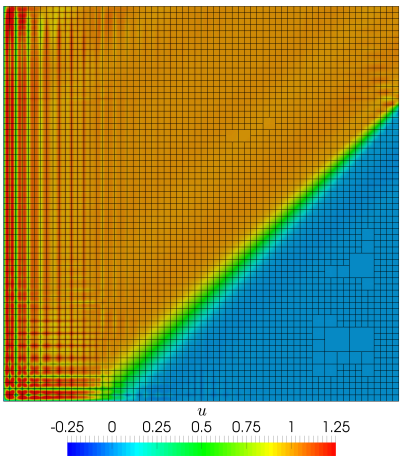
(b) $k = 1$, 345 dofs, 64 integration domains



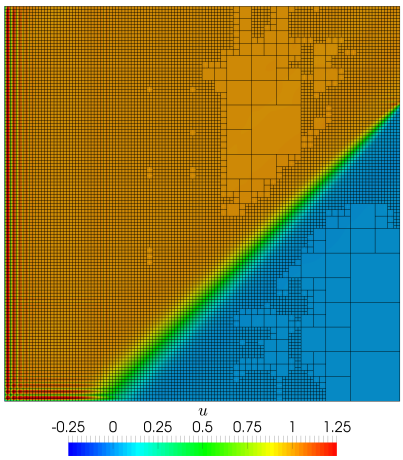
(c) $k = 2$, 553 dofs, 256 integration domains



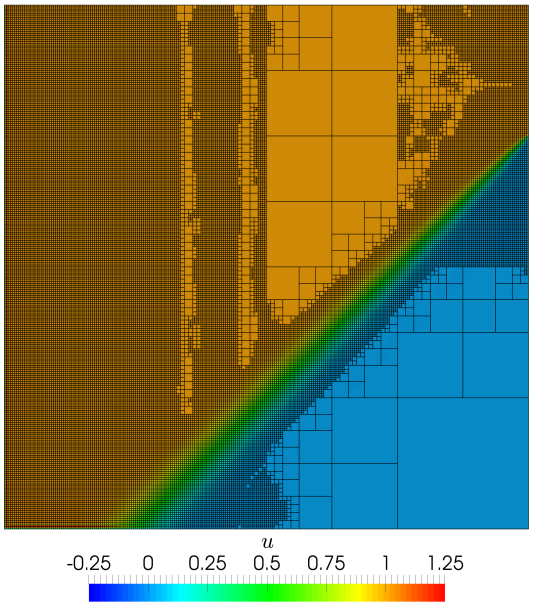
(d) $k = 3$, 1,353 dofs, 1,024 integration domains



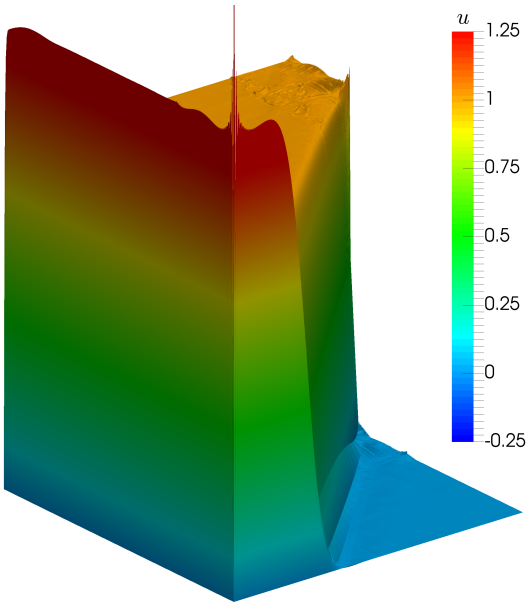
(e) $k = 4$, 4,354 dofs, 3,991 integration domains



(f) $k = 5$, 12,891 dofs, 12,679 integration domains



(g) $k = 6$, 40,624 dofs, 40,822 integration domains



(h) $k = 6$, warped mesh

Figure 12.4: Approximation using *hp-d*-refinement ($||\boldsymbol{a}|| = 1$, $\kappa = 5 \cdot 10^{-4}$, $p = 4$, no stabilization). Used abbreviations: k: number of refinements, dofs: degrees of freedom

number of seed points within each element. If the maximum of $|\nabla u_h|$ observed at the sampled seed points exceeds a predefined threshold, the respective element is marked for refinement. Otherwise, the element is marked for coarsening. For the following studies, the value of this threshold is set empirically to unity.

The numerical approximations obtained when using this refinement strategy in combination with the *hp-d*-refinement scheme are depicted for six refinement steps in Figure 12.4. The results demonstrate that the suggested feedback scheme automatically detects the internal and boundary layer and successfully steers the refinement accordingly. Moreover, the *hp-d*-refinement effectively reduces the amplitude of the spurious numerical oscillation such that the internal layer is solved with high accuracy as shown in Figure 12.4h. However, even six levels of *hp-d*-refinement are not sufficient to sharply resolve the boundary layers at the outflow without numerical oscillations.

The cause of this problem can be deduced from Figure 12.5 depicting the numerical approximation along the x-axis at $y = 0.5$. Here, the dashed vertical lines mark the boundaries of the four base elements along the path. The results show that the numerical approximation obtained using six-levels of *hp-d*-refinement is oscillatory along the *full* length of the left-most base-element, which incorporates the boundary layer. As for the examples discussed in the pre-

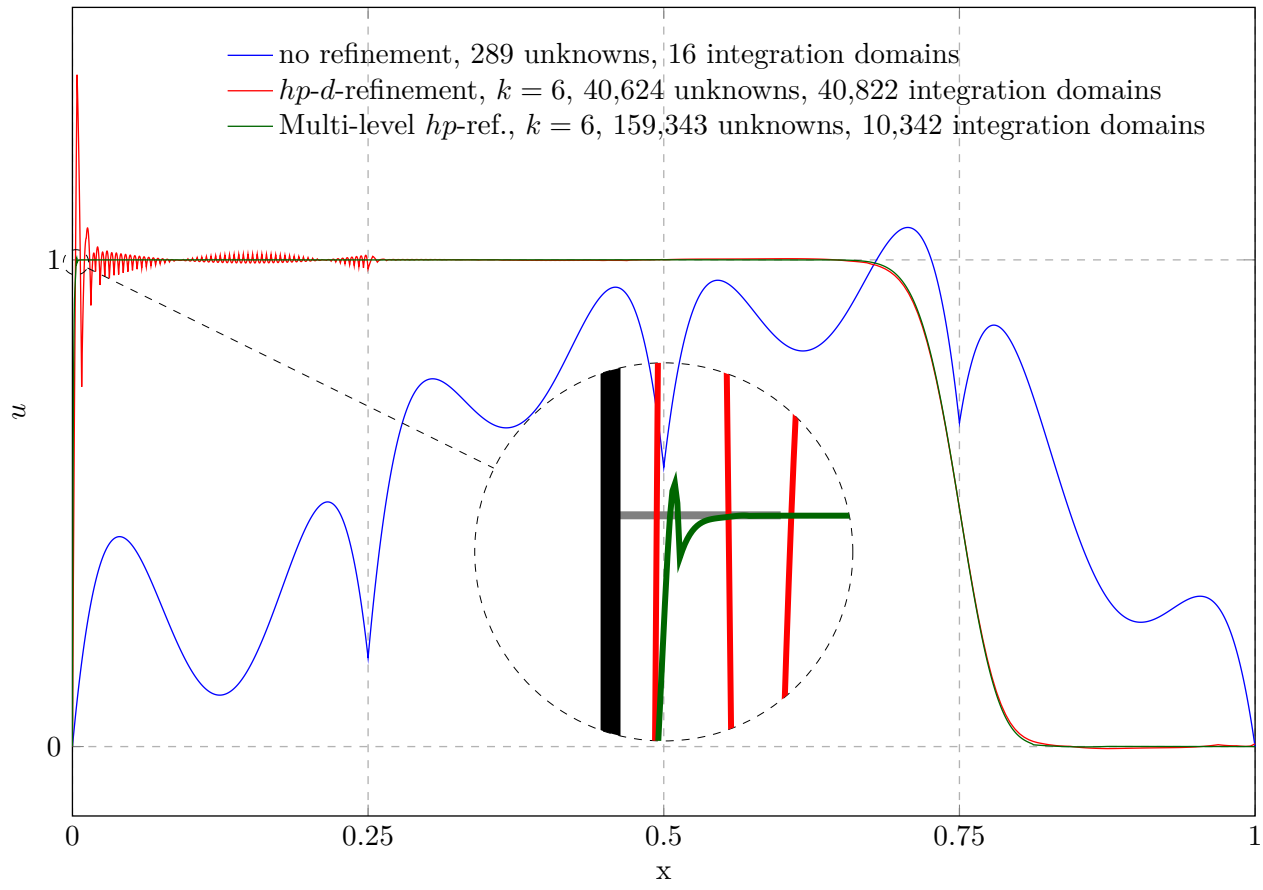
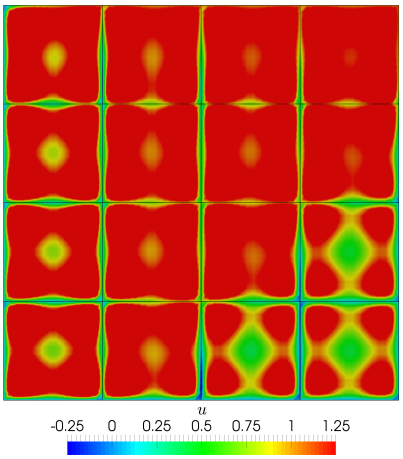
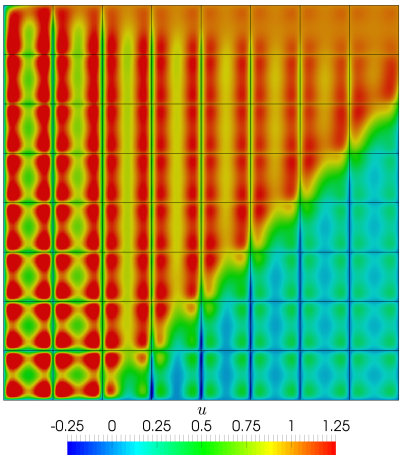


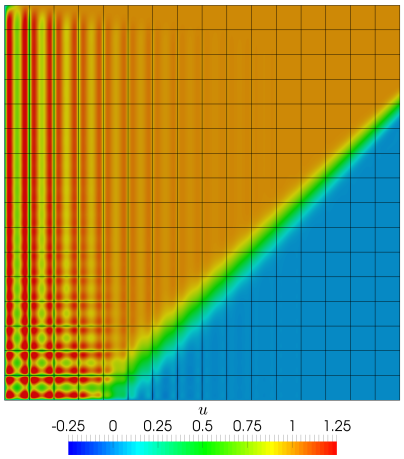
Figure 12.5: Solution evaluated over line at $y = 0.5$ ($||\boldsymbol{a}|| = 1$, $\kappa = 5 \cdot 10^{-4}$, $p = 4$, no stabilization). Used abbreviations: k: number of refinements.



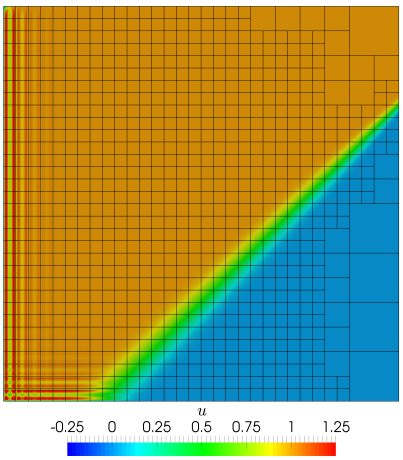
(a) $k = 0$, 289 dofs, 16 integration domains



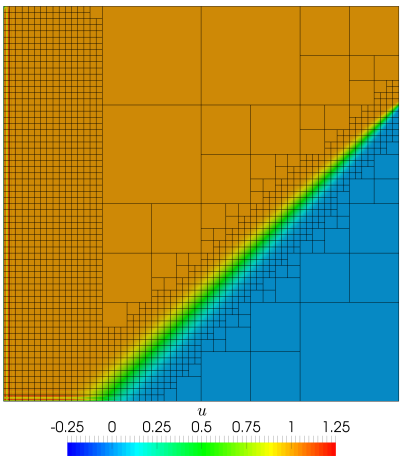
(b) $k = 1$, 1,089 dofs, 64 integration domains



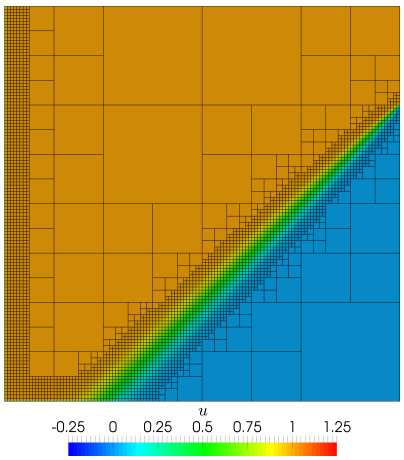
(c) $k = 2$, 4,225 dofs, 256 integration domains



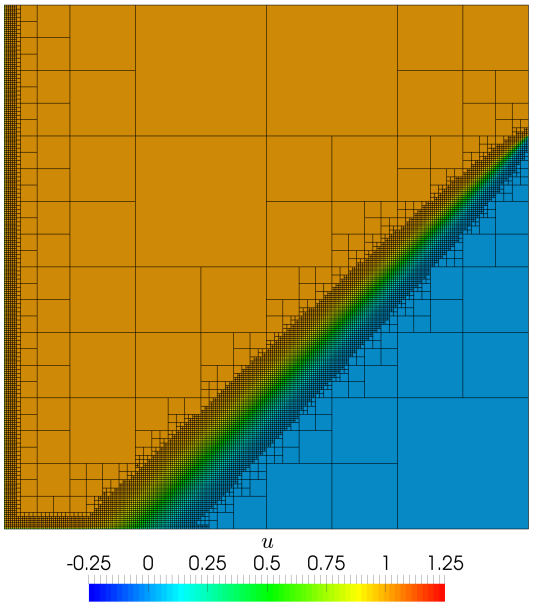
(d) $k = 3$, 14,248 dofs, 895 integration domains



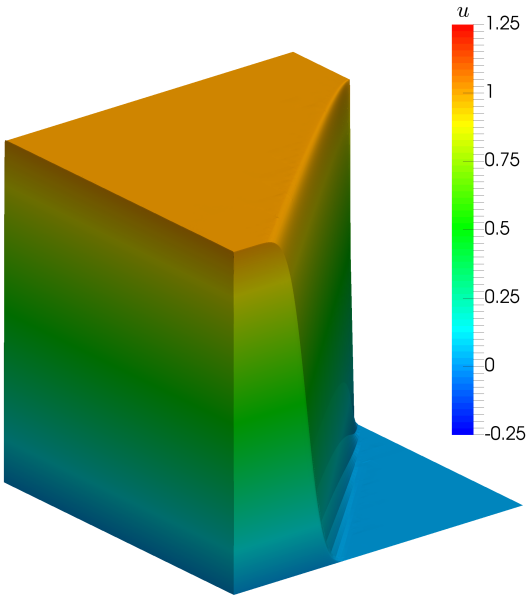
(e) $k = 4$, 26,868 dofs, 1,741 integration domains



(f) $k = 5$, 54,188 dofs, 3,550 integration domains



(g) $k = 6$, 159,243 dofs, 10,342 integration domains

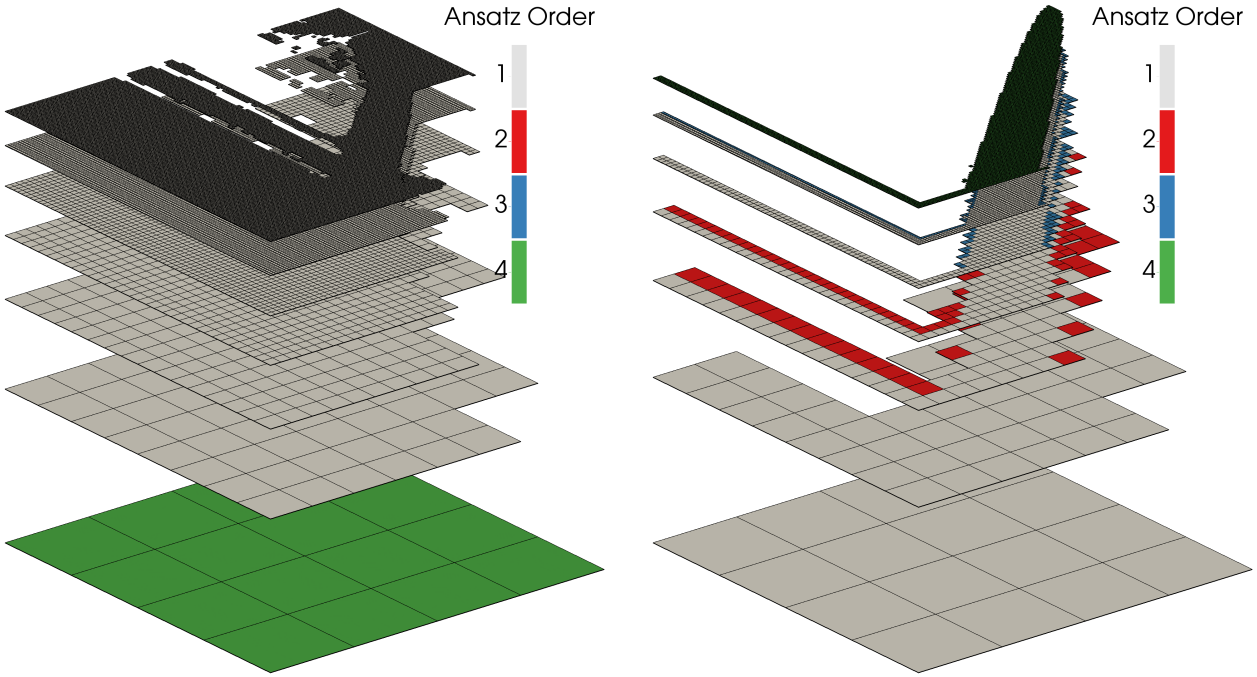


(h) $k = 6$, warped mesh

Figure 12.6: Approximation using multi-level *hp*-refinement ($||\boldsymbol{a}|| = 1$, $\kappa = 5 \cdot 10^{-4}$, $p = 4$, no stabilization). Used abbreviations: k: number of refinements, dofs: degrees of freedom

vious chapters, this characteristic of the *hp-d*-approximation is to be explained by the support of the high-order shape functions: since the *hp-d*-refinement scheme uses only elements with linear shape functions for refinement, the support of the high-order shape functions remains on the coarse base elements. Accordingly, the high-order shape functions have to account for *two* solution characteristics: the thin boundary layer and the virtually constant solution within the domain. Due to the large support of the high-order shape functions, the *hp-d*-refinement cannot decouple these two influences, which yields oscillations that spread over the full length of the base element.

As discussed in the previous chapters, the situation is fundamentally different when refining the base discretization with the multi-level *hp*-strategy. Since this approach reduces the support of the high-order shape functions to the leaf-elements, the solution characteristic of the boundary layer can be effectively separated from the virtually constant solution within the domain. In this way, the spread of the oscillations is limited to the finest element level at the boundary as depicted in Figures 12.5. Moreover, this decoupling of the two separate influences allows the feedback-scheme to coarsen the elements away from the boundary and internal layers as illustrated in Figures 12.6e to 12.6g. Furthermore, the virtually constant parts of the solution can be approximated with low order basis functions. For this reason,



(a) *hp-d*-mesh structure. High order modes have support on the coarse base mesh, while overlay mesh is linear. 40,624 unknowns, 40,822 integration domains.

(b) Multi-level *hp*-mesh structure. Approximation order is elevated linearly towards the characteristic layers, while low order shape functions approximate the solution where it is virtually constant. 159,243 unknowns, 10,342 integration domains.

Figure 12.7: Hierarchical mesh structures obtained by feedback-scheme after six refinement steps. ($||\boldsymbol{a}|| = 1$, $\kappa = 5 \cdot 10^{-4}$, $p = 4$, no stabilization).

the polynomial order of the shape functions can be linearly decreased towards the base of the refinement tree as illustrated in Figure 12.7b, which reduces the total number of unknowns.

Nevertheless, it has to be noted that the multi-level *hp*-discretization yields approximately four times more degrees of freedom than the *hp-d*-mesh when using the six levels of refinement. This directly effects the time and memory required for solving the system of linear equations. However, it has to be also noted that the number of integration domains—and with it the number of integration points *cf.* Figure 6.4—is four times higher when using the *hp-d*-approach. As noted at the end of Section 5.2, the number of element shape functions that have to be evaluated at each integration point is the same for both refinement strategies. Accordingly, the cost of integrating and assembling the element stiffness matrices and post-processing the numerical solution—the two pre-dominant factors in the case of such small number of unknowns—is significantly higher in the case of the *hp-d*-refinement.

12.2.2.2 Pure convective transport, SUPG-stabilization

The previous example can be rendered even more challenging by decreasing the diffusivity κ. These cases demand for an even finer resolution of the boundary and internal layers to prevent spurious oscillations. An alternative approach is to *stabilize* the numerical solution to prevent oscillations when using a coarser discretization. In the past decades, different stabilization strategies have been introduced, and a comprehensive review is given in e.g. [Donéa and Huerta, 2003]. One predominant approach is the Streamline Upwind Petrov-Galerkin (SUPG) scheme, introduced in [Brooks and Hughes, 1982]. The essential idea of this scheme is to account for the convection bias of the problem by increasing the weight of the trial functions in the up-stream direction. To this end, the variational problem (12.13) is augmented in as follows

$$(v, \nabla u \cdot \boldsymbol{a})_\Omega + (\nabla v, \boldsymbol{\kappa} \cdot \nabla u)_\Omega + \sum_{K \in \mathcal{T}_h} (\nabla v \cdot \boldsymbol{a}, \tau \mathcal{R})_K = (v, \hat{s})_\Omega + (v, \hat{q})_{\Gamma_N}\,, \tag{12.18}$$

where τ is the stabilization parameter, and $\mathcal{R}$ denotes the point wise residual

$$\mathcal{R} = \nabla u \cdot \boldsymbol{a} - \nabla \cdot (\boldsymbol{\kappa} \cdot \nabla u) - \hat{s}. \tag{12.19}$$

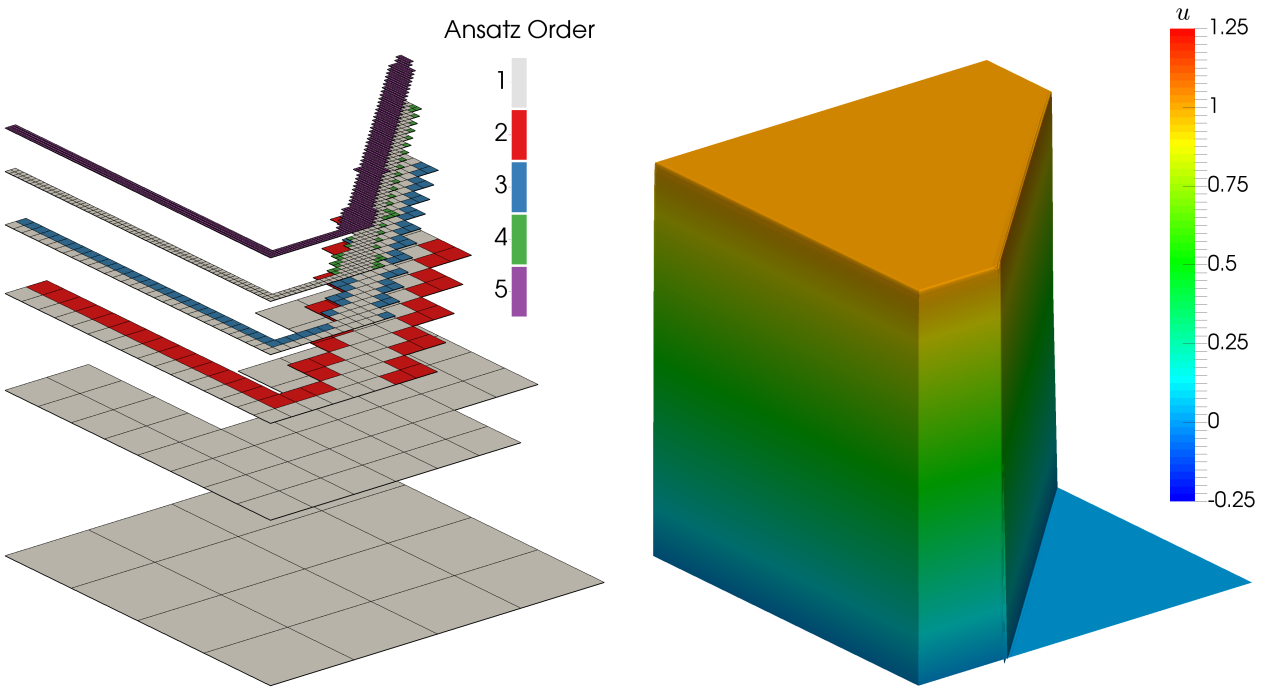
A more detailed description of this approach—as well as other stabilization schemes—can be found in the aforementioned works or e.g. [Zienkiewicz et al., 2005b, Section 2.3.2].

To analyze the combination of the suggested multi-level *hp*-refinement approach with the SUPG-stabilization scheme, the limit-case of $\kappa = 0$ is considered. This pure-transport configuration is numerically challenging as the width of the boundary layers decreases to zero and the internal layer remains sharp along the diagonal. Accordingly, the stabilization is essential in this setting. To account for the approximation quality of the high-order shape functions, the stabilization parameter τ is chosen as

$$\tau = \frac{\tau_h}{\sqrt{p}}, \tag{12.20}$$

with p denoting the approximation order and τ_h the asymptotic limit of the standard SUPG-parameter for h-FEM suggested in e.g. [Donéa and Huerta, 2003, Section 2.4.3], which reads

$$\tau_h = \frac{\sqrt{2}h}{2||\boldsymbol{a}||}. \tag{12.21}$$



(a) Refined discretization, 41,489 unknowns, 4,390 integration domains

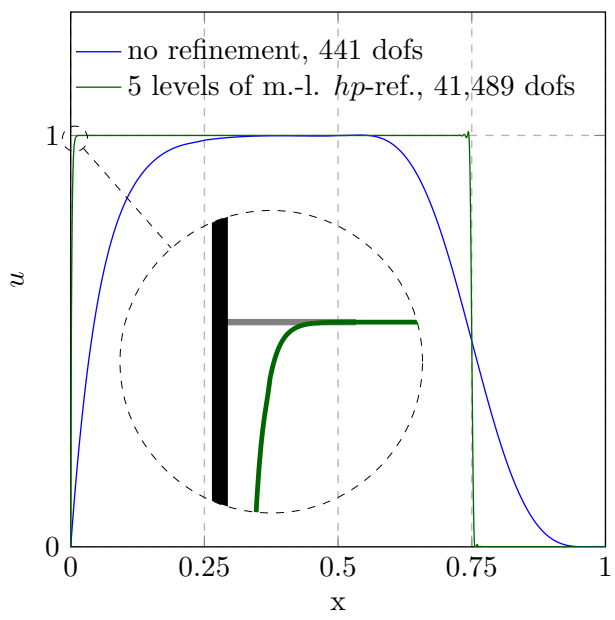
(b) Numerical approximation on refined discretization

Figure 12.8: Approximation using five levels of multi-level *hp*-refinement ($||\boldsymbol{a}|| = 1$, $\kappa = 0$, $p = 5$, $k = 5$, SUPG stabilization).

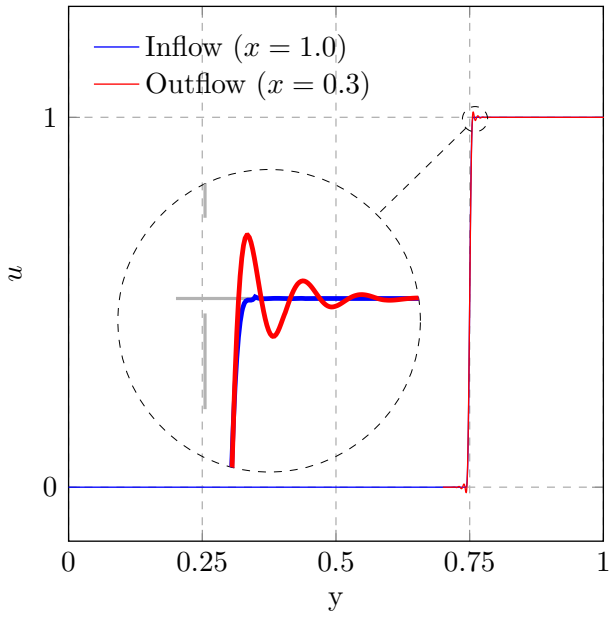
Here, h denotes the length the element edges, and the factor $\sqrt{2}$ accounts for the diagonal orientation of $\boldsymbol{a}$.

As in the previous study, a high-order base mesh consisting of 4×4 elements is used as an initial discretization, which is then refined using the suggested gradient-based indicator. Due to the mediocre quality of the *hp-d*-approximation in the previous example, only the multi-level *hp*-kind of overlay-refinement is considered in the following.

The numerical approximation obtained after five refinement steps is depicted in Figure 12.8. The results demonstrate that—also in the context of a SUPG-stabilized approximation—the suggested gradient-based indicator effectively steers the refinement towards the internal and boundary layers, whereas all other elements remain coarse. In this way, the discontinuity at the outflow can be approximated by a thin boundary layer that is free of spurious oscillations as shown in Figure 12.9a. Moreover, the comparison of the inflow and outflow profile presented in Figure 12.9b demonstrates that the discontinuous boundary condition is transported along the diagonal of the domain without any significant widening, which would indicate an artificial, unphysical cross-wind diffusion. Furthermore, the plot over line shows that spurious oscillations of only minor amplitude arise, whose spreading is limited to the finest element level. This indicates that the refinement by high-order overlay meshes effectively decouples the solution in the vicinity of the internal layer from the remaining parts of the domain.



(a) Plot over x at $y = 0.5$



(b) Comparison of inflow and outflow profile. Outflow profile is shifted by 0.7 in upstream direction.

Figure 12.9: Solution evaluated over line ($||\boldsymbol{a}|| = 1$, $\kappa = 0$, $p = 5$, $k = 5$, SUPG stabilization).

Chapter 13

Application to contact mechanics

The aim of this final chapter is to extend the application range of the suggested multi-level *hp*-refinement scheme to the domain of computational contact mechanics. Besides the highly non-linear nature of this problem and the inequality character of the contact conditions, the numerical simulation of contact faces the additional challenge that a contact between two elastic objects can result in highly localized stress characteristics. This commonly implies a non-smooth transition of the surface traction at the boundary of the regions in contact.

In analogy to the application of changing boundary conditions discussed in Section 8.3, this results in a reduced regularity of the analytical solution *cf.* [Schumann, 1989]. As presented in e.g. [Franke, 2011], resolving the boundary of the contact zone is therefore not sufficient to obtain an exponential convergence of the numerical approximation. In [Franke, 2011] it is instead demonstrated that the irregularity of the solution has to be confined by an *hp*-mesh that is geometrically graded towards the boundaries of the contact regions.

The essential challenge in this context is that the regions of contact might change their position during the course of the non-linear simulation. When using a *static* discretization scheme, the initial mesh has to be fine enough along the *complete* potential contact boundary. For large contact interfaces, this *a priori* refinement approach results in high computational costs.

When instead using a *dynamic* discretization approach, the refinement can stay local to the contact points by following the evolving boundary of the contact zone. In this way, the length of the *potential* contact interface is *decoupled* from the *actual* contact zone.

The aim of this chapter is to assess whether the suggested multi-level *hp*-scheme is able to provide these advantages. To this end, the chapter begins with a brief review of computational contact mechanics. In Section 13.2, algorithmic aspects of the dynamic *hp*-discretization in the context of contact mechanics are discussed. The chapter closes by assessing the approximation accuracy of the suggested *hp*-approach in Section 13.3. To this end, the well-known Hertz benchmark and a geometrically non-linear test case involving moving contact are considered. For the sake of brevity, the examples focus on two-dimensional, normal contact against a rigid plane. The extension to three-dimensional, multi-body contact in combination with the Finite Cell Method is subject to current research and will be published in the near future [Bog et al., 2017].

13.1 Review of computational contact mechanics

In the most generic case, computational contact mechanics aims at simulating the contact between two or more continua that are subjected to large deformations. For this reason, this section begins by extending the mathematical model derived for linear elasticity in Section 2.1 to the regime of non-linear solid mechanics. The description focuses on the aspects that are necessary in the context of contact mechanics of elastic bodies. More detailed descriptions discussing e.g. thermo-mechanical models and other material types can be found in e.g. [Holzapfel, 2000; Wriggers, 2006]

Following closely [Laursen, 2003], the second part of this section outlines the mathematical model for contact mechanics. As mentioned beforehand, the formulation in this work focuses on normal contact, neglecting effects like sliding and friction. To regularize the contact conditions, a conventional penalty approach is chosen. An outline of alternative approaches can be found in e.g. [Laursen, 2003; Wriggers, 2006]

The section closes with a brief discussion of computational aspects including the linearization and the discretization of the non-linear contact problem.

13.1.1 Non-linear solid mechanics

To define the notation later used in this chapter, the aim of this section is to give a brief review of the mathematical model used to describe non-linear solid mechanics. For this purpose, this section proceeds in five steps. First, non-linear kinematics are discussed. Second, the concept of stresses introduced in Chapter 2 is extended to the non-linear regime. In the third step, a brief outline of constitutive modeling is given. In the two final steps, the governing equations are stated in their strong and their weak form, respectively. The outline closely follows [Bonet and Wood, 2008; Holzapfel, 2000; Laursen, 2003].

13.1.1.1 Non-linear kinematics

The essential assumption of a kinematically linear regime—considered in the previous chapters—implies that the deformation resulting from the applied load has no significant effect on the control volume Ω. In the case of large deformations, this assumption is no longer valid. Instead, the control volume has to be considered in two different configurations: in an *initial, undeformed* configuration Ω_0 that is assumed to be stress free, and in a *current*, *spatial*, or *deformed* configuration Ω_c. In the following, the mapping of coordinates, vectors, volumes, areas, and lengths between these two configurations are discussed.

Change of coordinates

The coordinates in the initial configuration—so-called *material points*—are denoted by $\boldsymbol{X} \in \Omega_0$. The *current position* $\boldsymbol{x}$ of each material point $\boldsymbol{X}$ is given by a *mapping* $\boldsymbol{\varphi}$, i.e.

$$\boldsymbol{x} = \boldsymbol{\varphi}\left(\boldsymbol{X}\right). \tag{13.1}$$

The deformed configuration follows as

$$\Omega_c = \boldsymbol{\varphi}\left(\Omega_0\right), \tag{13.2}$$

and similarly, the deformed Dirichlet and Neumann boundaries are given by

$$\gamma_D = \boldsymbol{\varphi}(\Gamma_D) \tag{13.3a}$$
$$\gamma_N = \boldsymbol{\varphi}(\Gamma_N). \tag{13.3b}$$

Since each material point $\boldsymbol{X}$ has one and only one current position $\boldsymbol{x}$, the mapping $\boldsymbol{\varphi}$ is bijective.

Change of vectors

Beside individual points, also vectors can be mapped from the initial to the current configuration. To this end, the *deformation gradient* is defined as

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}. \tag{13.4}$$

Using this tensor, an initial infinitesimal vector $d\boldsymbol{X}$ can be mapped to its counterpart in the current configuration $d\boldsymbol{x}$ as follows

$$d\boldsymbol{x} = \boldsymbol{F} d\boldsymbol{X}. \tag{13.5}$$

This operation is commonly denoted as *push forward*. As $\boldsymbol{\varphi}$ is bijective, the *pull back* of spatial vector $d\boldsymbol{x}$ follows analogously

$$d\boldsymbol{X} = \boldsymbol{F}^{-1} d\boldsymbol{x}. \tag{13.6}$$

Change of volumes

In addition to the vector operation, the deformation gradient can also be used to express the change of volume as

$$dv = J\, dV \quad \text{with } J = \det(\boldsymbol{F}), \tag{13.7}$$

with dV and dv denoting a reference volume in the initial configuration and its deformed counterpart, respectively.

Change of areas

Similarly, the change of area can be computed by considering two elemental vectors $d\boldsymbol{x}_1$ and $d\boldsymbol{x}_2$ in the deformed configuration. The deformed reference area da is given by the cross product

$$d\boldsymbol{x}_1 \times d\boldsymbol{x}_2 = \boldsymbol{n} da, \tag{13.8}$$

with $\boldsymbol{n}$ being the unit normal vector in the deformed configuration. Using (13.5), the size of the deformed area can be expressed in terms of the undeformed area dA by means of Nanson's formula *cf.* [Laursen, 2003, Equation 2.82]

$$\boldsymbol{n} da = J \boldsymbol{F}^{-\top} \boldsymbol{N} dA, \tag{13.9}$$

with $\boldsymbol{N}$ denoting the outward pointing unit normal vector in the initial configuration.

Change of lengths

In analogy to the change of area, the length of a deformed line segment ds can be expressed using the inner product of an elemental vector $d\boldsymbol{x}$ in the following way

$$\begin{aligned} ds^2 &= d\boldsymbol{x} \cdot d\boldsymbol{x} = (d\boldsymbol{x})^\top d\boldsymbol{x} = (\boldsymbol{F} d\boldsymbol{X})^\top \boldsymbol{F} d\boldsymbol{X} \\ &= d\boldsymbol{X}^\top \boldsymbol{F}^\top \boldsymbol{F} d\boldsymbol{X} = d\boldsymbol{X}^\top \boldsymbol{C} d\boldsymbol{X} = d\boldsymbol{X} \cdot \boldsymbol{C} d\boldsymbol{X}, \end{aligned} \tag{13.10}$$

which gives rise to the *right Cauchy-Green* deformation tensor

$$\boldsymbol{C} = \boldsymbol{F}^\top \boldsymbol{F}. \tag{13.11}$$

Similarly, the initial length of the line segment can be computed as

$$\begin{aligned} dS^2 &= d\boldsymbol{X} \cdot d\boldsymbol{X} = (d\boldsymbol{X})^\top d\boldsymbol{X} = (\boldsymbol{F}^{-1} d\boldsymbol{x})^\top \boldsymbol{F}^{-1} d\boldsymbol{x} \\ &= d\boldsymbol{x}^\top \boldsymbol{F}^{-\top} \boldsymbol{F}^{-1} d\boldsymbol{x} = d\boldsymbol{x}^\top \boldsymbol{b}^{-1} d\boldsymbol{x}, \end{aligned} \tag{13.12}$$

giving rise to the *left Cauchy-Green* deformation tensor

$$\boldsymbol{b} = \boldsymbol{F} \boldsymbol{F}^\top. \tag{13.13}$$

Using these results, the change of length can be expressed relative to the initial length as follows

$$\frac{ds^2 - dS^2}{2dS^2} = \frac{d\boldsymbol{x} \cdot d\boldsymbol{x} - d\boldsymbol{X} \cdot d\boldsymbol{X}}{2dS^2} = \frac{d\boldsymbol{X}}{dS} \cdot \frac{1}{2} (\boldsymbol{C} - \boldsymbol{I}) \frac{d\boldsymbol{X}}{dS} = \boldsymbol{N} \cdot \boldsymbol{E} \boldsymbol{N}, \tag{13.14}$$

motivating the definition of the *Green-Lagrange* strain tensor

$$\boldsymbol{E} = \frac{1}{2} (\boldsymbol{C} - \boldsymbol{I}). \tag{13.15}$$

Similarly, the change of length can be expressed relative to deformed length as

$$\frac{ds^2 - dS^2}{2ds^2} = \boldsymbol{n} \cdot \boldsymbol{e} \boldsymbol{n}, \tag{13.16}$$

with $\boldsymbol{e}$ denoting the *Euler-Almansi* strain tensor

$$\boldsymbol{e} = \frac{1}{2} \left(\boldsymbol{I} - \boldsymbol{b}^{-1} \right). \tag{13.17}$$

13.1.1.2 Stresses

In case of a general motion, the deformed continuum is subjected to stresses. These stresses can be expressed by the *Cauchy stress tensor* $\boldsymbol{\sigma}$, whose components represent forces acting in the deformed continuum relative to unit areas in the deformed configuration at a spatial point $\boldsymbol{x}$. The corresponding traction vector representing the *current* force per unit of deformed area is then given by the Cauchy stress theorem

$$\boldsymbol{t} = \boldsymbol{\sigma} \boldsymbol{n}. \tag{13.18}$$

In analogy to the lengths, areas, and volumes discussed before, also the stress state and the traction vector can be transformed from their deformed configuration into the initial counterpart. To this end, the change of the vector itself *and* the change of area have to be taken into account. Considering a differential force in the deformed configuration

$$d\boldsymbol{f} = \boldsymbol{t}da, \tag{13.19}$$

and using Nanson's formula, it follows that

$$d\boldsymbol{f} = \boldsymbol{t}da = \boldsymbol{\sigma}\boldsymbol{n}da = J\boldsymbol{\sigma}\boldsymbol{F}^{-\top}\boldsymbol{N}dA. \tag{13.20}$$

This leads to the definition of the *first Piola Kirchoff stress tensor* measuring the *current* force per *reference* area dA

$$\boldsymbol{P} = J\boldsymbol{\sigma}\boldsymbol{F}^{-\top}. \tag{13.21}$$

The Piola traction follows in analogy as

$$\boldsymbol{T} = \boldsymbol{P}\boldsymbol{N}. \tag{13.22}$$

According to the above formula, $\boldsymbol{P}$ is a two-point tensor in the sense that it maps a normal vector from the initial configuration onto a traction vector in the deformed configuration. This traction can be pulled back to obtain a traction defined in the initial configuration. This gives rise to the *second Piola Kirchoff stress tensor*

$$\boldsymbol{S} = \boldsymbol{F}^{-1}\boldsymbol{P} = J\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-\top}, \tag{13.23}$$

measuring the stress in the initial configuration.

13.1.1.3 Constitutive modeling

To express the stresses in terms of the aforementioned strains, the material properties have to be taken into account by means of an appropriate constitutive model. This work focuses on hyper-elastic materials. The essential assumption used for these kind of materials is that the current stress state is path independent in the sense that it only depends on the initial state and the final configuration *cf.* [Bonet and Wood, 2008, Section 6.2]. This implies the existence of an energy potential W such that

$$\boldsymbol{S} = 2\frac{\partial W}{\partial \boldsymbol{C}} \tag{13.24a}$$

$$\boldsymbol{P} = 2\boldsymbol{F}\frac{\partial W}{\partial \boldsymbol{C}} \tag{13.24b}$$

$$\boldsymbol{\sigma} = \frac{2}{J}\boldsymbol{F}\frac{\partial W}{\partial \boldsymbol{C}}\boldsymbol{F}^{\top}. \tag{13.24c}$$

This allows the definition of a *material elasticity tensor*

$$\mathbb{C} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{E}}, \tag{13.25}$$

which expresses the incremental change of the stress $\boldsymbol{S}$ with respect to the strain $\boldsymbol{E}$.

13.1.1.4 Balance equations in the strong form

The final step of deriving the governing equations consists in connecting the stress state and the external loading. To this end, the concept of momentum balance is applied. Following the derivation given in Section 2.1, the final equilibrium equation reads

$$\begin{aligned} \mathrm{div}\boldsymbol{\sigma} + \hat{\boldsymbol{b}} &= \boldsymbol{0} \quad \forall \boldsymbol{x} \in \Omega_c && (13.26a)\\ \boldsymbol{\varphi} &= \hat{\boldsymbol{\varphi}} \quad \forall \boldsymbol{x} \in \gamma_D && (13.26b)\\ \boldsymbol{\sigma}\boldsymbol{n} &= \hat{\boldsymbol{t}} \quad \forall \boldsymbol{x} \in \gamma_N, && (13.26c) \end{aligned}$$

with $\hat{\boldsymbol{b}}$ denoting the prescribed body force acting per unit of deformed volume, $\hat{\boldsymbol{t}}$ being the prescribed Cauchy traction, and $\hat{\boldsymbol{\varphi}}$ denoting the prescribed final position of the Dirichlet boundary. The div-operator represents the divergence with respect to the deformed coordinates

$$\mathrm{div}(\cdot) = \frac{\partial(\cdot)_{ij}}{\partial x_j}. \tag{13.27}$$

The above formulation is challenging as it is formulated in the deformed configuration, which itself depends on the solution. Using the aforementioned pull back operations, the balance equation can be formulated in the initial configuration, reading

$$\begin{aligned} \mathrm{Div}\boldsymbol{P} + \hat{\boldsymbol{B}} &= \boldsymbol{0} \quad \forall \boldsymbol{X} \in \Omega_0 && (13.28a)\\ \boldsymbol{\varphi} &= \hat{\boldsymbol{\varphi}} \quad \forall \boldsymbol{X} \in \Gamma_D && (13.28b)\\ \boldsymbol{P}\boldsymbol{N} &= \hat{\boldsymbol{T}} \quad \forall \boldsymbol{X} \in \Gamma_N, && (13.28c) \end{aligned}$$

with $\hat{\boldsymbol{T}}$ denoting the prescribed Piola traction, and $\hat{\boldsymbol{B}}$ denoting the prescribed body force per reference volume

$$\hat{\boldsymbol{B}} = J\hat{\boldsymbol{b}}. \tag{13.29}$$

The operation Div represents the divergence with respect to the initial coordinates

$$\mathrm{Div}(\cdot) = \frac{\partial(\cdot)_{iJ}}{\partial X_J}. \tag{13.30}$$

13.1.1.5 Balance equations in the weak form

As discussed in Section 2.1, the discretization by the Finite Element Method is based on the weak form of the balance equations. In analogy to the small strain scenario, this weak form is obtained by multiplying the differential equation with an admissible test function $\boldsymbol{v} \in H_0^1(\Omega_0)$ and integrating the resulting expression over the initial domain Ω_0. After applying Gauss' divergence theorem, the final form reads

$$\text{Find } \boldsymbol{u} \in H^1_{\hat{\boldsymbol{\varphi}}}(\Omega_0) \text{ such that } G_{\mathrm{int}}(\boldsymbol{v}, \boldsymbol{u}) = G_{\mathrm{ext}}(\boldsymbol{v}) \quad \forall\, \boldsymbol{v} \in H_0^1(\Omega_0), \tag{13.31}$$

with

$$G_{\mathrm{int}}(\boldsymbol{v}, \boldsymbol{u}) = \int_{\Omega_0} \mathrm{Grad}\boldsymbol{v} : \boldsymbol{P}\, \mathrm{d}\Omega \quad \text{and} \quad G_{\mathrm{ext}}(\boldsymbol{v}) = \int_{\Omega_0} \boldsymbol{v} \cdot \hat{\boldsymbol{B}}\, \mathrm{d}s\Omega + \int_{\Gamma_N} \boldsymbol{v} \cdot \hat{\boldsymbol{T}}\, \mathrm{d}\Gamma, \tag{13.32}$$

and Grad denoting the gradient with respect to the coordinates in the initial configuration

$$\mathrm{Grad}(\cdot) = \frac{\partial(\cdot)_i}{\partial X_J}. \tag{13.33}$$

It has to be noted that G_{int} is not a bi-linear form as $\boldsymbol{P}$ is non-linear in $\boldsymbol{u}$. Thus, the mathematical concepts discussed in the first part of this work are not directly applicable. For this reason, textbooks commonly assume an engineering viewpoint and identify G_{int} and G_{ext} as the virtual work due to internal stresses and external loads, respectively [Laursen, 2003, Page 80].

13.1.2 Contact mechanics

The previous section outlines the equation of motion for a single body. To extend the mathematical model to the regime of contact mechanics, two separate bodies are considered, whose initial configurations are given by $\Omega_0^{(1)}$ and $\Omega_0^{(2)}$, respectively. Both continua undergo motions $\boldsymbol{\varphi}^{(1)}$ and $\boldsymbol{\varphi}^{(2)}$, such that their deformed counterparts follow as

$$\Omega_c^{(i)} = \boldsymbol{\varphi}^{(i)}(\Omega_0^{(i)}). \tag{13.34}$$

The motions $\boldsymbol{\varphi}^{(i)}$ have to comply with the balance of momentum for each of the two domains

$$\mathrm{Div}\boldsymbol{P}^{(i)} + \hat{\boldsymbol{B}}^{(i)} = \boldsymbol{0} \qquad \forall \boldsymbol{X} \in \Omega^{(i)} \tag{13.35a}$$

$$\boldsymbol{\varphi}^{(i)} = \hat{\boldsymbol{\varphi}}^{(i)} \quad \forall \boldsymbol{X} \in \Gamma_D^{(i)} \tag{13.35b}$$

$$\boldsymbol{P}^{(i)}\boldsymbol{N}^{(i)} = \hat{\boldsymbol{T}}^{(i)} \quad \forall \boldsymbol{X} \in \Gamma_N^{(i)}. \tag{13.35c}$$

Due to the motions, parts of the domain boundaries might come into contact. These possible contact surfaces are denoted as $\Gamma_C^{(i)}$ in the initial configuration and as $\gamma_C^{(i)}$ in the deformed configuration. One of the two surfaces is identified as the *master* or *target* surface. The other interface is chosen as *slave* or *contactor* surface. Without loss of generality, $\Gamma_C^{(1)}$ is used as slave in the following.

This slave side is parametrized to monitor the contact with respect to the master surface. To this end, the *contact point* $\bar{\boldsymbol{Y}}(\boldsymbol{X}) \in \Gamma_C^{(2)}$ is determined for each (slave) point $\boldsymbol{X} \in \Gamma_C^{(1)}$ via a closest point projection

$$\bar{\boldsymbol{Y}}(\boldsymbol{X}) = \arg \min_{\boldsymbol{Y}(\boldsymbol{X}) \in \Gamma_C^{(2)}} ||\varphi^{(1)}(\boldsymbol{X}) - \varphi^{(2)}(\boldsymbol{Y})||. \tag{13.36}$$

The *gap function* $g(\boldsymbol{X})$, measuring the penetration, is then given by the normal component of the distance

$$g(\boldsymbol{X}) = -\boldsymbol{\nu} \cdot \left(\boldsymbol{\varphi}^{(1)}(\boldsymbol{X}) - \boldsymbol{\varphi}^{(2)}(\bar{\boldsymbol{Y}})\right), \tag{13.37}$$

with $\boldsymbol{\nu}$ denoting the unit vector normal to the deformed master surface $\gamma_C^{(2)}$ at $\bar{\boldsymbol{y}} = \boldsymbol{\varphi}^{(2)}(\bar{\boldsymbol{Y}}(\boldsymbol{X}))$.

If the gap is closed, the two bodies can interact by means of a traction. The traction vector is given by

$$\boldsymbol{T}(\boldsymbol{X}) = \boldsymbol{P}(\boldsymbol{X})\boldsymbol{N}(\boldsymbol{X}). \tag{13.38}$$

This contact traction can be decomposed into its normal and tangential parts, reading

$$\boldsymbol{T}(\boldsymbol{X}) = \boldsymbol{t}_T(\boldsymbol{X}) + p_N(\boldsymbol{X})\boldsymbol{\nu}(\bar{\boldsymbol{Y}}(\boldsymbol{X})), \tag{13.39}$$

where $\boldsymbol{t}_T$ is the tangential traction vector, and $p_N(\boldsymbol{X})$ denotes the contact pressure, which assumes positive values in compression. As discussed in the introduction of this chapter, this work focuses on pure normal contact. Accordingly, the traction can be simplified as follows

$$\boldsymbol{T}(\boldsymbol{X}) = p_N(\boldsymbol{X})\boldsymbol{\nu}(\bar{\boldsymbol{Y}}(\boldsymbol{X})), \tag{13.40}$$

and the contact conditions can be written in the Kuhn-Tucker form, *cf.* [Wriggers, 2006]

$$p_N \geq 0 \quad \forall \boldsymbol{X} \in \Gamma_C^{(1)} \tag{13.41a}$$

$$g \leq 0 \quad \forall \boldsymbol{X} \in \Gamma_C^{(1)} \tag{13.41b}$$

$$p_N g = 0 \quad \forall \boldsymbol{X} \in \Gamma_C^{(1)}. \tag{13.41c}$$

In analogy to the single body case discussed in the previous section, the governing equations can also be formulated in the weak form in the following way

$$\begin{aligned} G(\boldsymbol{u},\boldsymbol{v}) = &\overbrace{\sum_{i=1}^{2} \int_{\Omega_0^{(i)}} \mathrm{Grad}\boldsymbol{v}^{(i)} : \boldsymbol{P}^{(i)} \,\mathrm{d}\Omega}^{G^{\mathrm{int}}(\boldsymbol{u},\boldsymbol{v})} - \overbrace{\sum_{i=1}^{2} \left\{ \int_{\Omega_0^{(i)}} \boldsymbol{v}^{(i)} \cdot \hat{\boldsymbol{B}}^{(i)} \,\mathrm{d}\Omega + \int_{\Gamma_N^{(i)}} \boldsymbol{v}^{(i)} \cdot \hat{\boldsymbol{T}}^{(i)} \,\mathrm{d}\Gamma \right\}}^{G^{\mathrm{ext}}(\boldsymbol{u},\boldsymbol{v})} \\ &\underbrace{- \sum_{i=1}^{2} \int_{\Gamma_C^{(i)}} \boldsymbol{v}^{(i)} \cdot \boldsymbol{T}^{(i)} \,\mathrm{d}\Gamma}_{G^C(\boldsymbol{u},\boldsymbol{v})} = 0, \end{aligned} \tag{13.42}$$

where the third summand adds the virtual work due to contact forces.

The above expression can be reformulated by considering that the contact forces acting on the two bodies have to be in balance *cf.* [Laursen, 2003, Equation 4.80]

$$\boldsymbol{T}^{(2)}(\bar{\boldsymbol{Y}}(\boldsymbol{X}))d\Gamma_C^{(2)} = -\boldsymbol{T}^{(1)}(\boldsymbol{X})d\Gamma_C^{(1)}. \tag{13.43}$$

Using this equality, the two separate integrals over the two contact surfaces can be joined in one integral of the following form

$$\begin{aligned} G^C(\boldsymbol{u},\boldsymbol{v}) &= -\int_{\Gamma_C^{(1)}} \boldsymbol{t}^{(1)} \cdot \left[\boldsymbol{v}^{(1)}(\boldsymbol{X}) - \boldsymbol{v}^{(2)}(\bar{\boldsymbol{Y}}(\boldsymbol{X}))\right] \,\mathrm{d}\Gamma \\ &= -\int_{\Gamma_C^{(1)}} p_N \boldsymbol{\nu} \cdot \left[\boldsymbol{v}^{(1)}(\boldsymbol{X}) - \boldsymbol{v}^{(2)}(\bar{\boldsymbol{Y}}(\boldsymbol{X}))\right] \,\mathrm{d}\Gamma. \end{aligned} \tag{13.44}$$

Comparing the integrand with the definition of the gap function in (13.37) leads to the definition of the *virtual gap function* as the difference of the two trial functions *cf.* [Laursen, 2003, Equation 4.83]

$$\delta g = -\boldsymbol{\nu} \cdot \left[\boldsymbol{v}^{(1)}(\boldsymbol{X}) - \boldsymbol{v}^{(2)}(\bar{\boldsymbol{Y}}(\boldsymbol{X})) \right]. \tag{13.45}$$

In this way, the virtual work of the normal contact can be written compactly as

$$G^C(\boldsymbol{u}, \boldsymbol{v}) = \int\limits_{\Gamma_C^{(1)}} p_N \delta g \, \mathrm{d}\Gamma. \tag{13.46}$$

To complete the mathematical model, the contact pressure p_N has to be expressed in terms of the penetration g. Different strategies for this step are discussed in the literature, the most prominent being the Lagrange multiplier method, the penalty method, the augmented Lagrange method, and the Barrier method. A comprehensive review and comparison is given in e.g. [Wriggers, 2006, Sections 6.3 and 6.4].

In this work, the penalty approach is chosen. In this strategy, the contact inequality (13.41) is regularized by assuming that the contact pressure increases linearly with the penetration

$$p_N = \epsilon \langle g \rangle. \tag{13.47}$$

Here, the operator $\langle \cdot \rangle$ denotes the Macaulay brackets

$$\langle x \rangle = \begin{cases} x & \forall x \geq 0 \\ 0 & \forall x < 0 \end{cases}, \tag{13.48}$$

and ϵ denotes a user-specified *penalty parameter*.

13.1.3 Computational aspects

For the numerical simulation of contact problems, the previously outlined continuous model has to be discretized. The following section gives a brief overview of the approaches applied in this work. For the sake of brevity, the different aspects are not discussed in detail. More information can be found in e.g. [Laursen, 2003, Sections 2.4.5.2 and 4.4.2], [Belytschko, 2000, Chapter 4], [Konyukhov and Izi, 2015, Chapter 6] and [Wriggers, 2006, Section 6.5].

13.1.3.1 Linearization

The first challenge is that the contact problem is non-linear due to the non-linear kinematics and the non-linear contact condition. For this reason, the problem has to be solved iteratively. As discussed in the context of cohesive delamination in Chapter 11, the Newton Raphson scheme is the prevalent approach for this aim.

To this end, the directional derivative of the virtual work has to be computed as follows

$$\Delta G(\boldsymbol{u}, \boldsymbol{v}) := \left. \frac{d}{d\alpha} \right|_{\alpha=0} \left[G(\boldsymbol{u} + \alpha \Delta \boldsymbol{u}, \boldsymbol{v}) \right]. \tag{13.49}$$

As discussed in the previous section, the virtual work is composed of an internal part G_{int}, an external part G_{ext}, and a contact contribution G_{C}.

Following e.g. [Holzapfel, 2000, Equation 8.81] or [Laursen, 2003, Equation 2.261], the directional derivative of the internal part reads

$$\Delta G^{\text{int}}(\boldsymbol{u}, \boldsymbol{v}) = \int_{\Omega_0} \underbrace{\text{Grad}\boldsymbol{v} : \text{Grad}\Delta\boldsymbol{u}\boldsymbol{S}}_{\text{geometric stiffness}} + \underbrace{\boldsymbol{F}^{\top}\text{Grad}\boldsymbol{v} : \boldsymbol{C} : \boldsymbol{F}^{\top}\Delta\boldsymbol{u}}_{\text{material stiffness}} \, \text{d}\Omega. \tag{13.50}$$

For the external part G_{ext}, the external forces are assumed to be independent of the solution $\boldsymbol{u}$. Therefore, the directional derivative is zero

$$\Delta G^{\text{ext}}(\boldsymbol{u}, \boldsymbol{v}) = 0. \tag{13.51}$$

As shown in e.g. [Laursen, 2003, Section 4.4.2], the directional derivative of the contact contribution reads

$$\Delta G^{C}(\boldsymbol{u}, \boldsymbol{v}) = \Delta \int_{\Gamma_C^{(1)}} p_N \delta g \, \text{d}\Gamma = \int_{\Gamma_C^{(1)}} \Delta \left[p_N \delta g \right] \, \text{d}\Gamma = \int_{\Gamma_C^{(1)}} \Delta p_N \delta g \, \text{d}\Gamma + \int_{\Gamma_C^{(1)}} p_N \Delta \delta g \, \text{d}\Gamma. \tag{13.52}$$

Konyukhov and Izi [2015] denote the first integral on the right hand side as the "main part" since it expresses "the constitutive law of the normal interaction force", while the second integral captures the curvature and the rotation of the contact interface. As discussed in [Konyukhov and Izi, 2015, Remark 6.3.3], these higher order derivatives have only secondary importance on the convergence of the Newton-Raphson scheme and are therefore omitted in the scope of this work. Details on this term can be found in e.g. [Laursen, 2003, Equation 4.96] or [Konyukhov and Izi, 2015, Section 6.3.1]

As shown in [Laursen, 2003, Equation 4.95], the directional derivative of p_N is given by

$$\Delta p_N = \Delta \left\{ \epsilon \langle g \rangle \right\} = H(g) \epsilon \Delta g, \tag{13.53}$$

where $H(\cdot)$ denotes the Heaviside function. Accordingly, the directional derivative of the virtual contact work is given by

$$\Delta G^{C}(\boldsymbol{u}, \boldsymbol{v}) = \int_{\Gamma_C^{(1)}} \delta g H(g) \epsilon \Delta g \, \text{d}\Gamma. \tag{13.54}$$

Based on a first or current estimate of the solution $\boldsymbol{u}^j$, the non-linear problem of contact mechanics can be solved iteratively by solving the linearized equation

$$G_{\text{int}}(\boldsymbol{u}^j, \boldsymbol{v}) + G_{\text{ext}}(\boldsymbol{u}^j, \boldsymbol{v}) + G_{\text{C}}(\boldsymbol{u}^j, \boldsymbol{v}) + \Delta G_{\text{int}}(\boldsymbol{u}^j, \boldsymbol{v}) + \Delta G_{\text{C}}(\boldsymbol{u}^j, \boldsymbol{v}) = 0 \tag{13.55}$$

for $\Delta\boldsymbol{u}$ and updating the solution estimate

$$\boldsymbol{u}^{j+1} = \boldsymbol{u}^j + \Delta\boldsymbol{u}. \tag{13.56}$$

Discretization

For solving the linearized problem, the finite element concept is applied. This requires the discretization of the volumetric contributions G_{int} and G_{ext} as well as the contact contribution G_{C}.

Since a geometrically non-linear problem implies two different configurations, the discretization of the volumetric contributions can be performed either in the initial or in the deformed domain. The first kind of discretization is denoted as a *total Lagrangian* formulation, whereas the second kind is commonly referred to as an *updated Lagrangian* formulation. A detailed comparison of the two techniques can be found in e.g. [Belytschko, 2000, Section 4.7]. Within this work, a total Lagrangian formulation in the spatial configuration is used as discussed in e.g. [Wriggers, 2008, Section 4.2.3].

The discretization of the contact contribution can be performed in different ways applying e.g. node to surface (NTS), surface to surface (STS), or Gauss-point-to-surface (GPTS) techniques. A comprehensive review of these approaches, discussing the advantages and disadvantages of the different methods, can be found in e.g. [De Lorenzis et al., 2014].

Within this work, a GPTS-like approach is used to capture the contact contribution. To this end, the slave interface is segmented conformingly to the element boundaries on the slave side, and an appropriate number of Gaussian integration points is distributed on each segment. For each integration point $\boldsymbol{X}$, the corresponding contact point $\bar{\boldsymbol{Y}}$ is found on the master side, and the contact status is check by evaluating the gap function. If penetration is detected, the contact contributions are evaluated and added to the integral. A more detailed discussion of this approach can be found in e.g. [Fischer and Wriggers, 2005].

13.2 Dynamic *hp*-refinement for moving contact[1]

As discussed in the introduction of this chapter, an accurate resolution of the contact stresses requires a refinement of the discretization at the boundary of the contact zone. In a generic setup in which the contact position might change, the following challenges are encountered: a) the boundary of the contact zone has to be detected automatically, such that b) the refinement can be guided accordingly, and c) the discretization has to be updated during the course of the simulation to account for the possible change of the contact position. The following section outlines the solution technique chosen to resolve these three challenges within this work.

13.2.1 Automatic detection of the contact zone boundary

For the automatic detection of the contact zone boundary—denoted by C in the following—the contact status serves as the refinement criterion. In a discrete setting, this implies to distribute an appropriate number of sample points on the slave contact interface Γ_C, and to compute the respective gap value.

A natural idea is to use the sign of the gap value to identify C. However, this approach turns out to be not robust as the "numerical zero" can have positive or negative sign. Therefore, the sign-change does not identify C but the complete zone in contact.

[1] This section is based on a collaboration with T. Bog *et al.* in the context of [Bog et al., 2017].

For this reason, an alternative approach is considered within this work by specifying a *gap range* $G_r = [g_{\min}, g_{\max}]$. If the gap value is within this range, the respective sample point is considered to be on the boundary of the contact zone and added to the set C.

This approach avoids the detection of "false positives" when choosing $g_{\min}$ slightly larger than zero (e.g. in the range of 10^{-12}). Moreover, the specification of $g_{\max}$ allows the user to choose a upper threshold on boundaries that are close to coming into contact. Clearly, the range G_r has to be chosen according to the stiffness of the structure and the value of the used penalty parameter.

13.2.2 Refinement towards the contact zone boundary

Having identified the set of boundary points C, the refinement has to be guided accordingly. To this end, all elements that incorporate one or more boundary points in their geometry are marked for refinement. For this geometry query, the point C is mapped into the reference space of the element K

$$\hat{C}_K = \boldsymbol{\Psi}_K^{-1}(C), \tag{13.57}$$

with $\boldsymbol{\Psi}_K^{-1}$ denoting the inverse geometry mapping of element K. The respective element is marked for refinement if the mapped point $\hat{C}_K$ lies within an *extended* index space $\hat{K}^{\text{ext}}$, which is chosen slightly larger than the actual reference space $\hat{K}$ i.e.

$$\boldsymbol{\Psi}_K^{-1}(C) \in \hat{K}^{\text{ext}}. \tag{13.58}$$

In this way, not only the one element incorporating the contact point but also the near neighbours are refined. This increases the robustness of the refinement with respect to changing contact points.

13.2.3 Dynamic discretization update

The essential difficulty when guiding the refinement is that the decision of where to refine for the *next* load step is based on the solution of the *current* load step. Hence, the refinement might be steered towards the wrong point. In principle, three different approaches can be chosen to solve this problem. A first possibility is to choose the gap range G_r large enough so that the contact zone of the next load step lies within the refinement zone of the current load step. The second approach—which goes in line with the first strategy—is to decrease the load step size to limit the progress of the contact zone between two consecutive load steps. The advantage of these two approaches is their simplicity in that they typically do not require an implementational change of the Newton-Raphson solver. However, both strategies might become computationally infeasible if the contact zone is changing its position rapidly.

A third possibility—applied in this work—is the use of a *repeated* Newton-Raphson scheme, in which each load step is computed several times with an updated discretization. This approach is illustrated in Algorithm 7. Although the repeated computation implies additional computational effort, it has to be noted that only the discretization changes within the mesh adaption steps, whereas the load level remains constant. Therefore, the previous result of the *current* load step serves as the initial guess for the *same* load step. For this reason, it can be expected that the mesh adaption steps do not require a significant number of iterations.

Algorithm 7 Repeated Newton-Raphson scheme

```
// load step loop
for each load step iLoad
{
  // mesh adaption loop
  for each mesh adaption step iMesh
  {
    // Newton iteration loop
    while residual > tolerance
    {
      compute tangent stiffness
      solve system: K_T du = f
      update solution u = u + du
    }
    // refine mesh based on current solution
    mark elements at contact zone boundary
    refine marked elements
  }
}
```

13.3 Numerical examples[2]

To analyze whether the multi-level *hp*-refinement technique is suitable to resolve the complex stress states emerging in contact mechanics, the discretization scheme is applied to two two-dimensional test cases with normal contact against a rigid plane. The aim of the first test case is to assess whether the *hp*-refinement scheme is able to yield an exponential decay of the approximation error. To this end, the well known unilateral Hertz benchmark is chosen. The aim of the second test case is to analyze whether the dynamic refinement capabilities of the suggested scheme is able to resolve contact areas that change their position during the course of the simulation. For this purpose, a benchmark involving large deformations and moving contact suggested in [Wriggers, 2006] is chosen.

13.3.1 Hertz benchmark

In this first test case, a Hertz-like contact between an elastic cylinder of infinite length and a rigid surface is computed [Hertz, 1882]. The problem is reduced to two-dimensions using plane strain conditions. As depicted in Figure 13.1, the symmetry of the problem is exploited by considering only one quarter of the cylinder. The horizontal middle-plane of the cylinder is loaded by a displacement $\hat{u}$ in vertical direction, which causes a contact between the boundary of the cylinder and the rigid surface positioned at $y = -R$. Due to the small scale of the applied displacement $\hat{u}$, the deformation of the cylinder remains in the linear regime. For this reason, the theory of linear elasticity outlined in Section 2.1 is augmented by the contact conditions discussed in the first part of this chapter to compute the deformation resulting from the applied displacement.

For the numerical approximation of the analytical solution, the domain is discretized by an initial mesh consisting of the three elements depicted in Figure 13.1. The outer two elements

[2]This section is based on a collaboration with T. Bog *et al.* in the context of [Bog et al., 2017].

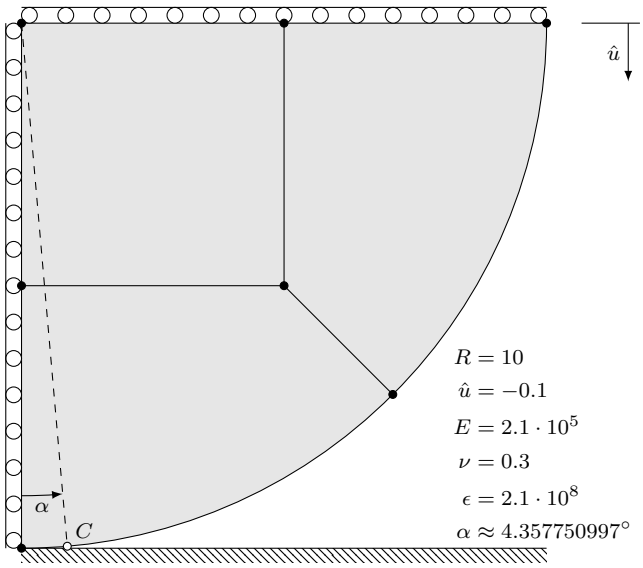


Figure 13.1: Setup and discretization of Hertz-benchmark (plane strain).

are blended on the exact arc geometry using a quasi-regional mapping, as discussed in the context of p-FEM in e.g. [Királyfalvi and Szabó, 1997]. It has to be noted that this initial mesh is *not* biased with respect to the contact expected at point C. To account for the non-linear nature of the contact, the displacement boundary condition is applied in five equidistant load steps, and each load step is solved iteratively using a Newton-Raphson scheme up to an accuracy of 10^{-8}.

To study the approximation quality, a convergence study is performed by uniformly elevating the ansatz order p of the three elements. The approximation quality is measured by considering the deviation of the strain energy

$$\mu = \sqrt{\frac{|\Pi_{\text{ref}} - \Pi_{\text{fe}}|}{\Pi_{\text{ref}}}} \cdot 100\%, \tag{13.59}$$

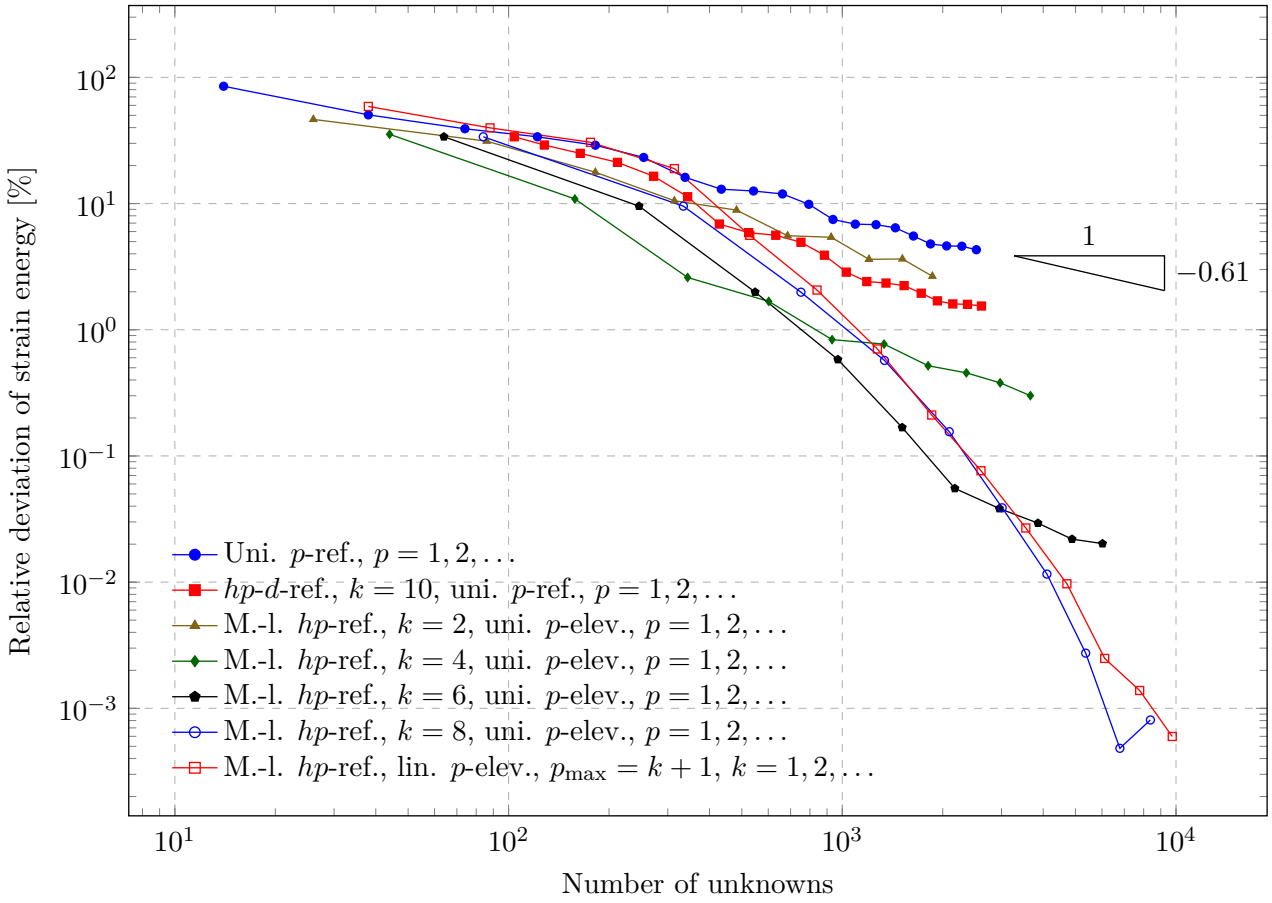
where Π_{fe} denotes the strain energy obtained with the current discretization. For the reference strain energy

$$\Pi_{\text{ref}} = 2.4094102826181199 \cdot 10^2, \tag{13.60}$$

an "overkill" computation is performed. To this end, the base mesh is refined towards the empirically determined contact point C specified in Figure 13.1 using 15 levels of multi-level hp-refinement and $p = 15$. The extended index space is chosen as $\hat{K}^{\text{ext}} = [-2, 2]^2$.

The convergence results with respect to this reference value are depicted in Figure 13.2. For an uniform p-elevation on the coarse base mesh, an algebraic increase of the approximation accuracy at a rate $\beta \approx 0.6$ can be observed. These findings are in agreement with the results presented in [Franke, 2011]. The reason for the non-exponential character of the convergence is the reduced regularity of the analytical solution mentioned in the introduction of this chapter.

This effect is also visible in Figure 13.3a, where the numerical approximation of the vertical stress component σ_{yy} obtained on the base mesh with $p = 20$ is depicted. The results show



(a) Comparison in log-log-scale

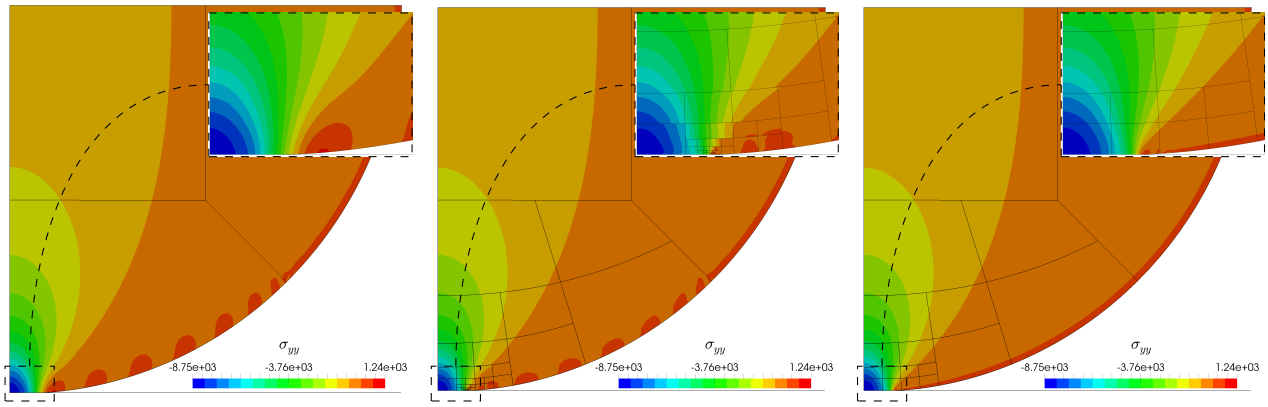
Figure 13.2: Direct comparison of the accuracy gained by different refinement strategies. Used abbreviations: uni.: uniform, lin.: linear, ref.: refinement, elev.: elevation, k: number of recursive refinements.

that the high-order shape functions can effectively resolve the complex stress field away from the contact point. However, the numerical approximation suffers from severe oscillations along the full boundary of the high-order element touching the contact point.

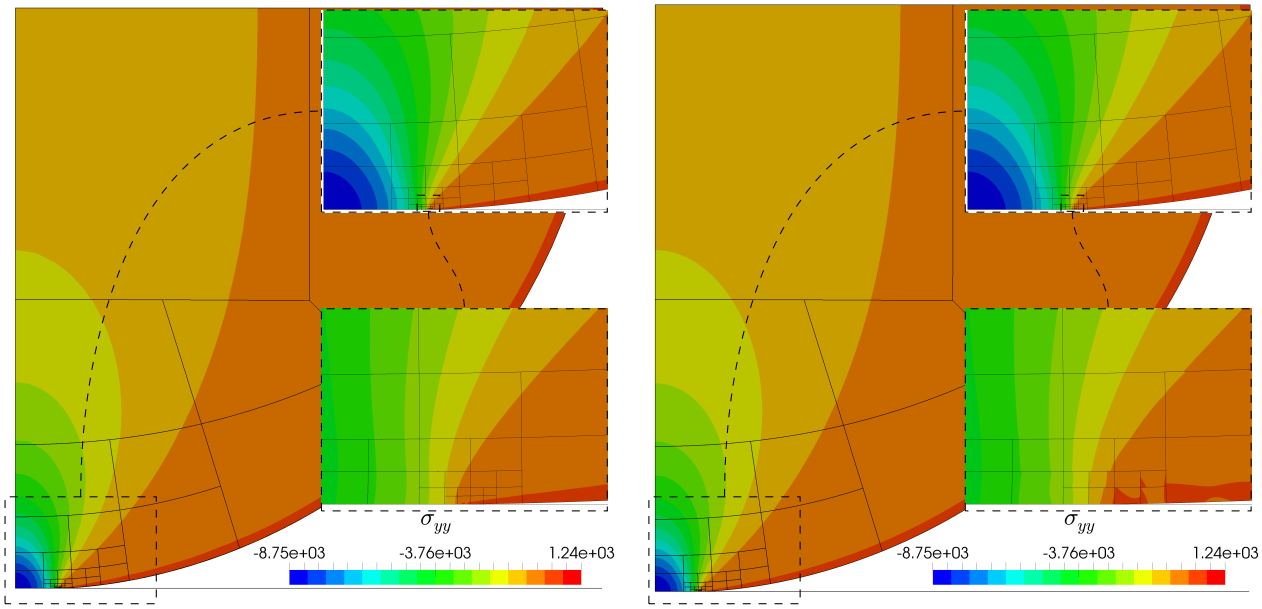
In analogy to the applications considered in the previous chapters, the reason for this poor approximation quality is that the boundary of the cylinder is composed of two part: the contact region and the free boundary. On each of the two parts, different stress profiles are present which are only C^0-continuous at the boundary of the contact zone. This reduces the regularity of the analytical solution.

As depicted in Figure 13.4, the high-order shape functions of the element incorporating this transition are not able to decouple these two solution characteristics and to effectively encapsulate the irregularity at the change of boundary conditions. Instead, the finite element solution approximates both characteristics, which causes the oscillations.

These observations motivate a refinement of the discretization in the vicinity of the contact point. In a first step, the refinement is performed using the hp-d-strategy. The numerical solution obtained with ten levels of refinement is depicted in Figure 13.3b. The results demonstrate



(a) p-FEM, $k = 0$, $p = 20$ uniform, 2,522 unknowns (b) hp-d, $k = 10$, $p = 20$ uniform, 2,612 unknowns (c) M.-l. hp, $k = 4$, $p = 9$ uniform, 2,972 unknowns



(d) M.-l. hp, $k = 10$, $p = 5$ uniform, 2,672 unknowns (e) M.-l. hp, $k = 9$, $p = 10$ linear, 2,604 unknowns

Figure 13.3: Comparison of stress distribution obtained with different discretizations.

that—despite the refinement—the oscillations remain present in the *complete* base element. Considering the findings discussed in the previous chapters, this solution characteristic is to be expected, since the hp-d-method uses overlay elements with only linear shape functions whereas the high-order modes remain on the base element. Hence, the support of the high-order functions still incorporates the change of boundary conditions, and the irregularity of the analytical solution is not shielded. Accordingly, the spread of the oscillations does not change as illustrated in Figure 13.4, and the convergence characteristic remains algebraic as demonstrated in Figure 13.2.

For this reason, the refinement strategy is changed to the multi-level hp-approach. Figures 13.3c-d compare a series of numerical approximations, in which the number of refinement

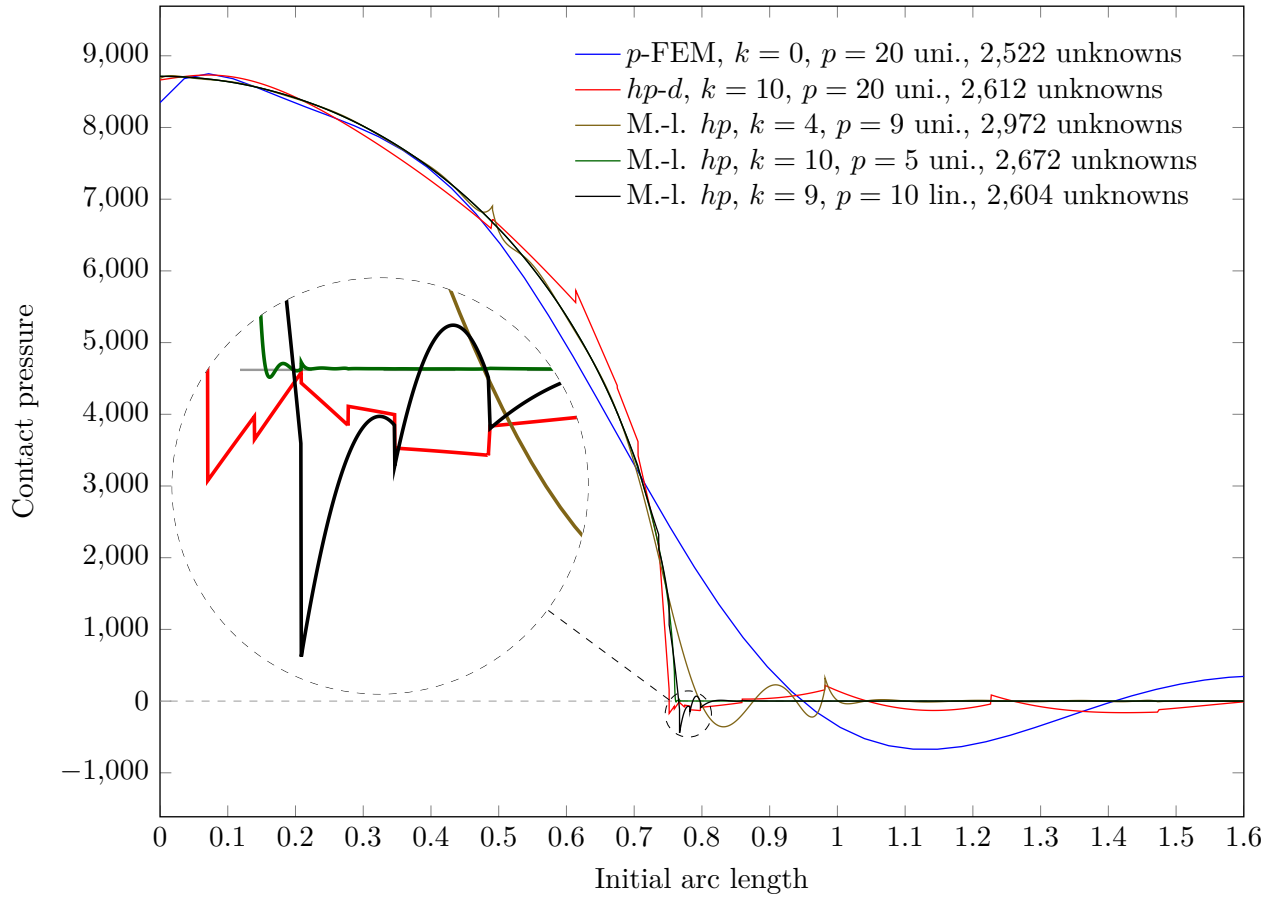


Figure 13.4: Comparison of contact pressure profiles obtained with different discretizations.

levels k and approximations order p are varied to obtain a similar number of unknowns. The results demonstrate that the reduced support of the high-order shape functions resulting from the multi-level *hp*-mesh structure effectively separates the two solution characteristics on the boundary and confines the influence of the irregularity at the boundary of the contact zone to the finest element level. In this way, the spread of the numerical oscillations is reduced to the leaf elements in the direct vicinity of the contact boundary as illustrated in Figure 13.4.

This essential difference in the solution characteristic also effects the convergence of the numerical approximation. As demonstrated in Figure 13.2, the approximation accuracy increases exponentially in the pre-asymptotic regime when performing a uniform order elevation on a pre-refined multi-level *hp*-mesh. Only when the ansatz order exceeds the optimal value of the respective number of refinements, the convergence turns algebraic in the asymptotic limit. In analogy to the benchmarks discussed in Chapter 8, this turning point can however be shifted to higher accuracies by increasing the number of pre-refinements.

As mentioned in the previous chapters, the reason for the algebraic convergence in the asymptotic regime is that the error contribution of the element incorporating the irregularity at the boundary of the contact zone only decays algebraically. For this reason, the final convergence study of this section increases the approximation quality by refining the elements in the vicinity of the contact point C and *simultaneously* elevating the ansatz order of all other

elements. Following the idea of [Gui and Babuška, 1986b], the approximation order is set to $p_{\min} = 1$ for the finest elements and then increased linear away from the irregularity up to a maximal order of $p_{\max} = k + 1$. The results presented in Figure 13.2 demonstrate that this simultaneous change of h and p yields an exponential increase of the approximation accuracy without an asymptotic decay to an algebraic convergence.

13.3.2 Moving contact

Having compared the convergence properties of the suggested refinement schemes in the previous benchmark, the second example aims at assessing the dynamic refinement capabilities of the multi-level *hp*-approach in the context of contact mechanics. To this end, the thin, bi-material half ring benchmark introduced by Wriggers [2006] is considered. As depicted in Figure 13.5, the half ring is fixed in the horizontal direction and displaced in the vertical direction on the top boundary. This displacement causes a contact between the outer ring surface and the rigid plane positioned at $y = -(R + t)$. Following e.g. [Bonet and Wood, 2008, Equations 6.53 and 6.61], a neo-Hookean type of hyper elastic strain energy function is chosen to model the stress-strain relation

$$W = \frac{\mu}{2}\left(\frac{\operatorname{tr}(\boldsymbol{C})}{J^{2/3}} - 3\right) + \frac{\kappa}{2}(J-1)^2 . \tag{13.61}$$

Here, μ and κ denote the shear and bulk modulus of the considered material, respectively. The problem is reduced to two dimensions using plane strain conditions.

For the numerical approximation of the analytical solution, the symmetry of the ring is exploited by considering only the right half. The x-displacement on the left boundary at $x = 0$ is constrained accordingly. This reduced domain is discretized using four elements in the thickness and 32 elements in the circumferential direction. In this way, the elements resolve the material interface in the thickness direction. To account for the curved geometry, the elements are blended on the exact arc using a quasi-regional mapping [Királyfalvi and Szabó, 1997]. The approximation order of this base mesh is set to $p = 7$. The displacement

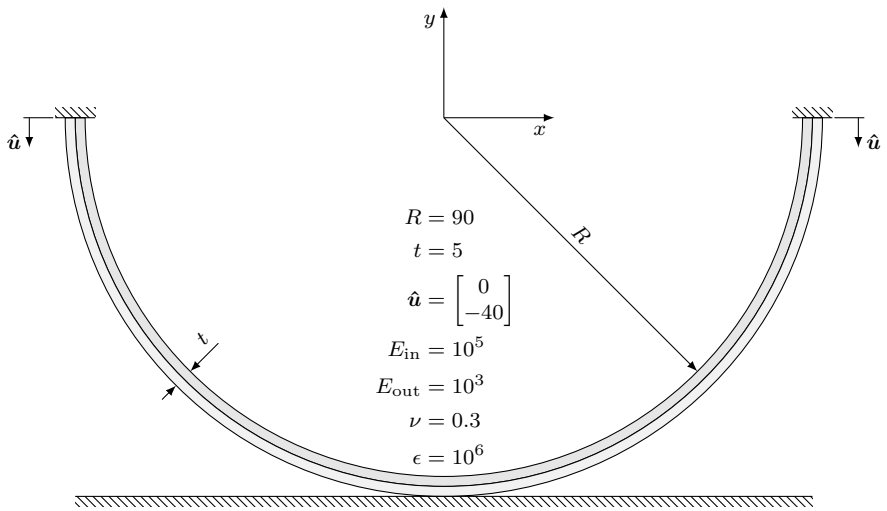
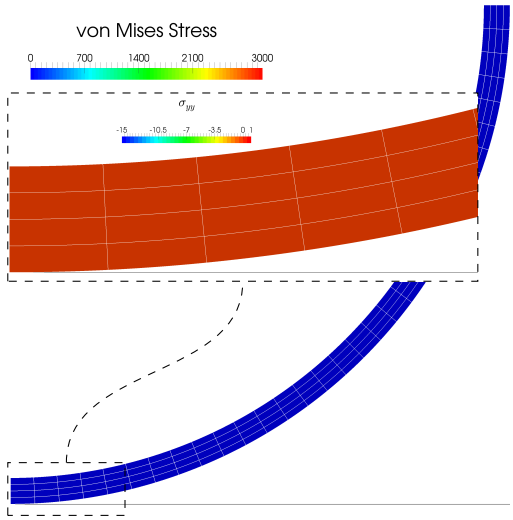
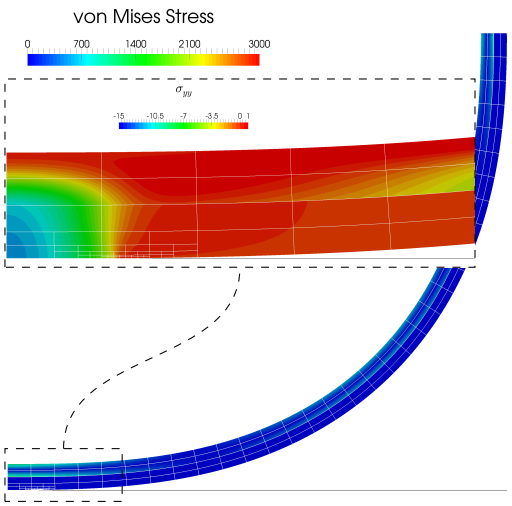


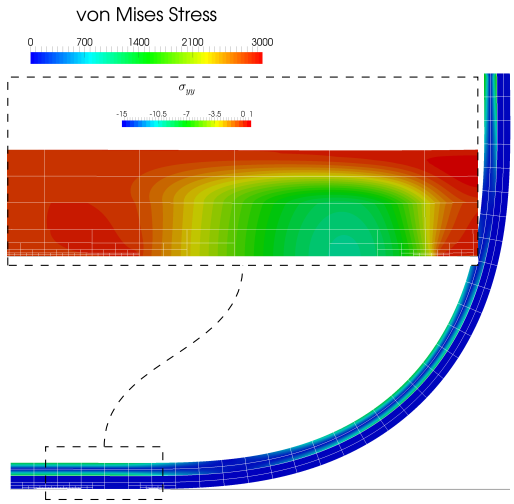
Figure 13.5: Setup of bi-material ring example following [Wriggers, 2006, Section 9.5.5].



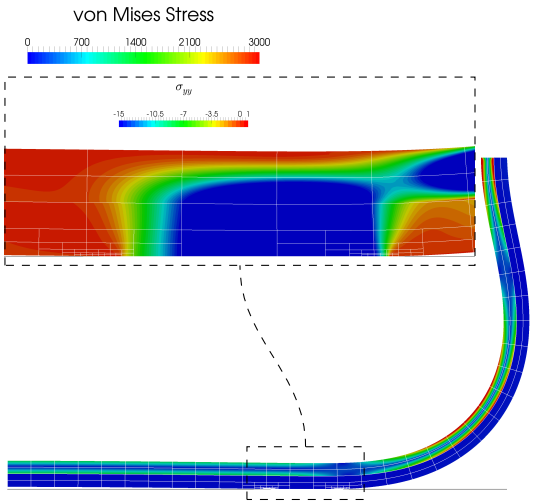
(a) Load step 0



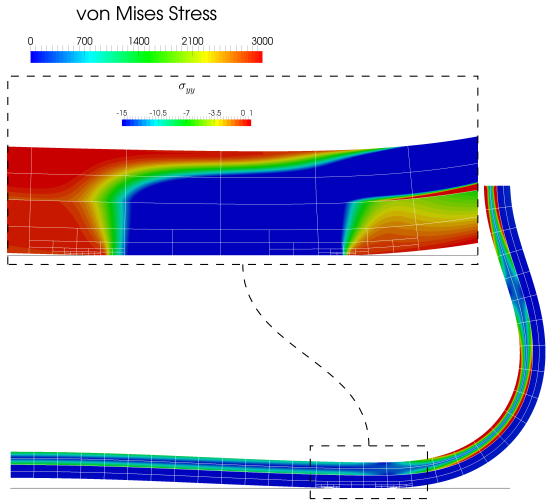
(b) Load step 32



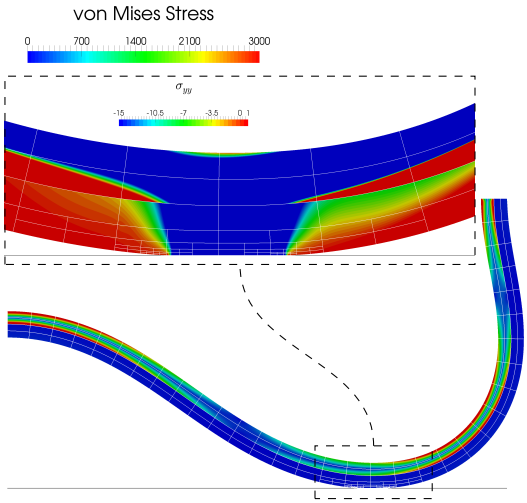
(c) Load step 64



(d) Load step 128



(e) Load step 150



(f) Load step 160

Figure 13.6: Deformation of bi-material ring for selected load steps (4×32 base elements, $p = 7$, $k = 4$, number of unknowns: max = 28,282, min = 13,344, average = 19,217).

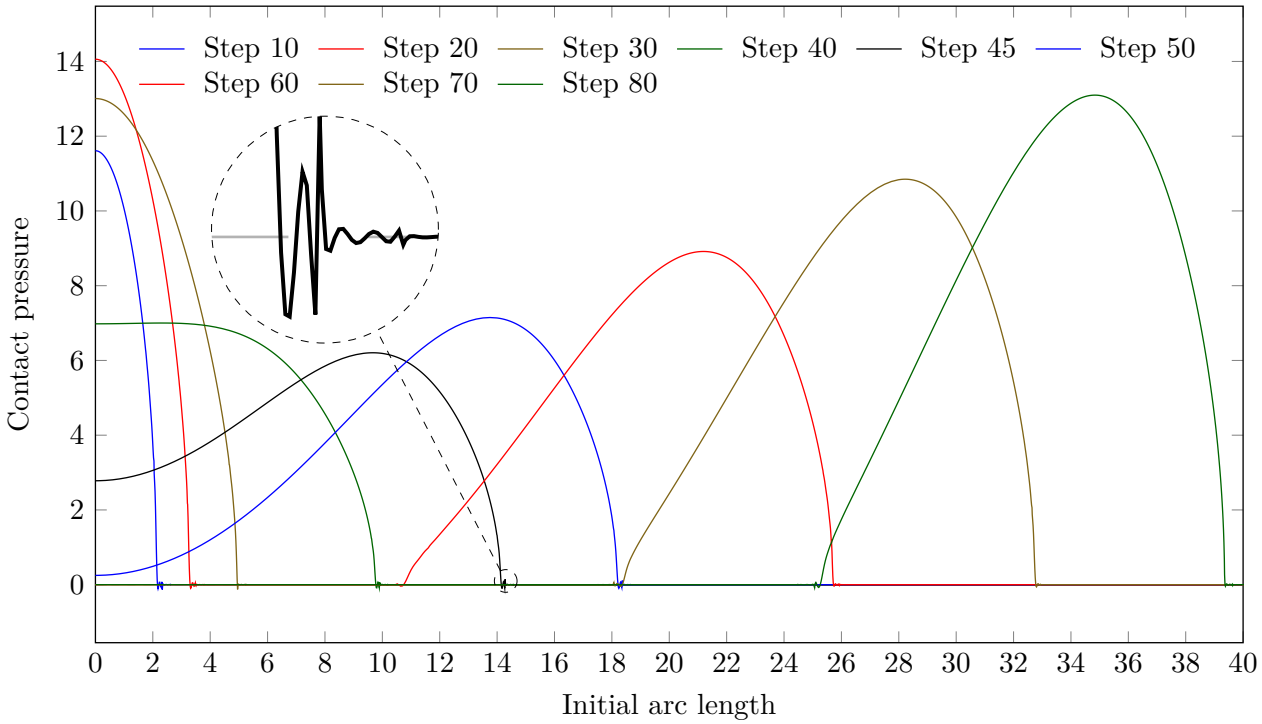


Figure 13.7: Contact pressure along contact interface for selected load steps.

boundary condition is applied in 160 equidistant load steps, and each load step is solved using a Newton-Raphson iteration scheme up to a residual tolerance of 10^{-6}.

To capture the expected stress concentration within the contact zone, the base discretization is refined using four levels of multi-level *hp*-overlay meshes. The boundary of the contact zone is determined automatically using the approach outlined in Section 13.2. The gap range is chosen as $G_r = [10^{-12}, 10^{-2}]$ and four mesh adaption steps are performed.

The numerical approximation obtained with this approach is depicted for selected load steps in Figure 13.6. The results demonstrate that the coarse, high-order mesh effectively captures the global deformation and stress state. At the same time, the suggested repeated Newton-Raphson scheme reliably steers the refinement to follow the moving boundaries of the contact zone. The contact pressure profiles depicted for selected load steps in Figure 13.7 demonstrates that—also in the case of a dynamically changing discretization—the oscillations in the computed contact pressure remain confined to the finest element level.

The computational overhead of the suggested repeated Newton-Raphson scheme is depicted in Figure 13.8. The results demonstrate that the first solution of a load step (mesh adaption step 1) requires the highest number of Newton-Raphson iterations. This is to expected since the load level increases in this step. In the remaining three mesh adaption steps, only the discretization is updated. The depicted results imply that this only introduces a minor residual that can be reduced below the specified tolerance within two or tree iterations in the second mesh adaption step. The adaption steps two and three only require one iteration each, which indicates that—for the present example—one predictor and one corrector step are sufficient.

Depending on the mesh structure of the respective load step, the number of unknowns varies between 20 and 30 thousand. In contrast, a static *a priori* discretization would require to refine

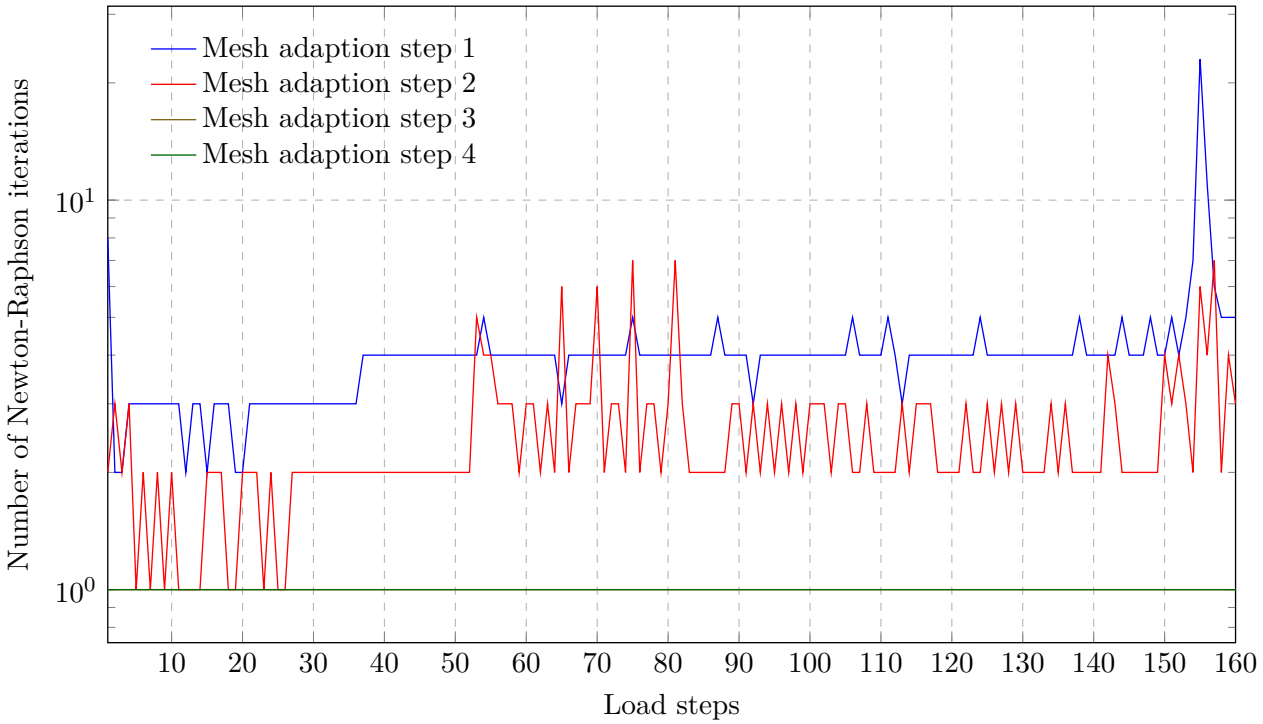


Figure 13.8: Number of Newton-Raphson iterations necessary to converge to a residual error of 10^{-6} for the four mesh adaption steps.

all elements along the outer boundary of the ring as almost the *complete* outer boundary comes temporarily into contact. Such a globally refined mesh of the same element size would yield more than 150 thousand degrees of freedom. This difference shows the advantage of decoupling the *potential* contact interface from the *actual* contact zone, a feature incorporated into the dynamic discretization possibilities of the multi-level *hp*-scheme.

Chapter 14

Conclusion and outlook

The central objective of the present work is to help to alleviate the algorithmic and implementational complexity of *hp*-finite elements. To this end, the multi-level *hp*-version of the Finite Element Method is applied.

This discretization method aims at formalizing an *hp*-refinement scheme that a) yields the same approximation quality as conventional *hp*-approaches but b) reduces the complexity necessary to handle irregular meshes. For this purpose, the classical *refine-by-replacement* strategy of conventional *hp*-formulations is dropped in favor of a *refine-by-superposition*. The essential idea of this alternative discretization concept is to locally increase the approximation quality of a coarse base-discretization by *superposing* a fine-scale overlay mesh in domains of interest. With this change of the refinement paradigm, the global continuity of the final approximation can be ensured easily by applying homogeneous Dirichlet boundary conditions on the overlay mesh. In this way, the problem of hanging nodes is avoided by construction, which allows for the definition of a simple rule-set that ensures the compatibility and linear independence of the global basis functions.

The idea of this discretization approach was pre-published in [D'Angella et al., 2016; Zander et al., 2016, 2015, 2017] by the author of this work. Following these publications, the first part of this work provides a complete and augmented summary of the method's concepts and puts these into the context of conventional *hp*-discretization approaches. The second part of this work presents a systematic analysis of the approximation properties of the multi-level *hp*-refinement method in different fields of engineering relevance. To this end, the example suite presented in the aforementioned pre-publications is summarized and supplemented by additional, more detailed numerical studies to assess the approximation quality of the suggested method in more depth.

14.1 Summary of scientific conclusions

The present work applies the suggested refine-by-superposition approach in different application fields of engineering interest. In the following, the central findings are systematically summarized.

14.1.1 Comparison to conventional *hp*-approaches

The central observation of this work is that the suggested refinement by high-order overlay meshes yields an exponential decay of the approximation error even in the presence of singularities. The presented results demonstrate this advantage in the context of one-, two- and three-dimensional applications in the field of linear heat conduction, linear elasticity, convection-dominated transport problems and computational contact mechanics. Moreover, runtime measurements for two- and three-dimensional examples show that this convergence characteristic can be obtained not only with respect to the number unknowns but also with respect to the computational time required to integrate, assemble, and solve the system of finite element equations. A further comparison of other numerical properties reveals that the refine-by-superposition approach also does not have any significant detrimental effect on the conditioning of the resulting equation system, but even shows a slightly better characteristic than conventional *p*- and *hp*-approaches. Furthermore, the presented results demonstrate that the superposition-based refinement scheme can be combined successfully with a conventional error and smoothness estimator to yield an automatic feedback scheme that is adaptive with respect to the convergence rate according to Babuška [1986]. Moreover, the refinement is not restricted to simply-shaped benchmark domains but is also applicable in the context of curved and unstructured meshes. In the author's opinion, the multi-level *hp*-refinement scheme therefore meets the major quality criteria of an *hp*-discretization scheme.

14.1.2 Comparison of *hp*-*d*- and multi-level *hp*-refinement

The second important aspect discussed in this work is the difference of the suggested multi-level *hp*-method to its predecessor: the *hp*-*d*-refinement approach. In this context, the essential insight is that the difference in the approximation quality of the two methods results from the size of the support of the high-order shape functions.

In the case of the *hp*-*d*-method, a high-order base mesh is superposed with elements using linear shape functions only. Accordingly, the support of the high-order shape functions remains unchanged. In contrast, the multi-level *hp*-approach employs high-order overlay elements, which renders the support of the high-order shape functions minimal.

The significance of this difference is demonstrated in the context of different two- and three-dimensional benchmark examples. In all cases, the multi-level *hp*-method achieves the aforementioned exponential decay of the approximation error, while the *hp*-*d*-approach can only attain an algebraic convergence.

Furthermore, the smaller support of the high-order shape functions obtained by the multi-level *hp*-refinement scheme allows the effective separation of distinct physical solutions. The resulting advantage is demonstrated in the context of sharp boundary layers common in convection dominated transport problems, changing boundary conditions caused contact, and thin, embedded inclusions in the context of the Finite Cell Method. The separation of the respective solution characteristics is not possible when refining using the *hp*-*d*-paradigm as the high-order shape functions remain active on the coarse base elements. This causes an oscillatory approximation in the coarse base elements that incorporate the distinct solution characteristics. In the author's opinion, the multi-level *hp*-method therefore presents an advance in the context of superposition based *hp*-refinement.

14.1.3 Dynamic discretization

The third important observation is that the followed refine-by-superposition schemes eases the implementational complexity of the scheme. This allows for a highly flexible discretization kernel featuring arbitrary irregular meshes, which can be refined and coarsened continuously throughout the simulation runtime. The present work discusses the advantages of this dynamic discretization capabilities exemplarily for the following two- and three-dimensional examples.

Benchmark examples with moving steep gradients or discontinuities

Natural application fields for a dynamically changing discretization are transient problems. In the context of parabolic and hyperbolic partial differential equations, the presented results demonstrate that adjusting the mesh structure in every time-step of a transient simulation allows the accurate resolution of solution features that change their position with time. The discussed benchmark show that this is of advantage when considering the propagation of sharp wave fronts or stress discontinuities caused by a moving load.

Resolution of embedded inclusions using the Finite Cell Method

In a similar way, the analyzed examples demonstrate that the dynamic discretization allows the resolution of the heat flow around moving inclusions when combining the suggested approach with the concept of the Finite Cell Method. The essential insight in this context is that—even if the analytical solution is *not* singular but smooth—the support of the high-order shape functions significantly effects the approximation quality of the numerical result. The reason for this importance in the context of the FCM is that coarse base elements might intersect the physical domain on two different sides of the embedded inclusion. Therefore, the used shape functions have to account for two potentially different analytical solutions. The presented studies show the resulting deterioration of the approximation quality in the form of severe oscillations. Typically, this problem does not occur when the inclusion does not change its position as the mesh can be pre-refined accordingly. However, this *a priori* refinement is no longer possible if the inclusion is moving. Accordingly, the dynamic discretization capabilities of the suggested refinement technique are of special importance in this context. However, the support of the high-order shape functions remains unchanged when refining in the *hp-d*-sense with only linear overlay meshes. Accordingly, the *hp-d*-refinement does *not* solve the original cause of the problem. This is different for the multi-level *hp*-version of the overlay refinement, since this approach decreases the support of the high-order shape functions. The presented two- and three-dimensional benchmark results confirm that the *hp-d*-refinement has only a very limited effect on the approximation quality, while the multi-level *hp*-approach effectively prevents oscillations in the numerical solution. In the author's opinion, this renders the multi-level *hp*-strategy as a suitable method to resolve moving inclusions embedded in a non-boundary conforming finite cell mesh.

Simulation of cohesive crack growth

A third application field of the dynamic discretization analyzed in this work is the de-bonding of laminated structures. To this end, the refinement scheme is combined with a cohesive failure model to capture crack initiation and progression. The major challenge of the numerical

handling of these models is their intrinsic discretization sensitivity, which was reported in many studies before. The core of this problem is that the interface elements used to model the state of the crack are either *fully* closed or opened along their *full* length. Accordingly, the crack can only advance element by element, which leads to sudden releases of energy and unphysical oscillations in the load-displacement curve. For this reason, the numerical simulation of cohesive crack growth demands for a fine resolution of the fracture process zone. Depending on the used materials of the laminated structure, this zone has the length of a few millimeters. In contrast, the length of the laminated interface itself can have the size of meters. This scale difference renders an *a priori* refinement of the discretization along the full potential crack path computationally unfeasible. Instead, this problem requires a non-uniform mesh refinement in vicinity of the crack tip to *decouple* the global discretization of the structure from the local resolution of the fracture process zone. However, the crack is changing its position while it is progressing. This requires a continuous adjustment of the mesh refinement throughout the simulation. Hence, the suggested refine-by-superposition scheme lends itself for this problem since the overlay mesh can be moved along to follow the propagating crack while the base-mesh is left unchanged.

The essential insight gained by the presented studies is that—also for this application—the size of the support of the high-order shape functions essentially influences the approximation quality of the numerical simulation. The reason follows in analogy to the aforementioned cases: high-order shape functions with a large support along the laminate interface have to account for two different solution states on the two sides of the crack tip. Therefore, a pure p-elevation is not sufficient to accurately resolve the propagating crack. Instead, this problem requires a high-order approximation *and* a fine discretization to capture the complex solution within the fracture process zone. Since the *hp-d*-approach does not decrease the support of the high-order shape functions, the transition from the closed to the cracked interface can not be captured more accurately when using this refinement technique. The analyzed examples confirm that the *hp-d*-scheme improves the result quality only marginally. In particular, the results show that the *hp-d*-approach does *not* prevent the discretization induced oscillations in the load-displacement curve. As in the previous cases, the situation is different when using the multi-level *hp*-refinement since this approach reduces the support of the high-order shape function to the finest element level. The presented results demonstrate that this allows for an accurate resolution of the complex stress state ahead of crack front, which is free of artificial, discretization-induced oscillations. At the same time, the aforementioned decoupling of the global and local discretization scales significantly reduces the sizes of the numerical problem and with it the computational times. For the analyzed benchmark, this manifested itself in a speedup of factor five compared to a global *a priori* refinement. In the author's opinion, the multi-level *hp*-version of the overlay refinement scheme therefore presents an advance compared to the *hp-d*-approach also in this application field.

Moving contact

The fourth application field of the dynamic discretization analyzed in this work is the normal contact of an elastic continuum subjected to larger deformations. From a high-order discretization point of view, the first essential challenge in this context is the reduced regularity of the solution at the boundary of the contact zone. The second challenge is that this contact zone might change its position during the course of a non-linear simulation.

The presented results demonstrate that the suggested multi-level *hp*-scheme is able to effectively encapsulate the irregularity of the solution by geometrically grading the element size towards the boundary of the contact zone. In this way, oscillations in the numerical solutions are confined to the finest element level, and an exponential decay of the approximation error is obtained. It is shown that this is not possible when refining with the *hp-d*-scheme since the support of the high-order shape functions remains on the base mesh.

Moreover, it is demonstrated that the suggested refinement algorithm automatically detects the irregularity of the solution. A continuous update of the discretization in every load step of the non-linear simulation allows the refinement to stay local to the moving boundary of the contact zone. This dynamic change of the discretization *decouples* the length of the *potential* contact interface form the *actual* contact zone. For the presented two-dimensional example, this reduces the number of unknowns by one order of magnitude compared to a static *a priori* refinement along the *full* length of the potential contact interface. In the author's opinion, this clearly demonstrates the advantages of the suggested dynamic *hp*-refinement in the context of this highly relevant application field.

14.2 Future research

Clearly, this work does not provide a full analysis suite but only provides a proof of concept with many points still open. The following section aims at summarizing these in a structured way.

14.2.1 Extensions of refinement capabilities

Within this work, the *h*-refinement of an element is understood as a bi-section of the respective element in every spatial dimension. While this is sufficient for the applications considered in this work, the following extensions might be valuable in other scenarios.

14.2.1.1 Arbitrary geometric refinement

In the present work, the refinement of elements is understood as an (isotropic) sub-division in the middle of the element. Accordingly, the ration q of the geometric progression is $\frac{1}{2}$. Although this is sufficient for an exponential convergence of the approximation error, the analysis presented by [Gui and Babuška, 1986b] shows that the convergence rate can be further improved by a more general sub-division. This sub-division in an arbitrary ratio q can be realized without conceptual problems also in the context of superposition-based refinement by a simple geometric re-location of the overlay nodes.

14.2.1.2 An-isotropic refinement

Instead of refining an element isotropically in all spatial dimensions, an an-isotropic refinement sub-divides an element in only selected directions. This is of major advantage when considering e.g. boundary layers or edge singularities. Here, the refinement is only necessary in one- or two-spatial dimensions, respectively. Accordingly, an isotropic refinement is clearly not optimal. For this reason, most *hp*-frameworks support an an-isotropic refinement.

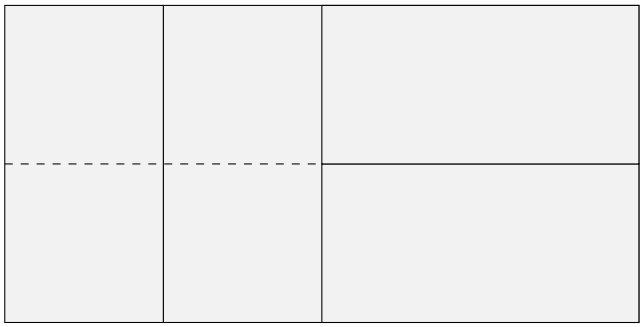


Figure 14.1: Non-matching discretization due to an-isotropic refinement. The dashed line illustrates a possible second bi-section that resolves the in-compatibility.

In principle, an an-isotropic refinement is also possible when following the suggested refine-by-superposition approach. To this end, the discretization kernel has to be extended such that quadrilateral faces can be overlayed by two *or* four sub-faces. Similarly, hexahedral volumes have to be superposable by two, four, *or* eight sub-volumes.

The major challenge in this context is the resolution of possible in-compatibilities. These arise when neighboring elements are sub-divided in different, non-matching directions as illustrated exemplarily in Figure 14.1. A simple solution—which can be realized also in the context of superposition-based refinement—is to restrict the refinement to only matching sub-divisions. For the scenario depicted in Figure 14.1, this implies that, as soon as the right element is refined, the left element is divided isotropically as well. In this way, the challenge of non-matching divisions is circumvented. However, this comes at the cost of additional, potentially unnecessary elements and degrees of freedom.

For this reason, most *hp*-formulations provide more sophisticated solutions for this problem. This aspect is discussed in e.g. [Demkowicz, 2007, Section 13.2]. In the author's opinion, similar solutions can also be applied for the multi-level formulation of *hp*-finite elements. However, they come at the same implementational complexity as in the context of conventional *hp*-approaches.

14.2.1.3 Other element types

The present work focuses on the refinement of quadrilateral and hexahedral elements. However, this is no conceptual restriction of the multi-level *hp*-method itself. In principle, the same concept and the same rule-set discussed in this work applies also to triangular, tetrahedral, and prism-shape elements. These element-types offer the advantage that more complex meshes created by a mesh-generator can be used. Therefore, the extension of the discretization kernel in this direction might be of advantage for applications depending on these kind of meshes.

14.2.2 Computational efficiency

One essential and everlasting task is the further increase of the computational efficiency of the scheme. This challenge can be addressed on the following different levels.

14.2.2.1 Node-level optimization

The first level is the node-level optimization of the developed code-framework. In the course of this work, many parts of the code have already been optimized, the detailed discussion thereof would go beyond the scope of this work. Examples of this work are the efficient evaluation of the non-zero shape functions in the hierarchical refinement tree, or a prediction of the sparsity pattern of the refined discretization which allows for a pre-allocation of the final system matrix. Of course, this optimization is a continuous process and requires every future developer to carefully profile her/his use of the developed program.

14.2.2.2 Parallelization

The second level is the parallelization of the code framework, which will allows the use of modern compute clusters. Here, very promising work has been done already, which currently shows an efficient scaling for the computation of the stiffness matrix on up to 128 cores. These results will be published soon in [Jomo et al., 2017].

14.2.2.3 Change of methods

A third level is the change of the applied numerical methods. Intrinsically, a change of method opens many novel research questions. This might include the following aspects.

Sum factorization

One possibility to further increase the computational performance of the numerical integration is the use of the sum factorization technique suggested in [Melenk et al., 2001]. Here, an interesting research question is how the factorization process can be implemented efficiently when using the suggested overlay refinement technique.

Pre-integration

Another approach targeting the same goal is the use of the pre-integration approach developed in [Yang, 2011; Yang et al., 2012a]. In this context, the essential question is how this approach can be combined with the multi-level *hp*-idea. While the author sees no conceptual problem in the case of a static mesh structure, the case of dynamic discretization might need for more research.

Multi-grid methods

Another problem of any finite element based solution scheme is the numerical cost associated with solving the system of linear equations. Here, not only the computational time but also the amount of required memory might be a limiting factor for large scale simulations. For this reason, a highly interesting point of research is the development of a multi-grid solution scheme that make use of the hierarchical nature of the basis functions. As discussed in e.g. [Bank et al., 1988; Mitchell, 1988, 1991, 1992; Yserentant, 1986a,b; Zienkiewicz and Craig, 1986], this combination of methods can yield a significant increase in performance. In particular using the multi-grid approach as a pre-conditioner for a conjugate gradient scheme might be beneficial not only with respect to the runtime but also with respect to the required memory.

Partitioned solution schemes

An alternative solution approach is the change from the *monolithic* formulation of the multi-level *hp*-method—suggested in this work—to a *partitioned* solution scheme. Here, the essential idea is to *not* formulate *one* global system of equations but to solve a local problem on each refinement level. The final approximation to the global solution is then determined by an iterative Gauss-Seidel approach that finds the global equilibrium.

The assets and drawbacks of this approach follow in analogy to the use of these principles in multi-physical applications such as fluid structure interaction [Felippa, 2004]. This iterative nature of the solution scheme has the potential drawback that several iteration cycles might be necessary until the solution is converged. This in particular applies for strongly coupled problems. On the other hand, the partitioned solution scheme offers a higher flexibility of the discretization. As e.g. demonstrated in [Düster et al., 2007a; Ledentsov, 2010; Ledentsov et al., 2010] for the *hp-d*-method, the split of the discretizations allows the coupling of dimensionally reduced models and superposed volumetric models. Furthermore, different time scales can be used on the separated levels. This sub-cycling like approach can be of advantage in cases where the physical state evolves significantly faster on a local than on the global scale. The first research results in this context are very promising [Korshunova, 2016].

14.2.3 Implementational challenges in other fields of application

Another point of extension is the application of the refinement method to other physical or technical problems. This is particularly interesting for cases that can profit from the ability of the method to decouple the local and the global discretization scale demonstrated in the simulation of cohesive de-bonding.

One possible application is the simulation of additive manufacturing processes such as Selective Laser Melting (SLM). Here, the intrinsic challenge is that the laser and the melting pool have the size of micrometers, while the structure might have the size of centimeters or even more. This difference in scale renders a simulation with a uniformly refined mesh computationally in-feasible. Instead, the computational effectivity significantly profits from a dynamic discretization kernel that allows to refine the mesh in the vicinity of the melting pool and to coarsen again once the laser has passed. First research results of applying the multi-level *hp*-concept in this field show very promising results and will be published soon in [Kollmannsberger et al., 2016].

Another possible applications are problems with sharp fronts and interfaces such as e.g. plasticity, waves or shocks in fluids, or phase-field formulations applied to model e.g. fracture [Ambati et al., 2014; Miehe et al., 2010], phase-change [Gonzalez-Ferreiro et al., 2014], or crystal growth [Bueno et al., 2016; Gomez and Nogueira, 2012]. Here, the change of state takes place in a very narrow band that changes its position with time. Accordingly, a high resolution is needed locally, while the remaining parts of the domain may fully profit from the use of high-order elements.

The major additional challenge in these contexts is the handling of history variables such as e.g. plastic strains or the maximum positive reference energy. This is an intrinsic problem of dynamic discretizations, since the position of the integration points change with the mesh. Accordingly, history variables cannot be simply stored on Gauss point level. Instead, suitable interpolation schemes have to be developed to transfer the plastic state between two meshes.

This aspect is subject of current research.

14.2.4 Extension to different compatibility constraints

Another highly interesting research field is the use of the multi-level *hp*-refinement scheme for applications that demand for a higher inter-element continuity. In the context of civil engineering, this includes e.g. dimensionally reduced models, such as beams, plates, and shells. Also some of the aforementioned phase-field models are based on higher-order models and therefore require a higher global continuity. In the context of mechanical or electrical engineering, a possible application of this kind is the numerical simulation of electro-magnetic effects. As demonstrated in e.g. [Demkowicz and Vardapetyan, 1998; Dubcova et al., 2010; Rachowicz and Demkowicz, 2000b, 2002; Schnepp and Weiland, 2012; Šolín et al., 2010a], the *hp*-version of the Finite Element Method yields excellent results in this highly important field of applications. For this reason, the extension of the formulated rule-set to other spaces—such as $H(\mathrm{div})$, $H(\mathrm{curl})$ or H^2—is an interesting field of future research.

Bibliography

Abedian, A., Parvizian, J., Düster, A., and Rank, E. (2013). The finite cell method for the J2 flow theory of plasticity. *Finite Elements in Analysis and Design*, 69:37–47.

Abedian, A., Parvizian, J., Düster, A., and Rank, E. (2014). Finite cell method compared to h-version finite element method for elasto-plastic problems. *Applied Mathematics and Mechanics*, 35(10):1239–1248.

Adjerid, S., Aiffa, M., and Flaherty, J. E. (1999). Computational methods for singularly perturbed systems. In *Proceedings of Symposia in Applied Mathematics*, volume 55, pages 47–84.

Ainsworth, M. (2000). *A Posteriori Error Estimation in Finite Element Analysis*. Pure and applied mathematics. Wiley, New York.

Ainsworth, M. and Oden, J. T. (1997). A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142(1–2):1–88.

Ainsworth, M. and Senior, B. (1998). An adaptive refinement strategy for hp-finite element computations. *Applied Numerical Mathematics*, 26(1):165–178.

Ainsworth, M. and Senior, B. (1999). Hp-Finite Element Procedures on Non-Uniform Geometric Meshes: Adaptivity and Constrained Approximation. In Bern, M. W., Flaherty, J. E., and Luskin, M., editors, *Grid Generation and Adaptive Algorithms*, number 113 in IMA Volumes in Mathematics and its Applications, pages 1–27. Springer, New York.

Alfano, G. and Crisfield, M. A. (2001). Finite element interface models for the delamination analysis of laminated composites: Mechanical and computational issues. *International Journal for Numerical Methods in Engineering*, 50(7):1701–1736.

Ambati, M., Gerasimov, T., and Lorenzis, L. D. (2014). A review on phase-field models of brittle fracture and a new fast hybrid formulation. *Computational Mechanics*, 55(2):383–405.

Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D. (2001). Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 35(5):1749–1779.

Babuska, I. (1973). The Finite Element Method with Penalty. *Mathematics of Computation*, 27(122):221–228.

Babuška, I. (1986). Feedback, Adaptivity, and a posteriori Estimates in Finite Elements: Aims, Theory, and Experience. In Babuška, I., Zienkiewicz, O. C., Gago, J., and de Oliviera, E. R., editors, *Accuracy Estimates and Adaptive Refinements in Finite Element Calculations*, pages 25–55. Wiley, New York.

Babuška, I. and Aziz, A. (1976). On the Angle Condition in the Finite Element Method. *SIAM Journal on Numerical Analysis*, 13(2):214–226.

Babuška, I. and Guo, B. (1988). The h-p Version of the Finite Element Method for Domains with Curved Boundaries. *SIAM Journal on Numerical Analysis*, 25(4):837–861.

Babuška, I. and Guo, B. (1996). Approximation properties of the h-p version of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 133(3-4):319–346.

Babuška, I. and Melenk, J. M. (1997). The Partition of Unity Method. *International Journal for Numerical Methods in Engineering*, 40(4):727–758.

Babuška, I. and Miller, A. (1984). A feedback finite element method with a posteriori error estimation: Part I. The finite element method and some basic properties of the a posteriori error estimator. Techincal Note BN-1031, University of Maryland, Institute for physical science and technology.

Babuška, I. and Miller, A. (1987). A feedback finite element method with a posteriori error estimation: Part I. The finite element method and some basic properties of the a posteriori error estimator. *Computer Methods in Applied Mechanics and Engineering*, 61(1):1–40.

Babuška, I. and Rank, E. (1986). An Expert-System-Like Feedback Approach in the hp-Version of the Finite Element Method. Techincal Note BN-1048, University of Maryland, Institute for physical science and technology.

Babuška, I. and Rank, E. (1987). An expert-system-like feedback approach in the hp-version of the finite element method. *Finite Elements in Analysis and Design*, 3(2):127–147.

Babuška, I. and Rheinboldt, W. C. (1978a). A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615.

Babuška, I. and Rheinboldt, W. C. (1978b). Analysis of optimal finite-element meshes in R^1. Technical Report BN-869, University of Maryland, Institute for physical science and technology.

Babuška, I. and Rheinboldt, W. C. (1979). Analysis of optimal finite-element meshes in R^1. *Mathematics of Computation*, 33(146):435–463.

Babuška, I. and Rheinboldt, W. C. (1981). A Posteriori Error Analysis of Finite Element Solutions for One-Dimensional Problems. *SIAM Journal on Numerical Analysis*, 18(3):565–589.

Babuška, I. and Suri, M. (1994). The p and h-p Versions of the Finite Element Method, Basic Principles and Properties. *SIAM Review*, 36(4):578–632.

Bangerth, W., Hartmann, R., and Kanschat, G. (2007). deal.II—A general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software*, 33(4):1–27.

Bank, R. E., Dupont, T. F., and Yserentant, H. (1988). The hierarchical basis multigrid method. *Numerische Mathematik*, 52(4):427–458.

Bank, R. E. and Sherman, A. H. (1980). A Refinement Algorithm and Dynamic Data Structure for Finite Element Meshes. Technical report, University of Texas at Austin, USA.

Bank, R. E. and Sherman, A. H. (1981). An adaptive, multi-level method for elliptic boundary value problems. *Computing*, 26(2):91–105.

Barenblatt, G. I. (1962). The Mathematical Theory of Equilibrium Cracks in Brittle Fracture. In Dryden, H., von Kármán, T., Kuerti, G., van den Dungen, F. H., and Howarth, L., editors, *Advances in Applied Mechanics*, volume 7, pages 55–129. Elsevier.

Bartels, S. (2016). *Numerical Approximation of Partial Differential Equations*, volume 64 of *Texts in Applied Mathematics*. Springer International Publishing.

Bathe, K. J. (2007). *Finite Element Procedures*. Prentice Hall, New Jersey.

Bazilevs, Y., Calo, V. M., Cottrell, J. A., Evans, J. A., Hughes, T. J. R., Lipton, S., Scott, M. A., and Sederberg, T. W. (2010). Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5–8):229–263.

Bazilevs, Y. and Hughes, T. J. R. (2007). Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36(1):12–26.

Beilina, L., Korotov, S., and Krizek, M. (2005). Nonobtuse Tetrahedral Partitions that Refine Locally Towards Fichera-Like Corners. *Applications of Mathematics*, 50(6):569–581.

Belytschko, T. (2000). *Nonlinear Finite Elements for Continua and Structures*. Wiley, Chichester, New York.

Belytschko, T. and Black, T. (1999). Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620.

Belytschko, T., Fish, J., and Bayliss, A. (1990). The spectral overlay on finite elements for problems with high gradients. *Computer Methods in Applied Mechanics and Engineering*, 81(1):71–89.

Belytschko, T., Gracie, R., and Ventura, G. (2009). A review of extended/generalized finite element methods for material modeling. *Modelling and Simulation in Materials Science and Engineering*, 17(4):1–24.

Belytschko, T. and Lu, Y. Y. (1992). A curvilinear spectral overlay method for high gradient problems. *Computer Methods in Applied Mechanics and Engineering*, 95(3):383–396.

Bernardi, C., Fiétier, N., and Owens, R. G. (2001). An error indicator for mortar element solutions to the Stokes problem. *IMA Journal of Numerical Analysis*, 21(4):857–886.

Bernardi, C. and Girault, V. (1998). A Local Regularization Operator for Triangular and Quadrilateral Finite Elements. *SIAM Journal on Numerical Analysis*, 35(5):1893–1916.

Betz, W., Papaioannou, I., and Straub, D. (2014). Numerical methods for the discretization of random fields by means of the Karhunen–Loève expansion. *Computer Methods in Applied Mechanics and Engineering*, 271:109–129.

Bey, K. S., Patra, A., and Oden, J. T. (1995). Hp-version discontinuous Galerkin methods for hyperbolic conservation laws: A parallel adaptive strategy. *International Journal for Numerical Methods in Engineering*, 38(22):3889–3908.

Bischoff, M., Bletzinger, K.-U., Wall, W. A., and Ramm, E. (2004). Models and Finite Elements for Thin-Walled Structures. In *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd, Chichester, West Sussex, England.

Bog, T., Mongeau, A., Zander, N., Kollmannsberger, S., and Rank, E. (2017). Weak imposition of contact constraints on automatically recovered high-order, embedded interfaces using the finite cell method. Submitted to Computational Mechanics.

Bog, T., Zander, N., Kollmannsberger, S., and Rank, E. (2015). Normal contact with high order finite elements and a fictitious contact material. *Computers & Mathematics with Applications*, 70(7):1370–1390.

Bonet, J. and Wood, R. D. (2008). *Nonlinear Continuum Mechanics For Finite Element Analysis*. Cambrigde University Press, Cambrigde, 2nd edition.

Braess, D. (2007). *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, London, 3rd edition.

Brenner, S. C. and Scott, L. R. (2008). *The Mathematical Theory of Finite Element Methods*. Number 15 in Texts in applied mathematics. Springer, New York, USA, 3rd edition.

Brooks, A. N. and Hughes, T. J. R. (1982). 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*, 32(1–3):199–259.

Bueno, J., Starodumov, I., Gomez, H., Galenko, P., and Alexandrov, D. (2016). Three dimensional structures predicted by the modified phase field crystal equation. *Computational Materials Science*, 111:310–312.

Byfut, A. and Schröder, A. (2012). Hp-adaptive extended finite element method. *International Journal for Numerical Methods in Engineering*, 89(11):1392–1418.

Cai, Q. (2013). *Finite Cell Method for Transport Problems in Porous Media*. PhD thesis, Technische Universität München, Munich.

Cai, S., Zhang, W., Zhu, J., and Gao, T. (2014). Stress constrained shape and topology optimization with fixed mesh: A B-spline finite cell method combined with level set function. *Computer Methods in Applied Mechanics and Engineering*, 278:361–387.

Cantwell, C. D., Moxey, D., Comerford, A., Bolis, A., Rocco, G., Mengaldo, G., De Grazia, D., Yakovlev, S., Lombard, J. E., Ekelschot, D., Jordi, B., Xu, H., Mohamied, Y., Eskilsson, C., Nelson, B., Vos, P., Biotto, C., Kirby, R. M., and Sherwin, S. J. (2015). Nektar++: An open-source spectral/hp element framework. *Computer Physics Communications*, 192:205–219.

Cantwell, C. D., Yakovlev, S., Kirby, R. M., Peters, N. S., and Sherwin, S. J. (2014). High-order spectral/hp element discretisation for reaction–diffusion problems on surfaces: Application to cardiac electrophysiology. *Journal of Computational Physics*, 257, Part A:813–829.

Carstensen, C. (2006). Clément Interpolation and Its Role in Adaptive Finite Element Error Control. In Koelink, E., van Neerven, J., de Pagter, B., Sweers, G., Luger, A., and Woracek, H., editors, *Partial Differential Equations and Functional Analysis*, number 168 in Operator Theory: Advances and Applications, pages 27–43. Birkhäuser Basel.

Cheng, K. W. and Fries, T.-P. (2009). Higher-order XFEM for curved strong and weak discontinuities. *International Journal for Numerical Methods in Engineering*, 82(5):564–590.

Cheng, K.-W. and Fries, T.-P. (2012). XFEM with hanging nodes for two-phase incompressible flow. *Computer Methods in Applied Mechanics and Engineering*, 245-246:290–312.

Chessa, J., Smolinski, P., and Belytschko, T. (2002). The extended finite element method (XFEM) for solidification problems. *International Journal for Numerical Methods in Engineering*, 53(8):1959–1977.

Cho, Y.-S. and Im, S. (2006). MLS-based variable-node elements compatible with quadratic interpolation. Part I: Formulation and application for non-matching meshes. *International Journal for Numerical Methods in Engineering*, 65(4):494–516.

Choi, C.-K. and Lee, N.-H. (1996). A 3-D Adaptive Mesh Refinement Using Variable-Node Solid Transition Elements. *International Journal for Numerical Methods in Engineering*, 39(9):1585–1606.

Ciarlet, P. G. (2002). *The Finite Element Method for Elliptic Problems*. Number 40 in Classics in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Clément, P. (1975). Approximation by finite element functions using local regularization. *Revue francaise d'automatique, informatique, recherche operationnelle. Analyse numerique*, 9(2):77–84.

Cottrell, J., Hughes, T. J., and Bazilevs, Y. (2009). *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley and Sons, New York.

Cottrell, J. A., Hughes, T. J. R., and Reali, A. (2007). Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196(41–44):4160–4183.

Crisfield, M. A. and Alfano, G. (2002). Adaptive hierarchical enrichment for delamination fracture using a decohesive zone model. *International Journal for Numerical Methods in Engineering*, 54(9):1369–1390.

D'Angella, D. (2015). *A Posteriori Error Estimator for the Multi-Level Hp-Finite Element Method*. Master's thesis, Technische Universität München.

D'Angella, D., Zander, N., Kollmannsberger, S., Frischmann, F., Schröder, A., Rank, E., and Reali, A. (2016). Multi-Level hp-Adaptivity and Explicit Error Estimation. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):33.

Dauge, M., Düster, A., and Rank, E. (2015). Theoretical and Numerical Investigation of the Finite Cell Method. *Journal of Scientific Computing*, 65(3):1039–1064.

de Borst, R. and Remmers, J. J. C. (2006). Computational modelling of delamination. *Composites Science and Technology*, 66(6):713–722.

de Borst, R., Remmers, J. J. C., and Needleman, A. (2004). Computational aspects of cohesive zone models. In *Advanced Fracture Mechanics for Life and Safety Assesments*, Stockholm.

De Lorenzis, L., Wriggers, P., and Hughes, T. J. (2014). Isogeometric contact: A review. *GAMM-Mitteilungen*, 37(1):85–123.

Demkowicz, L. (2007). *Computing with Hp-Adaptive Finite Elements, Vol. 1: One and Two Dimensional Elliptic and Maxwell Problems*. Applied mathematics and nonlinear science series. Chapman & Hall/CRC, Boca Raton.

Demkowicz, L., Bajer, A., Rachowicz, W., and Gerdes, K. (1999). 3D hp-Adaptive Finite Element Package Fortran 90 Implementation (3Dhp90). TICAM Report 99-29, The University of Texas at Austin, Texas Institute for Computational and Applied Mathematics.

Demkowicz, L., Gerdes, K., Schwab, C., Bajer, A., and Walsh, T. (1998). HP90: A general and flexible Fortran 90 hp-FE code. *Computing and Visualization in Science*, 1(3):145–163.

Demkowicz, L., Oden, J. T., Rachowicz, W., and Hardy, O. (1989). Toward a universal h-p adaptive finite element strategy, part 1. Constrained approximation and data structure. *Computer Methods in Applied Mechanics and Engineering*, 77(1–2):79–112.

Demkowicz, L., Rachowicz, W., and Devloo, P. (2002). A Fully Automatic hp-Adaptivity. *Journal of Scientific Computing*, 17(1-4):117–142.

Demkowicz, L. and Vardapetyan, L. (1998). Modeling of electromagnetic absorption/scattering problems using hp-adaptive finite elements. *Computer Methods in Applied Mechanics and Engineering*, 152(1-2):103–124.

Di Stolfo, P., Schröder, A., Zander, N., and Kollmannsberger, S. (2016). An easy treatment of hanging nodes in hp-finite elements. *Finite Elements in Analysis and Design*, 121:101–117.

Dimitri, R., Lorenzis, L. D., Wriggers, P., and Zavarise, G. (2014). NURBS- and T-spline-based isogeometric cohesive zone modeling of interface debonding. *Computational Mechanics*, 54(2):369–388.

Donéa, J. and Huerta, A. (2003). *Finite Element Methods for Flow Problems*. Wiley, Chichester, New York.

Dréau, K., Chevaugeon, N., and Moës, N. (2010). Studied X-FEM enrichment to handle material interfaces with higher order finite element. *Computer Methods in Applied Mechanics and Engineering*, 199(29-32):1922–1936.

Dubcova, L., Šolín, P., Cerveny, J., and Kus, P. (2010). Space and Time Adaptive Two-Mesh hp-Finite Element Method for Transient Microwave Heating Problems. *Electromagnetics*, 30(1-2):23–40.

Duczek, S., Berger, H., and Gabbert, U. (2015). The Finite Pore Method: A new approach to evaluate gas pores in cast parts by combining computed tomography and the finite cell method. *International Journal of Cast Metals Research*, 28(4):221–228.

Duczek, S. and Gabbert, U. (2016). The finite cell method for polygonal meshes: Poly-FCM. *Computational Mechanics*, pages 1–32.

Duczek, S., Joulaian, M., Düster, A., and Gabbert, U. (2014). Numerical analysis of Lamb waves using the finite and spectral cell methods. *International Journal for Numerical Methods in Engineering*, 99(1):26–53.

Dugdale, D. S. (1960). Yielding of steel sheets containing slits. *Journal of the Mechanics and Physics of Solids*, 8(2):100–104.

Düster, A. (2002). *High Order Finite Elements for Three-Dimensional, Thin Walled Nonlinear Continua*. PhD thesis, Lehrstuhl für Bauinformatik, Technische Universität München, Munich.

Düster, A., Bröker, H., and 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*, 52(7):673–703.

Düster, A., Niggl, A., and Rank, E. (2007a). Applying the hp–d version of the FEM to locally enhance dimensionally reduced models. *Computer Methods in Applied Mechanics and Engineering*, 196(37-40):3524–3533.

Düster, A., Parvizian, J., Yang, Z., and Rank, E. (2008). The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197(45–48):3768–3782.

Düster, A., Scholz, D., and Rank, E. (2007b). Pq-Adaptive solid finite elements for three-dimensional plates and shells. *Computer Methods in Applied Mechanics and Engineering*, 197(1–4):243–254.

Eibner, T. and Melenk, J. M. (2006). An adaptive strategy for hp-FEM based on testing for analyticity. *Computational Mechanics*, 39(5):575–595.

Elhaddad, M., Zander, N., Kollmannsberger, S., Shadavakhsh, A., Nübel, V., and Rank, E. (2015). Finite Cell Method: High-Order Structural Dynamics for Complex Geometries. *International Journal of Structural Stability and Dynamics*, 15(7):1540018.

Elsevier (2016). Elsevier journal publishing agreement. `https://www.elsevier.com/about/company-information/policies/copyright`.

Embar, A., Dolbow, J., and Harari, I. (2010). Imposing Dirichlet boundary conditions with Nitsche's method and spline-based finite elements. *International Journal for Numerical Methods in Engineering*, 83(7):877–898.

Felippa, C. A. (2004). Fluid-Structure Interaction. Lecture notes, Department of Aerospace Engineering Sciences, University of Colorado at Boulder, Boulder, Colorado, USA.

Felippa, C. A. (2013). FEM Convergence Requirements. Lecture Notes: Introduction to Finite Element Methods Chapter 19, Department of Aerospace Engineering Sciences, University of Colorado at Boulder, Boulder, Colorado, USA.

Fernández-Méndez, S. and Huerta, A. (2004). Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 193(12-14):1257–1275.

Fischer, K. A. and Wriggers, P. (2005). Frictionless 2D Contact formulations for finite deformations based on the mortar method. *Computational Mechanics*, 36(3):226–244.

Fish, J. (1992a). Hierarchical modelling of discontinuous fields. *Communications in Applied Numerical Methods*, 8(7):443–453.

Fish, J. (1992b). The s-version of the finite element method. *Computers & Structures*, 43(3):539–547.

Fish, J. and Guttal, R. (1996). The s-version of Finite Element Method for laminated composites. *International Journal for Numerical Methods in Engineering*, 39(21):3641–3662.

Fish, J. and Markolefas, S. (1993). Adaptive s-method for linear elastostatics. *Computer Methods in Applied Mechanics and Engineering*, 104(3):363–396.

Fish, J., Markolefas, S., Guttal, R., and Nayak, P. (1994). On adaptive multilevel superposition of finite element meshes for linear elastostatics. *Applied Numerical Mathematics*, 14(1–3):135–164.

Franke, D. (2011). *Investigation of Mechanical Contact Problems with High-Order Finite Element Methods*. PhD thesis, Technische Universität München.

Frauenfelder, P. (2004). *Hp-Finite Element Methods on Anisotropically, Locally Refined Meshes in Three Dimensions with Stochastic Data*. PhD thesis, Swiss Federal Institute of Technology Zürich, Zürich.

Frauenfelder, P. and Lage, C. (2002). Concepts—An Object-Oriented Software Package for Partial Differential Equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 36(5):937–951.

Fries, T.-P. (2008). A corrected XFEM approximation without problems in blending elements. *International Journal for Numerical Methods in Engineering*, 75(5):503–532.

Fries, T.-P. and Belytschko, T. (2010). The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304.

Fries, T.-P., Byfut, A., Alizada, A., Cheng, K. W., and Schröder, A. (2011). Hanging nodes and XFEM. *International Journal for Numerical Methods in Engineering*, 86(4-5):404–430.

Fries, T.-P. and Omerović, S. (2015). Higher-order accurate integration of implicit geometries. *International Journal for Numerical Methods in Engineering*, 106(5):323–371.

Gerstenberger, A. and Wall, W. A. (2008). An eXtended Finite Element Method/Lagrange multiplier based approach for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 197(19–20):1699–1714.

Gerstenberger, A. and Wall, W. A. (2010). An embedded Dirichlet formulation for 3D continua. *International Journal for Numerical Methods in Engineering*, 82(5):537–563.

Geubelle, P. H. and Baylor, J. S. (1998). Impact-induced delamination of composites: A 2D simulation. *Composites Part B: Engineering*, 29(5):589–602.

Girkmann, K. (1956). *Flächentragwerke*. Springer, Wien, 4th edition.

Gomez, H. and Nogueira, X. (2012). An unconditionally energy-stable method for the phase field crystal equation. *Computer Methods in Applied Mechanics and Engineering*, 249–252:52–61.

Gonzalez-Ferreiro, B., Gomez, H., and Romero, I. (2014). A thermodynamically consistent numerical method for a phase field model of solidification. *Communications in Nonlinear Science and Numerical Simulation*, 19(7):2309–2323.

Gu, L. and Belytschko, T. (1994). A numerical study of stress singularities in a two-material wedge. *International Journal of Solids and Structures*, 31(6):865–889.

Gui, W. and Babuška, I. (1986a). The h, p and h-p versions of the finite element method in 1 dimension Part I: The error analysis of the p-version. *Numerische Mathematik*, 49(6):577–612.

Gui, W. and Babuška, I. (1986b). The h, p and h-p versions of the finite element method in 1 dimension Part II: The error analysis of the h-and h-p versions. *Numerische Mathematik*, 49(6):613–657.

Gui, W. and Babuška, I. (1986c). The h, p and h-p versions of the finite element method in 1 dimension Part III: The Adaptive h-p Version. *Numerische Mathematik*, 49(6):659–683.

Guiamatsia, I., Ankersen, J. K., Davies, G. A. O., and Iannucci, L. (2009). Decohesion finite element with enriched basis functions for delamination. *Composites Science and Technology*, 69(15–16):2616–2624.

Guo, B. and Babuška, I. (1986a). The h-p version of the finite element method Part 1: The basic approximation results. *Computational Mechanics*, 1(1):21–41.

Guo, B. and Babuška, I. (1986b). The h-p version of the finite element method Part 2: General results and applications. *Computational Mechanics*, 1(3):203–220.

Guo, Y. and Ruess, M. (2015a). A layerwise isogeometric approach for NURBS-derived laminate composite shells. *Composite Structures*, 124:300–309.

Guo, Y. and Ruess, M. (2015b). Nitsche's method for a coupling of isogeometric thin shells and blended shell structures. *Computer Methods in Applied Mechanics and Engineering*, 284:881–905.

Gupta, A. K. (1978). A finite element for transition from a fine to a coarse grid. *International Journal for Numerical Methods in Engineering*, 12(1):35–45.

Hager, W. (1984). Condition Estimates. *SIAM Journal on Scientific and Statistical Computing*, 5(2):311–316.

Hansbo, P., Lovadina, C., Perugia, I., and Sangalli, G. (2005). A Lagrange multiplier method for the finite element solution of elliptic interface problems using non-matching meshes. *Numerische Mathematik*, 100(1):91–115.

Harper, P. W. and Hallett, S. R. (2008). Cohesive zone length in numerical simulations of composite delamination. *Engineering Fracture Mechanics*, 75(16):4774–4792.

Heinze, S., Joulaian, M., and Düster, A. (2015). Numerical homogenization of hybrid metal foams using the finite cell method. *Computers & Mathematics with Applications*, 70(7):1501–1517.

Hermes (2016). Hermes - Higher-Order Modular Finite Element System. User's guide, University of Reno, Nevada, USA.

Hermes, F. H. (2010). *Process Zone and Cohesive Element Size in Numerical Simulations of Delamination in Bi-Layers*. Master's thesis, Technische Universiteit Eindhoven, Eindhoven.

Hertz, H. (1882). Ueber die Berührung fester elastischer Körper. *Journal für die reine und angewandte Mathematik*, 92:156–171.

Hillerborg, A., Modéer, M., and Petersson, P. E. (1976). Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements. *Cement and Concrete Research*, 6(6):773–781.

Höllig, K., Reif, U., and Wipper, J. (2001). Weighted extended B-spline approximation of Dirichlet problems. *SIAM Journal on Numerical Analysis*, 39(2):442–462.

Holzapfel, G. A. (2000). *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Wiley, Chichester, New York.

Hosseini, S. (2014). *Numerical Simulation of Damage Mechanisms in Composite Materials*. PhD thesis, Technische Universiteit Eindhoven, Eindhoven.

Hosseini, S., Remmers, J. J., Verhoosel, C. V., and de Borst, R. (2015). Propagation of delamination in composite materials with isogeometric continuum shell elements. *International Journal for Numerical Methods in Engineering*, 102(3-4):159–179.

Houston, P., Senior, B., and Süli, E. (2003). Sobolev regularity estimation for hp-adaptive finite element methods. In Brezzi, F., Buffa, A., Corsaro, S., and Murli, A., editors, *Numerical Mathematics and Advanced Applications*, pages 631–656. Springer Milan.

Hsu, L.-C. and Mavriplis, C. (1998). Adaptive Meshes for the Spectral Element Method. In *Ninth International Conference on Domain Decomposition Methods*, pages 375–381.

Hughes, T. J. R. (2000). *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Mineola, NY.

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

IEEE Computer Society, Microprocessor Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board (2008). *IEEE Standard for Floating-Point Arithmetic*. Institute of Electrical and Electronics Engineers, New York, NY.

Johnson, C. (2009). *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover Publications, Mineola, N.Y.

Jomo, J., Zander, N., Elhaddad, M., Özcan, A. I., Kollmannsberger, S., Mundani, R.-P., and Rank, E. (2017). Parallelization of the multi-level hp-adaptive finite cell method. *Computers and Mathematics with Applications*, in press.

Joulaian, M., Duczek, S., Gabbert, U., and Düster, A. (2014). Finite and spectral cell method for wave propagation in heterogeneous materials. *Computational Mechanics*, 54(3):661–675.

Joulaian, M. and Düster, A. (2013). Local enrichment of the finite cell method for problems with material interfaces. *Computational Mechanics*, 52(4):741–762.

Joulaian, M., Hubrich, S., and Düster, A. (2016). Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Computational Mechanics*, pages 1–21.

Kamensky, D., Hsu, M.-C., Schillinger, D., Evans, J. A., Aggarwal, A., Bazilevs, Y., Sacks, M. S., and Hughes, T. J. R. (2015). An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053.

Karniadakis, G. and Sherwin, S. (2013). *Spectral/HP Element Methods for Computational Fluid Dynamics*. Oxford University Press, Oxford, 2nd edition.

Kelly, D. W., Gago, D. S., Zienkiewicz, O. C., and Babuska, I. (1983). A posteriori error analysis and adaptive processes in the finite element method: Part I—error analysis. *International Journal for Numerical Methods in Engineering*, 19(11):1593–1619.

Khoei, A. R., Moslemi, H., and Sharifi, M. (2012). Three-dimensional cohesive fracture modeling of non-planar crack growth using adaptive FE technique. *International Journal of Solids and Structures*, 49(17):2334–2348.

Kiendl, J., Bazilevs, Y., Hsu, M. C., Wüchner, R., and Bletzinger, K. U. (2010). The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199(37–40):2403–2416.

Kim, J. H., Lim, J. H., Lee, J. H., and Im, S. (2008). A new computational approach to contact mechanics using variable-node finite elements. *International Journal for Numerical Methods in Engineering*, 73(13):1966–1988.

Kim, Y. H., Levit, I., and Stanley, G. (1991). A finite element adaptive mesh refinement technique that avoids multipoint constraints and transition zones. In Parsons, I. and Nour-Omid, B., editors, *Iterative Equation Solvers for Structural Mechanics Problems*, volume 4, pages 27–35. ASME, New York.

Királyfalvi, G. and Szabó, B. A. (1997). Quasi-regional mapping for the p-version of the finite element method. *Finite elements in analysis and design*, 27(1):85–97.

Kollmannsberger, S., Özcan, A. I., and Rank, E. (2016). Computational modeling of the SLM process: A hierarchical treatment of the transient heat equation with phase changes. in preparation.

Konyukhov, A. and Izi, R. (2015). *Introduction to Computational Contact Mechanics: A Geometrical Approach*. Wiley series in computational mechanics. John Wiley & Sons, Chichester, West Sussex.

Kopp, P. (2014). *Implementation of the Characteristic Based Split Algorithm for the Shallow Water Equations in a High Order Finite Element Framework*. Bachelor's thesis, Technische Universität München, Munich.

Kopp, P. (2017). *Multi-Level Hp-FEM and the Finite Cell Method for the Navier-Stokes Equations Using a Variational Multiscale Formulation*. Master's thesis, Technische Universität München, Munich.

Korous, L. and Šolín, P. (2012). An adaptive hp-DG method with dynamically-changing meshes for non-stationary compressible Euler equations. *Computing*, 95(1):425–444.

Korshunova, N. (2016). *Partitioned Hp-D Approach for Multiscale Transient Heat Problems*. Master's thesis, Technische Universität München, Munich.

Kossaczký, I. (1994). A recursive approach to local mesh refinement in two and three dimensions. *Journal of Computational and Applied Mathematics*, 55(3):275–288.

Krantz, S. G. and Parks, H. R. (2002). *A Primer of Real Analytic Functions*. Birkhäuser, Boston, 2nd edition.

Krause, R. and Rank, E. (2003). Multiscale computations with a combination of the h- and p-versions of the finite-element method. *Computer Methods in Applied Mechanics and Engineering*, 192(35-36):3959–3983.

Kudela, L. (2013). *Highly Accurate Subcell Integration in the Context of The Finite Cell Method*. Master's thesis, Technische Universität München, Chair for Computation in Engineering.

Kudela, L., Zander, N., Bog, T., Kollmannsberger, S., and Rank, E. (2015). Efficient and accurate numerical quadrature for immersed boundary methods. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):1–22.

Kudela, L., Zander, N., Kollmannsberger, S., and Rank, E. (2016). Smart octrees: Accurately integrating discontinuous functions in 3D. *Computer Methods in Applied Mechanics and Engineering*, 306:406–426.

Kus, P. (2011). *Automatic Hp-Adaptivity on Meshes with Arbitrary-Level Hanging Nodes in 3D*. PhD thesis, Charles University, Institute of Mathematics, Prague.

Kus, P., Šolín, P., and Andrs, D. (2014). Arbitrary-level hanging nodes for adaptive -FEM approximations in 3D. *Journal of Computational and Applied Mathematics*, 270:121–133.

Laursen, T. A. (2003). *Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*. Springer, Berlin.

Ledentsov, D. (2010). *Model Adaptivity in Sheet Metal Forming Simulation*. PhD thesis, Technische Universität München, Munich.

Ledentsov, D., Düster, A., Volk, W., Wagner, M., Heinle, I., and Rank, E. (2010). Model adaptivity for industrial application of sheet metal forming simulation. *Finite Elements in Analysis and Design*, 46(7):585–600.

Legrain, G. (2013). A NURBS enhanced extended finite element approach for unfitted CAD analysis. *Computational Mechanics*, 52(4):913–929.

Legrain, G., Chevaugeon, N., and Dréau, K. (2012). High order X-FEM and levelsets for complex microstructures: Uncoupling geometry and approximation. *Computer Methods in Applied Mechanics and Engineering*, 241-244:172–189.

Lim, J. H., Sohn, D., and Im, S. (2012). Variable-node element families for mesh connection and adaptive mesh computation. *Structural Engineering and Mechanics*, 43(3):349–370.

Loehnert, S., Mueller-Hoeppe, D. S., and Wriggers, P. (2011). 3D corrected XFEM approach and extension to finite deformation theory. *International Journal for Numerical Methods in Engineering*, 86(4-5):431–452.

Malvern, L. E. (1977). *Introduction to the Mechanics of a Continuous Medium*. Prentice Hall, Englewood Cliffs.

Mann, A. (2011). Cracks in steel structures. *Proceedings of the ICE - Forensic Engineering*, 164(1):15–23.

Mäntylä, M. (1988). *An Introduction to Solid Modeling*. Number 13 in Principles of computer science series. Computer Science Press, Rockville.

Marsden, J. E. and Hughes, T. J. R. (1994). *Mathematical Foundations of Elasticity*. Dover Publications, New York, revised edition.

Mavriplis, C. (1994). Adaptive mesh strategies for the spectral element method. *Computer Methods in Applied Mechanics and Engineering*, 116(1):77–86.

Melenk, J. M. and Babuška, I. (1996). The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1):289–314.

Melenk, J. M., Gerdes, K., and Schwab, C. (2001). Fully discrete hp-finite elements: Fast quadrature. *Computer Methods in Applied Mechanics and Engineering*, 190(32–33):4339–4364.

Melenk, J. M. and Wohlmuth, B. I. (2001). On residual-based a posteriori error estimation in hp-FEM. *Advances in Computational Mathematics*, 15(1-4):311–331.

Mi, Y., Crisfield, M. A., Davies, G. A. O., and Hellweg, H. B. (1998). Progressive Delamination Using Interface Elements. *Journal of Composite Materials*, 32(14):1246–1272.

Miehe, C., Welschinger, F., and Hofacker, M. (2010). Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations. *International Journal for Numerical Methods in Engineering*, 83(10):1273–1311.

Mitchell, W. F. (1988). *Unified Multilevel Adaptive Finite Element Methods for Elliptic Problems*. PhD thesis, University of Illinois at Urbana-Champaign.

Mitchell, W. F. (1989). A Comparison of Adaptive Refinement Techniques for Elliptic Problems. *ACM Transactions on Mathematical Software*, 15(4):326–347.

Mitchell, W. F. (1991). Adaptive refinement for arbitrary finite-element spaces with hierarchical bases. *Journal of Computational and Applied Mathematics*, 36(1):65–78.

Mitchell, W. F. (1992). Optimal Multilevel Iterative Methods for Adaptive Grids. *SIAM Journal on Scientific and Statistical Computing*, 13(1):146–167.

Mitchell, W. F. and McClain, M. A. (2011a). A comparison of hp-adaptive strategies for elliptic partial differential equations. Technical Report NISTIR 7824, National Institute of Standards and Technology, Gaithersburg.

Mitchell, W. F. and McClain, M. A. (2011b). A Survey of hp-Adaptive Strategies for Elliptic Partial Differential Equations. In Simos, T. E., editor, *Recent Advances in Computational and Applied Mathematics*, pages 227–258. Springer, Dordrecht, Netherlands.

Mitchell, W. F. and McClain, M. A. (2014). A Comparison of hp-Adaptive Strategies for Elliptic Partial Differential Equations. *ACM Transactions on Mathematical Software*, 41(1):1–39.

Moës, N. and Belytschko, T. (2002). Extended finite element method for cohesive crack growth. *Engineering Fracture Mechanics*, 69(7):813–833.

Moës, N., Dolbow, J., and Belytschko, T. (1999). A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150.

Moore, P. K. and Flaherty, J. E. (1992). Adaptive local overlapping grid methods for parabolic systems in two space dimensions. *Journal of Computational Physics*, 98(1):54–63.

Mote, C. D. (1971). Global-local finite element. *International Journal for Numerical Methods in Engineering*, 3(4):565–574.

Needleman, A. (1987). A Continuum Model for Void Nucleation by Inclusion Debonding. *Journal of Applied Mechanics*, 54(3):525–531.

Needleman, A. (1990a). An analysis of decohesion along an imperfect interface. *International Journal of Fracture*, 42(1):21–40.

Needleman, A. (1990b). An analysis of tensile decohesion along an interface. *Journal of the Mechanics and Physics of Solids*, 38(3):289–324.

Newmark, N. M. (1959). A Method of Computation for Structural Dynamics. *Journal of the Engineering Mechanics Division*, 85(EM3):67–94.

Nguyen, V. P., Kerfriden, P., and Bordas, S. P. A. (2014). Two- and three-dimensional isogeometric cohesive elements for composite delamination analysis. *Composites Part B: Engineering*, 60:193–212.

Nguyen, V. P. and Nguyen-Xuan, H. (2013). High-order B-splines based finite elements for delamination analysis of laminated composites. *Composite Structures*, 102:261–275.

Nitsche, J. (1971). Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1):9–15.

Noor, A. K. (1986). Global-local methodologies and their application to nonlinear analysis. *Finite Elements in Analysis and Design*, 2(4):333–346.

Oden, J. T. (1969). A general theory of finite elements. I. Topological considerations. *International Journal for Numerical Methods in Engineering*, 1(2):205–221.

Oden, J. T. and Patra, A. (1995). A parallel adaptive strategy for hp finite element computations. *Computer Methods in Applied Mechanics and Engineering*, 121(1):449–470.

Pandolfi, A., Guduru, P. R., Ortiz, M., and Rosakis, A. J. (2000). Three dimensional cohesive-element analysis and experiments of dynamic fracture in C300 steel. *International Journal of Solids and Structures*, 37(27):3733–3760.

Park, K., Paulino, G. H., Celes, W., and Espinha, R. (2012). Adaptive mesh refinement and coarsening for cohesive zone modeling of dynamic fracture. *International Journal for Numerical Methods in Engineering*, 92(1):1–35.

Parvizian, J., Düster, A., and Rank, E. (2007). Finite cell method. *Computational Mechanics*, 41(1):121–133.

Parvizian, J., Düster, A., and Rank, E. (2011). Topology optimization using the finite cell method. *Optimization and Engineering*, 13(1):57–78.

Paszyński, M. and Demkowicz, L. (2006). Parallel, fully automatic hp-adaptive 3D finite element package. *Engineering with Computers*, 22(3-4):255–276.

Payette, G. S. (2012). Spectral/Hp Finite Element Models for Fluids and Structures. Technical Report SAND2012-7615, Sandia National Laboratories.

Planas, J., Elices, M., Guinea, G. V., Gómez, F. J., Cendón, D. A., and Arbilla, I. (2003). Generalizations and specializations of cohesive crack models. *Engineering Fracture Mechanics*, 70(14):1759–1776.

Quarteroni, A. and Valli, A. (1994). *Numerical Approximation of Partial Differential Equations*, volume 23 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg.

Rachowicz, W. and Demkowicz, L. (2000a). An hp-adaptive finite element method for electromagnetics: Part 1: Data structure and constrained approximation. *Computer Methods in Applied Mechanics and Engineering*, 187(1–2):307–335.

Rachowicz, W. and Demkowicz, L. (2000b). An hp-adaptive finite element method for electromagnetics: Part 1: Data structure and constrained approximation. *Computer Methods in Applied Mechanics and Engineering*, 187(1–2):307–335.

Rachowicz, W. and Demkowicz, L. (2002). An hp-adaptive finite element method for electromagnetics—part II: A 3D implementation. *International journal for numerical methods in engineering*, 53(1):147–180.

Rachowicz, W., Oden, J. T., and Demkowicz, L. (1989). Toward a universal h-p adaptive finite element strategy part 3. design of h-p meshes. *Computer Methods in Applied Mechanics and Engineering*, 77(1–2):181–212.

Rachowicz, W., Pardo, D., and Demkowicz, L. (2006). Fully automatic hp-adaptivity in three dimensions. *Computer Methods in Applied Mechanics and Engineering*, 195(37–40):4816–4842.

Rank, E. (1992). Adaptive remeshing and h-p domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 101(1–3):299–313.

Rank, E. and Babuška, I. (1987). An expert system for the optimal mesh design in the hp-version of the finite element method. *International Journal for Numerical Methods in Engineering*, 24(11):2087–2106.

Rank, E., Düster, A., Nübel, V., Preusch, K., and Bruhns, O. (2005). High order finite elements for shells. *Computer Methods in Applied Mechanics and Engineering*, 194(21-24):2494–2512.

Rank, E., Kollmannsberger, S., Sorger, C., and Düster, A. (2011). Shell Finite Cell Method: A high order fictitious domain approach for thin-walled structures. *Computer Methods in Applied Mechanics and Engineering*, 200(45-46):3200–3209.

Rank, E. and Krause, R. (1997). A multiscale finite-element method. *Computers & structures*, 64(1):139–144.

Rank, E., Ruess, M., Kollmannsberger, S., Schillinger, D., and Düster, A. (2012). Geometric modeling, isogeometric analysis and the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 249-252:104–115.

Rank, E. and Zienkiewicz, O. C. (1987). A simple error estimator in the finite element method. *Communications in Applied Numerical Methods*, 3(3):243–249.

Reddy, B. D. (1997). *Introductory Functional Analysis: With Applications to Boundary Value Problems and Finite Elements*. Number 27 in Texts in applied mathematics. Springer, New York, 1998 edition.

Rheinboldt, W. C. and Mesztenyi, C. K. (1980). On a Data Structure for Adaptive Finite Element Mesh Refinements. *ACM Transactions on Mathematical Software*, 6(2):166–187.

Rivara, M.-C. (1984a). Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *International Journal for Numerical Methods in Engineering*, 20(4):745–756.

Rivara, M.-C. (1984b). Design and Data Structure of Fully Adaptive, Multigrid, Finite-element Software. *ACM Transactions on Mathematical Software*, 10(3):242–264.

Rivara, M.-C. (1984c). Mesh Refinement Processes Based on the Generalized Bisection of Simplices. *SIAM Journal on Numerical Analysis*, 21(3):604–613.

Ruess, M., Schillinger, D., Bazilevs, Y., Varduhn, V., and Rank, E. (2013). Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method. *International Journal for Numerical Methods in Engineering*, 95(10):811–846.

Ruess, M., Schillinger, D., Özcan, A. I., and Rank, E. (2014). Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269:46–71.

Ruess, M., Tal, D., Trabelsi, N., Yosibash, Z., and Rank, E. (2012). The finite cell method for bone simulations: Verification and validation. *Biomechanics and modeling in mechanobiology*, 11(3-4):425–37.

Ruiz, G., Pandolfi, A., and Ortiz, M. (2001). Three-dimensional cohesive modeling of dynamic mixed-mode fracture. *International Journal for Numerical Methods in Engineering*, 52(1-2):97–120.

Sadd, M. H. (2009). *Elasticity Theory, Applications and Numerics*. Elsevier Butterworth-Heinemann, Burlington.

Samimi, M., van Dommelen, J. A. W., and Geers, M. G. D. (2009). An enriched cohesive zone model for delamination in brittle interfaces. *International Journal for Numerical Methods in Engineering*, 80(5):609–630.

Samimi, M., van Dommelen, J. A. W., and Geers, M. G. D. (2011a). A self-adaptive finite element approach for simulation of mixed-mode delamination using cohesive zone models. *Engineering Fracture Mechanics*, 78(10):2202–2219.

Samimi, M., van Dommelen, J. A. W., and Geers, M. G. D. (2011b). A three-dimensional self-adaptive cohesive zone model for interfacial delamination. *Computer Methods in Applied Mechanics and Engineering*, 200(49–52):3540–3553.

Sanches, R., Bornemann, P., and Cirak, F. (2011). Immersed b-spline (i-spline) finite element method for geometrically complex domains. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):1432–1445.

Schellekens, J. C. J. and de Borst, R. (1993a). A non-linear finite element approach for the analysis of mode-I free edge delamination in composites. *International Journal of Solids and Structures*, 30(9):1239–1253.

Schellekens, J. C. J. and de Borst, R. (1993b). On the numerical integration of interface elements. *International Journal for Numerical Methods in Engineering*, 36(1):43–66.

Schillinger, D. (2012). *The P- and B-Spline Versions of the Geometrically Nonlinear Finite Cell Method and Hierarchical Refinement Strategies for Adaptive Isogeometric and Embedded Domain Analysis*. PhD thesis, Technische Universität München, Munich.

Schillinger, D., Dedè, L., Scott, M. A., Evans, J. A., Borden, M. J., Rank, E., and Hughes, T. J. (2012a). An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249-252:116–150.

Schillinger, D., Düster, A., and Rank, E. (2012b). The hp-d-adaptive finite cell method for geometrically nonlinear problems of solid mechanics. *International Journal for Numerical Methods in Engineering*, 89(9):1171–1202.

Schillinger, D., Evans, J. A., Reali, A., Scott, M. A., and Hughes, T. J. R. (2013). Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Computer Methods in Applied Mechanics and Engineering*, 267:170–232.

Schillinger, D., Hossain, S. J., and Hughes, T. J. R. (2014). Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 277:1–45.

Schillinger, D. and Rank, E. (2011). 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*, 200(47-48):3358–3380.

Schillinger, D. and Ruess, M. (2014). The Finite Cell Method: A Review in the Context of Higher-Order Structural Analysis of CAD and Image-Based Geometric Models. *Archives of Computational Methods in Engineering*, 22(3):391–455.

Schillinger, D., Ruess, M., Zander, N., Bazilevs, Y., Düster, A., and Rank, E. (2012c). Small and large deformation analysis with the p- and B-spline versions of the Finite Cell Method. *Computational Mechanics*, 50(4):445–478.

Schmidt, A. and Siebert, K. G. (2000). A posteriori estimators for the h-p version of the finite element method in 1D. *Applied Numerical Mathematics*, 35(1):43–66.

Schneiders, R. (2000). Algorithms for quadrilateral and hexahedral mesh generation. *Proceedings of the VKI Lecture Series on Computational Fluid Dynamic, VKI-LS*, 4.

Schneiders, R. and Debye, J. (1995). Refining Quadrilateral and Brick Element Meshes. In Babuska, I., Henshaw, W. D., Oliger, J. E., Flaherty, J. E., Hopcroft, J. E., and Tezduyar, T., editors, *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, number 75 in The IMA Volumes in Mathematics and its Applications, pages 53–65. Springer New York.

Schnepp, S. M. and Weiland, T. (2012). Efficient large scale electromagnetic simulations using dynamically adapted meshes with the discontinuous Galerkin method. *Journal of Computational and Applied Mathematics*, 236(18):4909–4924.

Scholz, D., Kollmannsberger, S., Düster, A., and Rank, E. (2006). Thin Solids for Fluid-Structure Interaction. In Bungartz, H.-J. and Schäfer, M., editors, *Fluid-Structure Interaction*, number 53 in Lecture Notes in Computational Science and Engineering, pages 294–335. Springer Berlin Heidelberg.

Schröder, A. (2008). Constraints Coefficients in hp-FEM. In Kunisch, K., Of, G., and Steinbach, O., editors, *Numerical Mathematics and Advanced Applications*, pages 183–190. Springer Berlin Heidelberg.

Schröder, A. (2011). Constrained Approximation in hp-FEM: Unsymmetric Subdivisions and Multi-Level Hanging Nodes. In Hesthaven, J. S. and Ronquist, E. M., editors, *Spectral and High Order Methods for Partial Differential Equations*, number 76 in Lecture Notes in Computational Science and Engineering, pages 317–325. Springer Berlin Heidelberg.

Schumann, R. (1989). Regularity for Signorini's problem in linear elasticity. *Manuscripta Mathematica*, 63(3):255–291.

Schwab, C. (1998). *P- and Hp-Finite Element Methods: Theory and Applications in Solid and Fluid Mechanics*. Numerical mathematics and scientific computation. Oxford University Press, Oxford.

Schwab, C. and Suri, M. (1999). Mixed hp finite element methods for Stokes and non-Newtonian flow. *Computer Methods in Applied Mechanics and Engineering*, 175(3–4):217–241.

Sevilla, R., Fernández-Méndez, S., and Huerta, A. (2008). NURBS-enhanced finite element method (NEFEM). *International Journal for Numerical Methods in Engineering*, 76(1):56–83.

Sevilla, R., Fernández-Méndez, S., and Huerta, A. (2011a). 3D NURBS-enhanced finite element method (NEFEM). *International Journal for Numerical Methods in Engineering*, 88(2):103–125.

Sevilla, R., Fernández-Méndez, S., and Huerta, A. (2011b). Comparison of high-order curved finite elements. *International Journal for Numerical Methods in Engineering*, 87(8):719–734.

Sevilla, R., Fernández-Méndez, S., and Huerta, A. (2011c). NURBS-Enhanced Finite Element Method (NEFEM). *Archives of Computational Methods in Engineering*, 18(4):441–484.

Sewell, E. G. (1972). *Automatic Generation of Triangulations for Piecewise Polynomial Approximation*. PhD thesis, Purdue University.

Shewchuk, J. R. (1994). An introduction to the conjugate gradient method without the agonizing pain. Technical report, Carnegie-Mellon University. Department of Computer Science.

Shewchuk, J. R. (1999). Lecture Notes on Delaunay Mesh Generation. Technical report, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720.

Šolín, P. (2004). *Higher-Order Finite Element Methods*. Studies in advanced mathematics. Chapman & Hall/CRC, Boca Raton.

Šolín, P., Červený, J., and Doležel, I. (2008). Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM. *Mathematics and Computers in Simulation*, 77(1):117–132.

Šolín, P. and Demkowicz, L. (2004). Goal-oriented hp-adaptivity for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 193(6-8):449–468.

Šolín, P., Dubcova, L., and Doležel, I. (2010a). Adaptive hp-FEM with arbitrary-level hanging nodes for Maxwell's equations. *Advances in Applied Mathematics and Mechanics*, 2(4):518–532.

Šolín, P., Dubcova, L., and Kruis, J. (2010b). Adaptive -FEM with dynamical meshes for transient heat and moisture transfer problems. *Journal of Computational and Applied Mathematics*, 233(12):3103–3112.

Šolín, P. and Korous, L. (2012a). Adaptive higher-order finite element methods for transient PDE problems based on embedded higher-order implicit Runge–Kutta methods. *Journal of Computational Physics*, 231(4):1635–1649.

Šolín, P. and Korous, L. (2012b). Space-time adaptive hp-FEM for problems with traveling sharp fronts. *Computing*, 95(1):709–722.

Springer (2016). Springer Copyright Transfer Statement. `http://static.springer.com/sgw/documents/1384288/application/pdf/62482_CTS+Format_T1.pdf`.

Stavrev, A. (2012). *The Role of Higher-Order Geometry Approximation and Accurate Quadrature in NURBS Based Immersed Boundary Methods*. Master's thesis, Technische Universität München.

Stavrev, A., Nguyen, L. H., Shen, R., Varduhn, V., Behr, M., Elgeti, S., and Schillinger, D. (2016). Geometrically accurate, efficient, and flexible quadrature techniques for the tetrahedral finite cell method. *Computer Methods in Applied Mechanics and Engineering*, in press.

Strouboulis, T., Babuška, I., and Copps, K. (2000a). The design and analysis of the Generalized Finite Element Method. *Computer Methods in Applied Mechanics and Engineering*, 181(1–3):43–69.

Strouboulis, T., Copps, K., and Babuska, I. (2000b). The generalized finite element method: An example of its implementation and illustration of its performance. *International Journal for Numerical Methods in Engineering*, 47(8):1401–1417.

Strouboulis, T., Copps, K., and Babuska, I. (2001). The generalized finite element method. *Computer methods in applied mechanics and engineering*, 190(32):4081–4193.

Szabó, B. A. (1986). Estimation and Control of Error Based on p Convergence. In Babuska, I., Zienkiewicz, O. C., Gago, J., and de Oliviera, E. R., editors, *Accuracy Estimates and Adaptive Refinements in Finite Element Calculations*, pages 25–55. Wiley, New York.

Szabó, B. A. and Babuška, I. (1991). *Finite Element Analysis*. John Wiley & Sons, New York.

Szabó, B. A., Düster, A., and Rank, E. (2004). The p-version of the finite element method. In Stein, E., editor, *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Chichester, West Sussex.

Thompson, J. F., Soni, B. K., and Weatherill, N. P. (1998). *Handbook of Grid Generation*. CRC Press.

Tomar, V., Zhai, J., and Zhou, M. (2004). Bounds for element size in a variable stiffness cohesive finite element model. *International Journal for Numerical Methods in Engineering*, 61(11):1894–1920.

TUM (2010). Promotionsordnung der Technischen Universität München vom 1. August 2001 in der Fassung der Neunten Änderungssatzung vom 2. August 2010.

Turon, A., Dávila, C. G., Camanho, P. P., and Costa, J. (2007). An engineering solution for mesh size effects in the simulation of delamination using cohesive zone models. *Engineering Fracture Mechanics*, 74(10):1665–1682.

Tvergaard, V. and Hutchinson, J. W. (1992). The relation between crack growth resistance and fracture process parameters in elastic-plastic solids. *Journal of the Mechanics and Physics of Solids*, 40(6):1377–1397.

van den Bosch, M. J., Schreurs, P. J. G., and Geers, M. G. D. (2006). An improved description of the exponential Xu and Needleman cohesive zone law for mixed-mode decohesion. *Engineering Fracture Mechanics*, 73(9):1220–1234.

Varduhn, V., Hsu, M.-C., Ruess, M., and Schillinger, D. (2016). The tetrahedral finite cell method: Higher-order immersogeometric analysis on adaptive non-boundary-fitted meshes. *International Journal for Numerical Methods in Engineering*, 107(12):1054–1079.

Verfürth, R. (1996). *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Teubner Verlag, Chichester, New York.

Verfürth, R. (1999). A review of a posteriori error estimation techniques for elasticity problems. *Computer Methods in Applied Mechanics and Engineering*, 176(1–4):419–440.

Verhoosel, C., van Zwieten, G., van Rietbergen, B., and de Borst, R. (2015). Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. *Computer Methods in Applied Mechanics and Engineering*, 284:138–164.

Verhoosel, C. V., Remmers, J. J. C., and Gutiérrez, M. A. (2009). A dissipation-based arc-length method for robust simulation of brittle and ductile failure. *International Journal for Numerical Methods in Engineering*, 77(9):1290–1321.

Verhoosel, C. V., Scott, M. A., de Borst, R., and Hughes, T. J. R. (2011). An isogeometric approach to cohesive zone modeling. *International Journal for Numerical Methods in Engineering*, 87(1-5):336–360.

Wang, Y. and Waisman, H. (2015). Progressive delamination analysis of composite materials using XFEM and a discrete damage zone model. *Computational Mechanics*, 55(1):1–26.

WHE (2016). World Housing Encyclopedia (WHE) Report Database. `http://db.world-housing.net/building/21`.

Wiley (2016). Wiley-VCH Copyright Transfer Agreement. `http://www.wiley-vch.de/util/cta/PS_Socs.pdf`.

Wille, H., Ruess, M., Rank, E., and Yosibash, Z. (2016). Uncertainty quantification for personalized analyses of human proximal femurs. *Journal of Biomechanics*, 49(4):520–527.

Wriggers, P. (2006). *Computational Contact Mechanics*. Springer, Berlin, New York, 2nd edition.

Wriggers, P. (2008). *Nonlinear Finite Element Methods*. Springer, Berlin.

Xu, F., Schillinger, D., Kamensky, D., Varduhn, V., Wang, C., and Hsu, M.-C. (2015). The tetrahedral finite cell method for fluids: Immersogeometric analysis of turbulent flow around complex geometries. *Computers & Fluids*, in press.

Xu, X.-P. and Needleman, A. (1993). Void nucleation by inclusion debonding in a crystal matrix. *Modelling and Simulation in Materials Science and Engineering*, 1(2):111–132.

Xu, X. P. and Needleman, A. (1994). Numerical simulations of fast crack growth in brittle solids. *Journal of the Mechanics and Physics of Solids*, 42(9):1397–1434.

Yang, Q. and Cox, B. (2005). Cohesive models for damage evolution in laminated composites. *International Journal of Fracture*, 133(2):107–137.

Yang, Z. (2011). *The Finite Cell Method for Geometry-Based Structural Simulation*. PhD thesis, Technische Universität München, Munich.

Yang, Z., Kollmannsberger, S., Düster, A., Ruess, M., Garcia, E. G., Burgkart, R., and Rank, E. (2012a). Non-standard bone simulation: Interactive numerical analysis by computational steering. *Computing and Visualization in Science*, 14(5):207–216.

Yang, Z., Ruess, M., Kollmannsberger, S., Düster, A., and Rank, E. (2012b). An efficient integration technique for the voxel-based finite cell method. *International Journal for Numerical Methods in Engineering*, 91(5):457–471.

Yosibash, Z. (2012). *Singularities in Elliptic Boundary Value Problems and Elasticity and Their Connection with Failure Initiation*. Number 37 in Interdisciplinary applied mathematics. Springer, New York, USA.

Yserentant, H. (1986a). Hierarchical bases give conjugate gradient type methods a multigrid speed of convergence. *Applied Mathematics and Computation*, 19(1):347–358.

Yserentant, H. (1986b). On the multi-level splitting of finite element spaces. *Numerische Mathematik*, 49(4):379–412.

Zander, N., Bog, T., Elhaddad, M., Espinoza, R., Hu, H., Joly, A., Wu, C., Zerbe, P., Düster, A., Kollmannsberger, S., Parvizian, J., Ruess, M., Schillinger, D., and Rank, E. (2014). FCMLab: A finite cell research toolbox for MATLAB. *Advances in Engineering Software*, 74:49–63.

Zander, N., Bog, T., Elhaddad, M., Frischmann, F., Kollmannsberger, S., and Rank, E. (2016). The multi-level hp-method for three-dimensional problems: Dynamically changing high-order mesh refinement with arbitrary hanging nodes. *Computer Methods in Applied Mechanics and Engineering*, 310:252–277.

Zander, N., Bog, T., Kollmannsberger, S., Schillinger, D., and Rank, E. (2015). Multi-level hp-adaptivity: High-order mesh adaptivity without the difficulties of constraining hanging nodes. *Computational Mechanics*, 55(3):499–517.

Zander, N., Kollmannsberger, S., Ruess, M., Yosibash, Z., and Rank, E. (2012). The Finite Cell Method for linear thermoelasticity. *Computers & Mathematics with Applications*, 64(11):3527–3541.

Zander, N., Ruess, M., Bog, T., Kollmannsberger, S., and Rank, E. (2017). Multi-level hp-adaptivity for cohesive fracture modeling. *International Journal for Numerical Methods in Engineering*, 109(13):1723–1755.

Zhuang, X. (2010). *Meshless Methods: Theory and Application in 3D Fracture Modelling with Level Sets*. PhD thesis, University of Durham.

Zienkiewicz, O., Taylor, R., and Zhu, J. (2005a). *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, 6th edition.

Zienkiewicz, O. C. and Craig, A. (1986). Adaptive refinement, error estimates, multigrid solution and hierarchic finite element method concepts. In Babuska, I., Zienkiewicz, O. C., Gago, J., and de Oliviera, E. R., editors, *Accuracy Estimates and Adaptive Refinements in Finite Element Calculations*, pages 25–55. Wiley, New York.

Zienkiewicz, O. C., Taylor, R. L., and Nithiarasu, P. (2005b). *The Finite Element Method for Fluid Dynamics*. Butterworth-Heinemann, Oxford, 6th edition.