TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Computation in Engineering

Frictionless contact simulation using the finite cell method

Tino Bog

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

Doktor-Ingenieurs

genehmigten Dissertation

Vorsitzender: Prof. Dr.-Ing. habil. Fabian Duddeck

Prüfer der Dissertation:

1. Prof. Dr.rer.nat. Ernst Rank
2. Prof. Dr.-Ing. Laura de Lorenzis
   Technische Universität Braunschweig

Die Dissertation wurde am 19.07.2017 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 03.10.2017 angenommen.

To my family

Zusammenfassung

Die vorliegende Arbeit behandelt zwei Formulierungen für die Simulation reibungsfreier Kontaktprobleme mit der Finiten Zellen Methode (FCM). Die FCM ist ein *Fictitious Domain* Ansatz, der auf Finiten Elementen hoher Ordnung basiert.
Die erste Formulierung erweitert das Konzept schwacher Randbedingungen hin zu Ungleichheitsbedingungen für reibungsfreien Kontakt. Die Kontaktbedingungen werden an eingebetteten Rändern aufgebracht, welche aus Polynomsegmenten hoher-Ordnung bestehen. Für die Erstellung dieser Segmente wird ein erweitertes *Maching Cubes* (MC) Verfahren vorgestellt. Irreguläre Bereiche am Rand des Kontaktgebiets werden mit Hilfe des *multi-level hp* Verfahrens eingegrenzt um die Genauigkeit der Approximation zu verbessern. Numerische Beispiele und systematische Studien mit *h*-, *p*- und *hp*-Verfeinerungen zeigen, dass diese Kontaktformulierung sehr exakte Ergebnisse erzielen kann.
Die zweite Kontaktformulierung verwendet ein spezielles Materialmodell, welches in die Zwischenräume von zwei- bzw. dreidimensionalen Körpern eingefügt wird. Die Kontaktbedingungen werden somit in der gleichen Raumdimension wie das zugrunde liegende mechanische Problem aufgebracht. Die Formulierung ist den Barriereverfahren zuzuordnen und führt zu einer Regularisierung der *Karush-Kuhn-Tucker* Bedingungen. Die vorgestellte Methode unterstützt Selbstkontakt direkt und bedarf keiner expliziten Kontaktsuche. Numerische Untersuchen werden für zwei- und dreidimensionale Probleme, welche auch ein Beispiel nach Hertz beinhalten, durchgeführt. Zudem wird die Methode mit einem kommerziellen Finiten Elemente Paket verglichen.

Abstract

This thesis presents two formulations to model frictionless contact using the finite cell method (FCM), a fictitious domain approach based on high-order finite elements.
The first contact formulation extends the concept of weakly enforced boundary conditions to inequality constraints for frictionless contact. The constraints are enforced on embedded interfaces, which are represented by high-order polynomial segments. The segments are recovered automatically using an extended Marching Cubes (MC) algorithm. To further improve the accuracy of the discretization, irregularities at the boundary of contact zones are treated with multi-level *hp*-refinements. Numerical results and systematic studies of *h*-, *p*- and *hp*-refinements show that this contact formulation can provide accurate results for problems involving contact.
The second contact formulation employs a specially designed material model that is inserted into two- respectively three-dimensional regions surrounding contacting bodies. Contact constraints are thus enforced on the same manifold as the accompanying structural problem. The application of the current material formulation leads to a regularization of the Karush-Kuhn-Tucker conditions. The formulation can be classified as a barrier-type method. The proposed formulation handles self-contact naturally. Since the non-penetrating conditions are solved in a physically consistent manner, there is no need for an explicit contact search. Results are obtained for two- and three-dimensional problems, including a Hertzian contact problem. Comparisons to a commercial FEA package are provided.

Acknowledgments

This thesis has been created during my time at the chair of Computation in Engineering of Prof. rer. nat. Ernst Rank at the Technical University of Munich. From April 2011 until December 2016 I had the chance to conduct scientific research and actively engage in the organization and support of lectures and events. I joined the development of the chairs new numerical simulation framework.

I would, therefore, first like to thank my doctoral supervisor Prof. Rank for his constant support and all the opportunities he provided. I also thank the German Research Foundation (DFG) and the Elite Network of Bavaria (ENB) for their financial support.

Furthermore, I would like to thank the many colleagues, students and researchers I met along the way, who made this journey even more enjoyable and interesting. These include all the people at the Chair for Computation in Engineering and the Chair for Computational Modeling and Simulation as well as the organizers and students of the ENB's Bavarian Graduate School of Computational Engineering.

I am especially thankful for getting the chance to share an office with my colleague and friend Nils Zander, who could always inspire me with his scientific curiosity and knowledge.

I thank Prof. Dr.-Ing. Laura de Lorenzis for reviewing my thesis and Prof. Dr.-Ing. Fabian Duddeck for chairing my examination.

Finally, I want to thank my family for their love and their constant and ongoing support; above all my wife Verena.

Contents

Chapter 1

Introduction

1.1 Motivation

Computer hardware has seen a massive increase in performance in the last decades while getting more affordable at the same time. Today, there are supercomputing clusters that allow for large simulations both at the global and the molecular scale. Also, standard workstations are now able to perform computations, which would have needed a small cluster a decade ago.

In the field of computer-aided engineering (CAE) these developments allow for simulations with much higher fidelity than ever before. As a consequence, typical simulation models have also become more computationally involved. Compare for example the rather crude models used for early space craft design (Figure 1.1a) to today's detailed, fully three-dimensional models (Figure 1.1b).

While numerical simulations largely utilize all the performance currently available, the workflows leading to these simulations (model setup) can still be considered classical, since they involve manual intervention for the most part [23]. In the field of solid mechanics, where the finite element method (FEM) is the prevalent numerical method [56, 9], geometries have to be discretized (meshed) to perform an analysis. Meshing can be very involved, especially if the geometric model under consideration contains faults or superfluous features [40], which is generally the case for a sophisticated engineering design.

The emerging field of fictitious or embedded domain methods offers a solution for the problem of mesh generation [43, 52, 97]. The main idea of these methods is to embed the problem geometry in an unfitted mesh, thus avoiding all the issues related to mesh generation. The geometry is taken into consideration at the integration level using specialized quadrature schemes. This numerical integration can be computationally more expensive but may be performed automatically. Then, considering that *computing hours* are typically much cheaper than *(human) engineering hours*, fictitious domain methods have become a notable alternative to classical, body-fitted approaches.

One such fictitious domain approach, which will also be the focus of this work, is the finite cell method (FCM) [89, 26]. The FCM combines the idea of fictitious domains with high-order finite elements. Unlike low-order finite elements, whose accuracy is improved by reducing the element size in regions containing high-error contributions, high-order finite elements reduce the error by elevating their polynomial degree [121]. Besides their superior accuracy, high-order finite elements also alleviate many locking problems and perform robustly under large strain

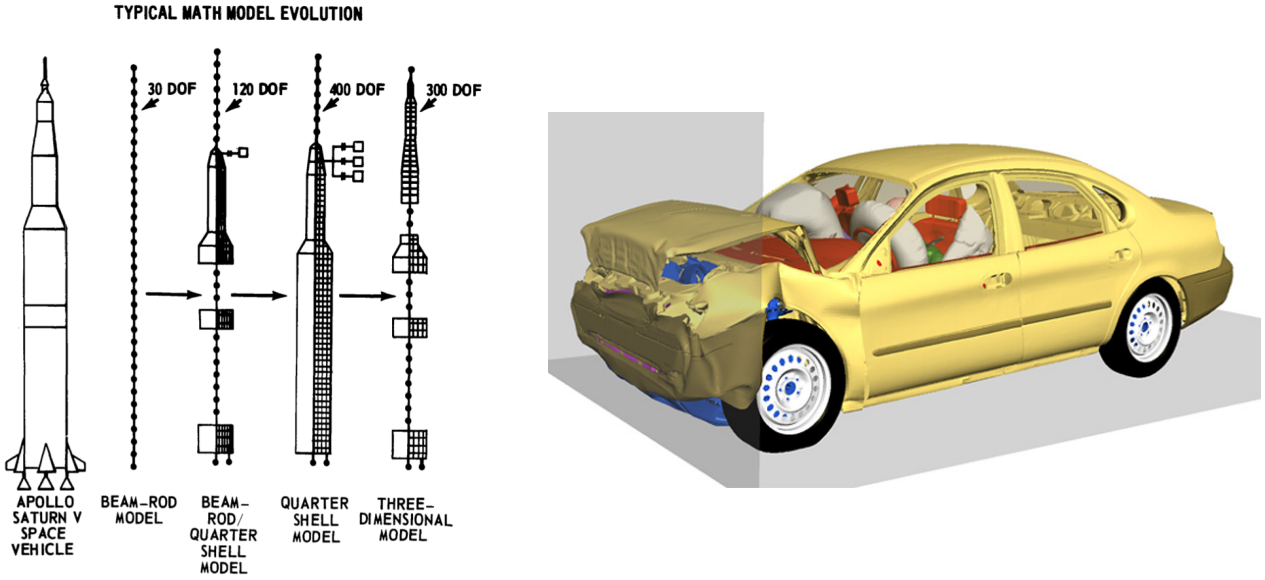


Figure 1.1: Complexity of finite element simulations then and now. (a) Different models for studying the dynamics of the Apollo Saturn V. [45] (b) Crash simulation using 10 million elements. (Image courtesy of Altair Inc.)

deformations [50, 84]. Due to their high approximation power, larger and thus fewer elements can be utilized, which reduces the number of unknowns in a numerical simulation.
Another approach based on higher order shape functions, which tries to reduce the effort of geometry cleanup and mesh generation is iso-geometric analysis (IGA) [55, 23]. While this method is not in the focus of this work, it gained very much attention by the computational mechanics community in recent years. The main idea of IGA is to directly utilize the B-Spline* or NURBS† functions [90], which are used by most commercial computer-aided design (CAD) systems to represent the geometry, as shape functions for the analysis. Unfortunately, IGA often cannot be applied directly to three-dimensional solid CAD models, since these are mostly represented by their boundaries and not by tri-variate Splines. Another problem of IGA is that bodies and their boundaries can be constructed arbitrarily, which may lead to a high number of trimmed spline-patches. While there are ways to re-parametrize trimmed surfaces (patches), e.g. using T-Splines [11], these approaches are algorithmically complex.
The goal of this work is to carry over the advantages of fictitious domain approaches to the field of contact mechanics. Contact plays an important role in engineering practice. Mechanical systems are typically made up of several parts whose interaction needs to be represented correctly. Contact simulations help to predict the load paths resulting from different loading scenarios and thus allow to assess the level of utilization for mechanical parts. Hence, a crucial aspect of many solid mechanics problems is the incorporation of contact effects. Application scenarios range from quasi-static analysis of multi-part assemblies to highly dynamic crash simulations.

*Basis-spline
†Non-uniform rational b-spline

While there is a large body of research on contact for standard finite elements (e.g. [133, 71]), this aspect has so far gained comparatively little attention in the field of fictitious domain methods.

1.2 Scope

The aim of this work is to investigate formulations to enforce contact constraints within the framework of high-order fictitious domain methods, especially the finite cell method. To this end, we extend the concept of weakly enforced boundary and interface conditions to a frictionless *mortar-type* contact formulation based on the penalty method.
The constraints will generally be applied on embedded interfaces. Depending on the type of geometry representation used, these interfaces (surfaces) might not be available explicitly. Therefore, an approach is presented to automatically recover embedded interfaces of any geometry model that provides a *point-in-membership* test. The recovered surfaces will be approximated using geometric segments of arbitrary polynomial order.
Furthermore, a *contact material* approach is presented, which enforces the contact constraints on the same manifold as the problem geometry. The basic idea is to *fill* the fictitious domain with an appropriate hyperelastic material model.
Both contact formulations will be applied to two- and three-dimensional elastic problems in the small and large strain regime.
Local mesh refinement based on the multi-level *hp*-method [139] is employed, to reduce oscillations stemming from the singularities at the boundary of the contact zone.

1.3 Structure

This thesis is structured as follows. Chapter 2 recalls some basic concepts of continuum mechanics, which are needed to formulate a boundary value problem for elasticity with small as well as large deformations. The chapter ends with the weak form of the boundary value problem, which is the basis for any finite element formulation.
A short overview of classical computational contact mechanics for frictionless problems is given in Chapter 3. The basic steps of a standard finite element contact simulation are explained, and the most common constraint formulations are recalled.
Chapter 4 presents the finite cell method for problems in nonlinear elasticity. Also, the fundamental ingredients like numerical integration of discontinuous integrands as well as the weak enforcement of essential boundary and interface conditions in a nonlinear setting are recalled.
In Chapter 5 a formulation is presented to recover embedded interfaces using high-order geometric segments. For completeness, the chapter provides an overview of the most common geometric models.
Chapter 6 introduces a mortar-type formulation for frictionless contact constraints on embedded interfaces for the FCM. The aspects of global and local searches are also addressed since they need special consideration due to the separation of the contact boundary and the solution mesh. The chapter is complemented by numerical studies investigating the accuracy of the proposed formulation.

A material contact approach based on a hyperelastic material law is introduced in Chapter 7. The frictionless behaviour of the formulation is discussed and a safeguard algorithm is presented, which avoids interpenetration of adjacent contact boundaries. The chapter finishes with some studies investigating the nonlinear solution behavior. Also, a comparison is provided with a commercial FE-software package regarding accuracy and efficiency.
This thesis closes with a summary, conclusions and an outlook in Chapter 8.

Chapter 2

Basic continuum mechanics

This chapter briefly summarizes the basic concepts of continuum mechanics, to provide a foundation for the topics discussed in the following chapters. These concepts include large strain kinematics, stress tensors as well as quasi-static equilibrium and constitutive material models for elasticity. For further details, the reader is referred to the classical monographs by Malvern [75] and Ogden [87] or the more recent works by Holzapfel [53] and Bonet and Wood [17]. The notation used in this book is mostly based on the work by Bonet and Wood [17].
The field of continuum mechanics provides the means to mathematically describe the mechanical behavior of deformable bodies. Therefore the bodies are assumed to be composed of a continuous medium, which does not contain any gaps or empty spaces [75]. This assumption is valid only at length scales much greater than the inter-atomic distances (macro-scale). However, it applies for the majority of engineering problems in solid mechanics. Engineers are mostly interested in the configuration that deformable bodies assume when they are subjected to boundary conditions and external loads. In most practical cases these boundary value problems cannot be solved in closed form. Hence, numerical methods are employed to arrive at approximate solutions. In the field of solid mechanics, the most prominent approach is the finite element method (FEM), which is based on the weak form of equilibrium. The weak form is obtained with the help of variational calculus taking into account the kinematic assumptions and constitutive equations for the problem at hand. In the case of nonlinearities, the weak form needs to be linearized to allow for a solution using a Newton-type method.

2.1 Kinematics

The field of kinematics provides the mathematical description of the geometry of a mechanical system. Kinematics also describe how the various parts of such a system move and deform, without considering the causes.

2.1.1 Motion

The coordinate $\boldsymbol{x}$ of a point in the current or deformed configuration is connected to the point's initial coordinate $\boldsymbol{X}$ via a deformation function $\boldsymbol{\varphi}$

$$\boldsymbol{x} = \boldsymbol{\varphi}\left(\boldsymbol{X}, t\right) . \tag{2.1}$$

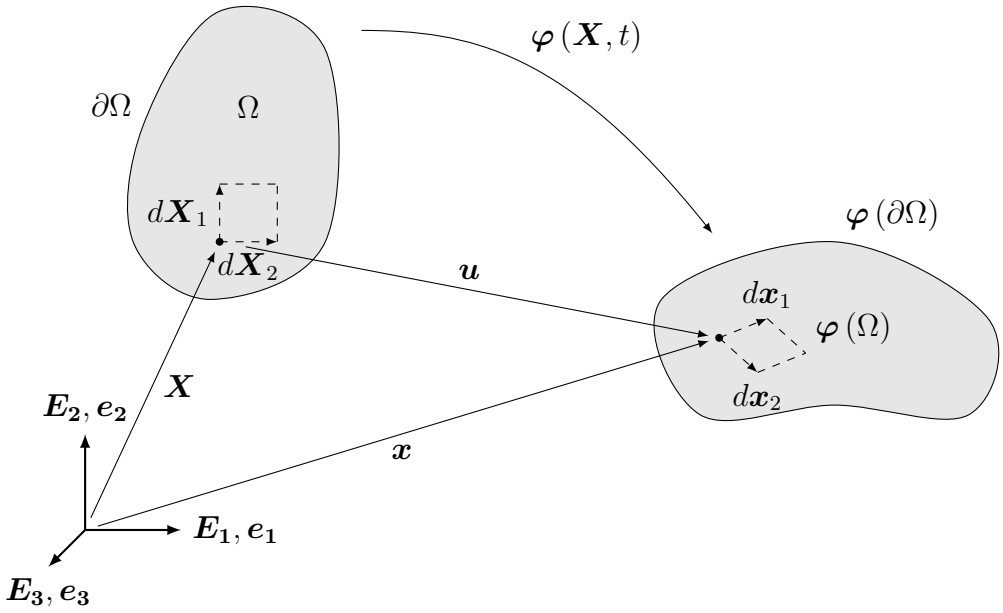


Figure 2.1: Motion of a material point.

This is depicted in Figure 2.1 for a single elastic body. In its initial configuration, this body occupies a domain Ω, whose boundary is denoted by $\partial\Omega$. With the help of the deformation function $\boldsymbol{\varphi}$, the domain and its boundary are pushed forward to their deformed configurations $\boldsymbol{\varphi}(\Omega)$ and $\boldsymbol{\varphi}(\partial\Omega)$.
The coordinates $\boldsymbol{x}$ and $\boldsymbol{X}$ are defined using orthonormal bases with the unit vectors $\boldsymbol{E}_i$ in the initial configuration and $\boldsymbol{e}_i$ the current configuration

$$\boldsymbol{X} = X_i \boldsymbol{E}_i \tag{2.2a}$$
$$\boldsymbol{x} = x_i \boldsymbol{e}_i \,. \tag{2.2b}$$

For convenience, we follow the generally adopted approach of choosing both bases to be the same [53, 17]. The difference between the initial and the deformed coordinates of a point is denoted by the displacement vector $\boldsymbol{u}(t)$:

$$\boldsymbol{u} = \boldsymbol{u}(t) = \boldsymbol{x} - \boldsymbol{X} \ . \tag{2.3}$$

2.1.2 Deformation

The deformation of a body Ω can be expressed with the help of the deformation gradient $\boldsymbol{F}$. The deformation gradient maps an infinitesimal line element $d\boldsymbol{X}$ from the initial configuration to a line element in the deformed configuration $d\boldsymbol{x}$, such that

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{F} \cdot \mathrm{d}\boldsymbol{X} \ . \tag{2.4}$$

Hence, the deformation gradient can be computed as

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \frac{\partial (\boldsymbol{X} + \boldsymbol{u})}{\partial \boldsymbol{X}} = \boldsymbol{I} + \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}} = \boldsymbol{I} + \boldsymbol{\nabla}_{\boldsymbol{X}} \boldsymbol{u} \ , \tag{2.5}$$

where $\boldsymbol{\nabla}_{\boldsymbol{X}}(\cdot)$ denotes the gradient with respect to the initial configuration. The deformation gradient is a non-symmetric two-point tensor, which maps tensors from the initial configuration to the deformed configuration

$$\boldsymbol{F} = F_{ij}\boldsymbol{e}_i \otimes \boldsymbol{E}_j \ . \tag{2.6}$$

To be physically meaningful, the determinant of the deformation gradient has to fulfill the following condition

$$\det \boldsymbol{F} = J > 0 \ . \tag{2.7}$$

This means that material does not vanish or interpenetrate itself. Hence, an inverse mapping of (2.4) is defined as

$$\mathrm{d}\boldsymbol{X} = \boldsymbol{F}^{-1}\mathrm{d}\boldsymbol{x} \ . \tag{2.8}$$

The deformation at any point inside an elastic body can be decomposed into a rigid body rotation and a change of shape (stretch)

$$\boldsymbol{F} = \boldsymbol{R} \cdot \boldsymbol{U} \ , \tag{2.9a}$$
$$\boldsymbol{F} = \boldsymbol{V} \cdot \boldsymbol{R} \ . \tag{2.9b}$$

Here, $\boldsymbol{R}$ denotes the tensor of rotation and $\boldsymbol{U}$ and $\boldsymbol{V}$ are the symmetric material and spatial stretch tensors, respectively (see Figure 2.2).

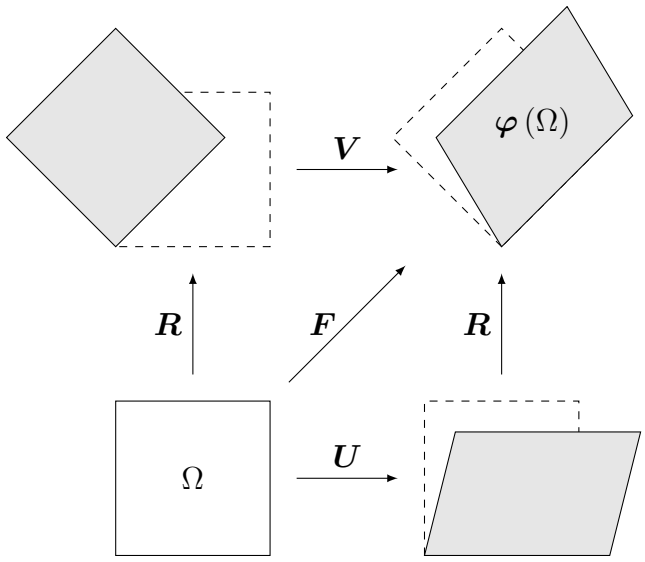


Figure 2.2: Polar decomposition of the deformation gradient.

Many constitutive models are formulated in principal quantities. Hence, it is useful to express the deformation gradient using its spectral decomposition

$$\boldsymbol{F} = \sum_{i=1}^{3} \lambda_i \boldsymbol{n}_i \otimes \boldsymbol{N}_i \ . \tag{2.10}$$

Here, the quantities λ_i denote principal stretches, which can be computed from the following eigenvalue problems:

$$\left(\boldsymbol{U}^2 - \lambda_i^2 \boldsymbol{I}\right) \boldsymbol{N}_i = \left(\boldsymbol{C} - \lambda_i^2 \boldsymbol{I}\right) \boldsymbol{N}_i = \boldsymbol{0} \tag{2.11a}$$

$$\left(\boldsymbol{V}^2 - \lambda_i^2 \boldsymbol{I}\right) \boldsymbol{n}_i = \left(\boldsymbol{b} - \lambda_i^2 \boldsymbol{I}\right) \boldsymbol{n}_i \quad = \boldsymbol{0} \ . \tag{2.11b}$$

The vectors $\boldsymbol{N}_i$ and $\boldsymbol{n}_i$ are the associated eigenvectors, generally referred to as material and spatial principal directions, respectively. The tensors $\boldsymbol{C}$ and $\boldsymbol{b}$ are the right and left Cauchy-Green tensors, respectively

$$\boldsymbol{C} = \boldsymbol{F}^{\mathrm{T}} \boldsymbol{F} = \left(\boldsymbol{U}^{\mathrm{T}} \cdot \boldsymbol{R}^{\mathrm{T}}\right) \cdot (\boldsymbol{R} \cdot \boldsymbol{U}) = \boldsymbol{U}^2 \ , \tag{2.12a}$$

$$\boldsymbol{b} = \boldsymbol{F} \boldsymbol{F}^{\mathrm{T}} = (\boldsymbol{V} \cdot \boldsymbol{R}) \cdot \left(\boldsymbol{R}^{\mathrm{T}} \cdot \boldsymbol{V}^{\mathrm{T}}\right) = \boldsymbol{V}^2 \ . \tag{2.12b}$$

In the same manner as the deformation gradient (2.10) the stretch and Cauchy-Green tensors can be expressed in polar form

$$\boldsymbol{U} = \sum_{i=1}^{3} \lambda_i \boldsymbol{N}_i \otimes \boldsymbol{N}_i \ , \qquad \boldsymbol{C} = \sum_{i=1}^{3} \lambda_i^2 \boldsymbol{N}_i \otimes \boldsymbol{N}_i \ , \tag{2.13a}$$

$$\boldsymbol{V} = \sum_{i=1}^{3} \lambda_i \boldsymbol{n}_i \otimes \boldsymbol{n}_i \ , \qquad \boldsymbol{b} = \sum_{i=1}^{3} \lambda_i^2 \boldsymbol{n}_i \otimes \boldsymbol{n}_i \ . \tag{2.13b}$$

In addition to mapping line elements from the initial configuration to the deformed configuration, the deformation gradient can also be used to map area and volume elements. Nanson's formula [17] maps an area element $\mathrm{d}A$ from the initial configuration to an area element $\mathrm{d}a$ in the deformed configuration

$$\mathrm{d}a \ \boldsymbol{n} = \mathrm{d}A \ J \boldsymbol{F}^{-\mathrm{T}} \cdot \boldsymbol{N} \ , \tag{2.14}$$

where $\boldsymbol{n}$ and $\boldsymbol{N}$ are unit normal vectors on the initial and deformed area element, respectively. The volumetric mapping is simply expressed by

$$\mathrm{d}v = J \ \mathrm{d}V \ , \tag{2.15}$$

where J is the determinant of the deformation gradient (2.7).

2.1.3 Strains

In engineering practice deformations are generally not measured using the deformation gradient $\boldsymbol{F}$ directly. First, $\boldsymbol{F}$ is not zero (but one) for an undeformed body. Also, since $\boldsymbol{F}$ includes rigid body rotations, different measures are obtained for undeformed bodies subjected to rotations. Therefore, a better option is to employ derived measures, called strains, which do not suffer from these drawbacks. The most frequently employed strains stem from the Seth-Hill family of strain measures. They can be expressed as

$$\boldsymbol{E}_{(m)} = \frac{1}{m} \left(\boldsymbol{U}^m - \boldsymbol{I}\right) = \frac{1}{m} \sum_{i=1}^{N} \left(\lambda_i^m - 1\right) \boldsymbol{N}_i \otimes \boldsymbol{N}_i \ . \tag{2.16}$$

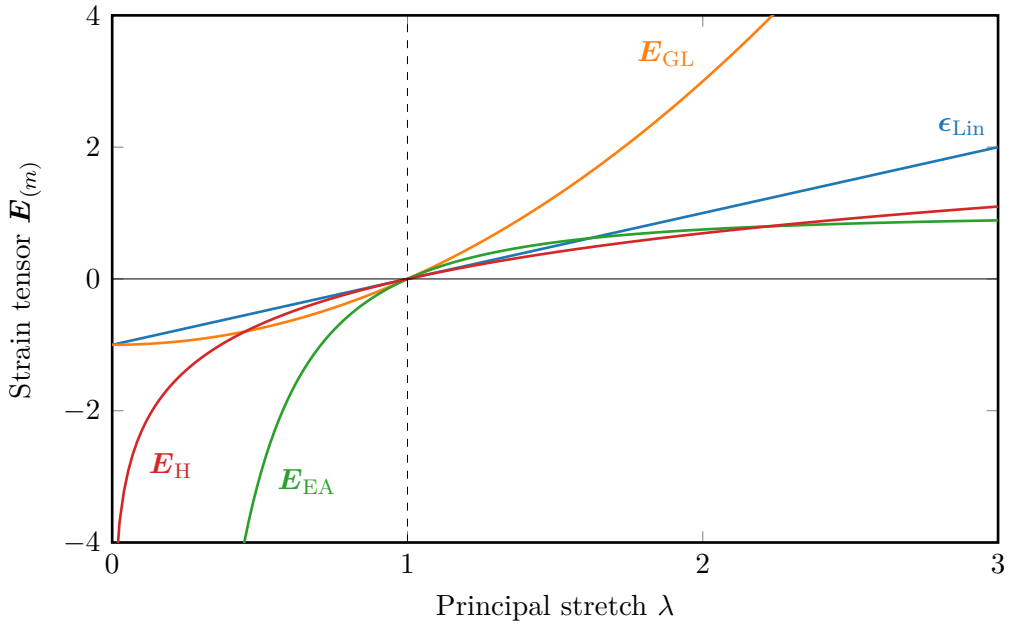


Figure 2.3: Comparison of different strain measures for the one-dimensional case.

Depending on the value of m, different measures can be obtained (see Figure 2.3):

$$\boldsymbol{E}_{(2)} = \boldsymbol{E}_{\mathrm{GL}} = \frac{1}{2}\left(\boldsymbol{U}^2 - \boldsymbol{I}\right) = \frac{1}{2}\sum_{i=1}^{N}\left(\lambda_i^2 - 1\right)\boldsymbol{N}_i \otimes \boldsymbol{N}_i \qquad \text{Green-Lagrange} \tag{2.17a}$$

$$\boldsymbol{E}_{(0)} = \boldsymbol{E}_{\mathrm{H}} = \ln \boldsymbol{U} = \sum_{i=1}^{N} \ln \lambda_i \boldsymbol{N}_i \otimes \boldsymbol{N}_i \qquad \text{Hencky} \tag{2.17b}$$

$$\boldsymbol{E}_{(-2)} = \boldsymbol{E}_{\mathrm{EA}} = \frac{1}{2}\left(\boldsymbol{I} - \boldsymbol{U}^{-2}\right) = \frac{1}{2}\sum_{i=1}^{N}\left(1 - \lambda_i^{-2}\right)\boldsymbol{N}_i \otimes \boldsymbol{N}_i \qquad \text{Euler-Almansi} \tag{2.17c}$$

Note that, in the case of small deformations ($\|\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u}\| \ll \boldsymbol{1}$) all of these strain measures reduce to the linear strain tensor $\boldsymbol{\epsilon}_{\mathrm{Lin}}$. Consider for example the Green-Lagrange strain tensor

$$\begin{aligned}
\boldsymbol{E}_{\mathrm{GL}}\big|_{\|\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u}\|\to\boldsymbol{0}} &= \frac{1}{2}\left(\boldsymbol{U}^2 - \boldsymbol{I}\right) = \frac{1}{2}\left(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F} - \boldsymbol{I}\right) \\
&= \frac{1}{2}\left[\boldsymbol{I} + \left(\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u}\right)^{\mathrm{T}} + \boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u} + \underbrace{\left(\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u}\right)^{\mathrm{T}}\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u}}_{\to\boldsymbol{0}} - \boldsymbol{I}\right] \\
&= \frac{1}{2}\left[\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u} + \left(\boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{u}\right)^{\mathrm{T}}\right] = \boldsymbol{\epsilon}_{\mathrm{Lin}} \ .
\end{aligned} \tag{2.18}$$

2.2 Stress and equilibrium

In a purely mechanical setting deformations are a result of external loads and body forces. These loads result in stress fields inside the elastic bodies, which is of major interest in engineering

practice, since stress results allow for assertions about the degree of utilization and possible failure of materials. Stresses are in equilibrium with applied loads as stated by the conservation laws of linear and angular momentum.

2.2.1 Stress tensor

The resultant of a force $\mathrm{d}\boldsymbol{f}$ acting on a surface $\mathrm{d}a$ in the current configuration is called surface load or traction

$$\boldsymbol{t}\left(\boldsymbol{x}, t, \boldsymbol{n}\right) = \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}a} \ . \tag{2.19}$$

The Cauchy theorem states that this surface load and the outward unit normal $\boldsymbol{n}$ defined on the surface $\mathrm{d}a$ are linearly connected by the Cauchy stress tensor $\boldsymbol{\sigma}$

$$\boldsymbol{t}\left(\boldsymbol{x}, t\right) = \boldsymbol{\sigma}\left(\boldsymbol{x}, t\right)\boldsymbol{n} \ . \tag{2.20}$$

The Cauchy stress tensor is a symmetric tensor, which implies the conservation of angular momentum [17]

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} = \boldsymbol{\sigma}^{\mathrm{T}} \ . \tag{2.21}$$

Figure 2.4 depicts the common sign convention for the Cauchy stress tensor.

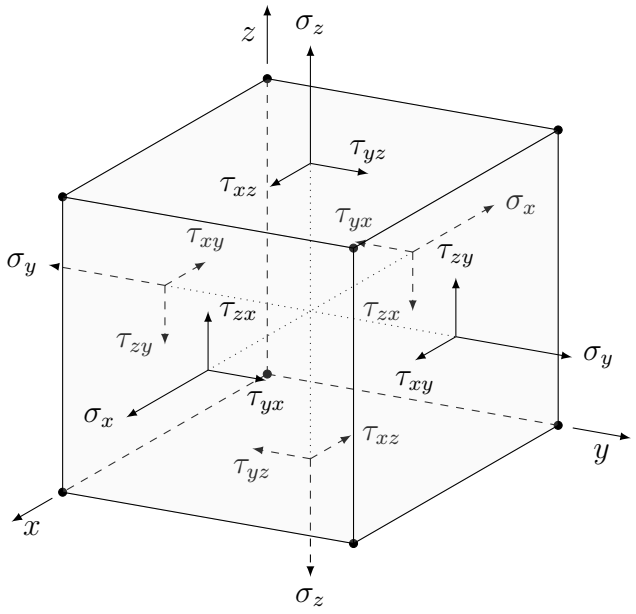


Figure 2.4: Sign convention of stress tensor components on an infinitesimal volume element.

Other well-known stress tensors include

$$\boldsymbol{\tau} = J\boldsymbol{\sigma} \qquad \text{Kirchhoff stress} \tag{2.22}$$

$$\boldsymbol{P} = J\boldsymbol{\sigma}\boldsymbol{F}^{-\mathrm{T}} \qquad 1^{\mathrm{st}} \text{ Piola-Kirchhoff stress} \tag{2.23}$$

$$\boldsymbol{S} = J\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-\mathrm{T}} \qquad 2^{\mathrm{nd}} \text{ Piola-Kirchhoff stress .} \tag{2.24}$$

The 2nd Piola-Kirchhoff stress tensor is generally employed for the derivation and formulation of nonlinear elasticity in the initial (undeformed) configuration.

2.2.2 Equilibrium

Using the Cauchy stress tensor the conservation of linear momentum can be expressed by a set of partial differential equations. If inertia effects can be neglected or the loads are applied very slowly, Cauchy's equation of equilibrium for elastostatics [53] holds and reads as follows for the three-dimensional case

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + b_x = 0 \tag{2.25a}$$

$$\frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} + b_y = 0 \tag{2.25b}$$

$$\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + b_z = 0 \ . \tag{2.25c}$$

Here, $\boldsymbol{b} = [b_x, b_y, b_z]$ denotes a prescribed body force per unit volume. By applying tensor notation the equilibrium equations (2.25) can be written compactly as

$$\operatorname{div} \boldsymbol{\sigma} + \boldsymbol{f} = 0 \ . \tag{2.26}$$

2.3 Constitutive models for isotropic elasticity

Constitutive models connect the deformation occurring inside deformable bodies to the resulting internal stresses. The simplest forms of constitutive models in solid mechanics are elastic models. Elastic bodies return to their initial configuration after external loads and body forces have been removed. Hence, the stress field in the current configuration does not depend on the load history or the rate of loading [53, 17]. Furthermore, if the stress response is independent of the direction of application, the constitutive model is said to be isotropic.

2.3.1 Linear elasticity

In the case of small deformations the model of linear elasticity is applicable. This model, also known as Hooke's law, connects the linear strain tensor with the Cauchy stress tensor*

$$\boldsymbol{\sigma} = \boldsymbol{\mathcal{C}}\boldsymbol{\epsilon}_{\mathrm{Lin}} \ . \tag{2.27}$$

Here, $\boldsymbol{\mathcal{C}}$ is the linear elasticity tensor defined as

$$\boldsymbol{\mathcal{C}} = \Lambda \boldsymbol{I} \otimes \boldsymbol{I} + 2\mathrm{G}\boldsymbol{i} \ , \tag{2.28}$$

*Note that any of the stress tensors introduced before would be applicable, since they are all equivalent for small deformations: $\boldsymbol{F} \gg \boldsymbol{I}$ and $J \gg 1$.

where $\boldsymbol{I}$ denotes the second order identity tensor and $\boldsymbol{i}$ is the symmetric part of the fourth order identity tensor [17]

$$\boldsymbol{I} = \delta_{ij} \ , \quad \boldsymbol{i} = \frac{1}{2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) \ , \quad \delta_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ else} \end{cases} \ . \tag{2.29}$$

The parameters Λ and G are the well-known Lamé coefficients

$$\Lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \ , \qquad \mathrm{G} = \frac{E}{2(1+2\nu)} \ , \tag{2.30}$$

where G is also commonly referred to as shear modulus.

2.3.2 Hyperelasticity

In the case of hyperelasticity, also called Green-elasticity, the stress-strain-relationship can be derived from a scalar valued strain energy function $W(\boldsymbol{F})$ [87, 17, 53]. For an isotropic response the strain energy function can be expressed in terms of the invariants of the left and right Cauchy-Green tensors

$$W(\boldsymbol{F}) = W(I_1, I_2, I_3) = W(\lambda_1, \lambda_2, \lambda_3) \ , \tag{2.31}$$

where the invariants are defined as

$$I_1 = \boldsymbol{C} : \boldsymbol{I} \ = \boldsymbol{b} : \boldsymbol{I} \ = \lambda_1 + \lambda_2 + \lambda_3 \tag{2.32a}$$
$$I_2 = \boldsymbol{C} : \boldsymbol{C} = \boldsymbol{b} : \boldsymbol{b} \ = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \tag{2.32b}$$
$$I_3 = \det \boldsymbol{C} \ = \det \boldsymbol{b} = \lambda_1^2 \lambda_2^2 \lambda_3^2 \ . \tag{2.32c}$$

2.3.2.1 Stress tensors from strain energy functions

The 2nd Piola-Kirchhoff stress tensor can be derived using the following relation

$$\boldsymbol{S}(I_1, I_2, I_3) = 2 \underbrace{\frac{\partial W}{\partial I_1}}_{W_1} \frac{\partial I_1}{\partial \boldsymbol{C}} + 2 \underbrace{\frac{\partial W}{\partial I_2}}_{W_2} \frac{\partial I_2}{\partial \boldsymbol{C}} + 2 \underbrace{\frac{\partial W}{\partial I_3}}_{W_3} \frac{\partial I_3}{\partial \boldsymbol{C}} \ , \tag{2.33}$$

and can be written in compact form as

$$\boldsymbol{S} = 2W_{I_1}\boldsymbol{I} + 4W_{I_2}\boldsymbol{C} + 2J^2 W_{I_3}\boldsymbol{C}^{-1} \ , \tag{2.34}$$

where

$$\frac{\partial I_1}{\partial \boldsymbol{C}} = \boldsymbol{I} \ , \qquad \frac{\partial I_2}{\partial \boldsymbol{C}} = 2\boldsymbol{C} \ , \qquad \frac{\partial I_3}{\partial \boldsymbol{C}} = J^2 \boldsymbol{C}^{-1} \ . \tag{2.35}$$

The Cauchy stress tensor follows from the Piola push-forward [53, 17] of the 2nd Piola-Kirchhoff stress tensor

$$\boldsymbol{\sigma} = J^{-1}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^{\mathrm{T}} = 2J^{-1}W_{I_1}\boldsymbol{b} + 4J^{-1}W_{I_2}\boldsymbol{b}^2 + 2JW_{I_3}\boldsymbol{I} \ . \tag{2.36}$$

Furthermore, the 2nd Piola-Kirchhoff and Cauchy stress tensors can be expressed using principal stretches

$$\boldsymbol{S} = \sum_{i=1}^{3} S_i \boldsymbol{N}_i \otimes \boldsymbol{N}_i \ , \qquad S_i = 2\frac{\partial W}{\partial \lambda_i^2} \ , \tag{2.37a}$$

$$\boldsymbol{\sigma} = \sum_{i=1}^{3} \sigma_i \boldsymbol{n}_i \otimes \boldsymbol{n}_i \ , \qquad \sigma_i = \frac{\lambda_i}{J}\frac{\partial W}{\partial \lambda_i} = \frac{1}{J}\frac{\partial W}{\partial \ln \lambda_i} \ . \tag{2.37b}$$

Here, S_i and σ_i are the principal components of the respective tensors.

2.3.2.2 Elasticity tensors from strain energy functions

The elasticity tensor in the initial configuration, also called material elasticity tensor, is defined as

$$\boldsymbol{\mathcal{C}} = 2\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{C}} = 4\frac{\partial^2 W}{\partial \boldsymbol{C} \partial \boldsymbol{C}} \ . \tag{2.38}$$

Applying the Piola push-forward to the material elasticity tensor leads to the spatial elasticity tensor [17],

$$\boldsymbol{c} = \sum_{\substack{i,j,k,l=1 \\ I,J,K,L=1}} J^{-1} F_{iI} F_{jJ} F_{kK} F_{lL} \mathcal{C}_{IJKL} \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \otimes \boldsymbol{e}_l \ . \tag{2.39}$$

Again, both elasticity tensors can also be formulated in terms of principal stretches as

$$\boldsymbol{\mathcal{C}} = \sum_{i,j=1}^{3} 4\frac{\partial^2 W}{\partial \lambda_i^2 \partial \lambda_j^2} \boldsymbol{\mathcal{N}}_{iijj} + \sum_{\substack{i,j=1 \\ i \neq j}}^{3} \frac{S_i - S_j}{\lambda_i^2 - \lambda_j^2} \left(\boldsymbol{\mathcal{N}}_{ijij} + \boldsymbol{\mathcal{N}}_{ijji} \right) \ , \tag{2.40a}$$

$$\boldsymbol{c} = \sum_{i,j=1}^{3} \frac{1}{J}\frac{\partial^2 W}{\partial \ln \lambda_i \partial \ln \lambda_j} \boldsymbol{\eta}_{iijj} + \sum_{\substack{i,j=1 \\ i \neq j}}^{3} \frac{\sigma_i \lambda_j^2 - \sigma_j \lambda_i^2}{\lambda_i^2 - \lambda_j^2} \left(\boldsymbol{\eta}_{ijij} + \boldsymbol{\eta}_{ijji} \right) - \sum_{i=1}^{3} 2\sigma_i \boldsymbol{\eta}_{iiii} \tag{2.40b}$$

with

$$\boldsymbol{\mathcal{N}}_{ijkl} = \boldsymbol{N}_i \otimes \boldsymbol{N}_j \otimes \boldsymbol{N}_k \otimes \boldsymbol{N}_l \tag{2.41a}$$

$$\boldsymbol{\eta}_{ijkl} = \boldsymbol{n}_i \otimes \boldsymbol{n}_j \otimes \boldsymbol{n}_k \otimes \boldsymbol{n}_l \ . \tag{2.41b}$$

2.3.2.3 Some common hyperelastic formulations

For completeness, this section recalls two widely used hyperelastic material formulations which will also be applied in the numerical examples in the following chapters.

Neo-Hookean material

One of the most commonly used hyperelastic materials is the Neo-Hookean formulation. Its strain energy function can be defined as [17]

$$W^{\mathrm{NH}} = \frac{\mathrm{G}}{2}\left(\operatorname{tr}\hat{\mathbf{b}} - 3\right) + \frac{\mathrm{K}}{2}\left(J-1\right)^2 \ , \tag{2.42}$$

where K is the initial bulk modulus defined as [17]

$$\mathrm{K} = \frac{E}{3\left(1-2\nu\right)} \ , \tag{2.43}$$

and $\hat{\mathbf{b}}$ denotes the deviatoric part of the left Cauchy-Green tensor. The Cauchy stress tensor for the neo-Hookean formulation reads

$$\boldsymbol{\sigma}^{\mathrm{NH}} = \mathrm{G}J^{-5/3}\left(\mathbf{b} - \tfrac{1}{3}I_1\mathbf{I}\right) + \mathrm{K}\left(J-1\right)\mathbf{I} \ . \tag{2.44}$$

Hencky material based on logarithmic principal stretches

The Hencky material is an example of a hyperelastic material formulated in terms of principal stretches. It can be used for the simulation of rubber materials undergoing very large deformations. The strain energy function for the Hencky material is defined as

$$W^{\mathrm{H}}\left(\lambda_1,\lambda_2,\lambda_3\right) = \mathrm{G}\sum_{i=1}^{3}\left(\ln\lambda_i\right)^2 + \frac{\Lambda}{2}\left(\ln J\right)^2 \ , \tag{2.45}$$

where λ_i are the principal stretches, and G and Λ are the Lamé parameters. The principal stresses needed to compute the Cauchy stress tensor (Equation (2.37)) follow as

$$\sigma_i = \frac{c}{J}\frac{\partial W_H}{\partial \ln\lambda_i} = \frac{c}{J}\left(2\mathrm{G}\ln\lambda_i + \Lambda\ln J\right) \ . \tag{2.46}$$

2.4 Variational formulation

The previous sections introduced the fundamental relations to describe elastic strains and stresses as functions of deformation. Deformation, however, is generally an unknown quantity that needs to be determined given a set of boundary conditions. This leads to a so-called boundary value problem. In most cases, the problem geometry or the boundary conditions are too complex, to find a closed form solution. In this case, we seek to find an approximate solution using the finite element method (FEM), which can be considered as state of the art in the field of solid mechanics [120, 56, 145, 9]. To apply a finite element discretization the strong form of equilibrium, Equation (2.26), needs to be transferred to its weak counterpart. This can be achieved by using the weighted residuals approach, the principle of virtual work or the principle of stationary potential energy. If nonlinearities are present in the kinematics, constitutive relations or boundary conditions, the boundary value problem can be solved iteratively using the Newton-Raphson method [53, 17]. In this case, a linearization of the weak form is needed.

2.4.1 Boundary value problem

The boundary-value problem depicted in Figure 2.5 can be stated in its strong form as

$$\operatorname{div}\boldsymbol{\sigma} + \boldsymbol{f} = \boldsymbol{0} \qquad \text{in } \boldsymbol{\varphi}(\Omega) \tag{2.47a}$$

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \qquad \text{on } \boldsymbol{\varphi}(\Gamma_{\mathrm{D}}) \tag{2.47b}$$

$$\boldsymbol{t} = \boldsymbol{\sigma}\cdot\boldsymbol{n} = \bar{\boldsymbol{t}} \quad \text{on } \boldsymbol{\varphi}(\Gamma_{\mathrm{N}})\ . \tag{2.47c}$$

Here, $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{t}}$ denote prescribed displacements and tractions, respectively. The domain boundary $\partial\Omega$ is assumed to be composed of disjoint parts

$$\partial\Omega = \Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N}} \qquad \text{with} \qquad \Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{N}} = \emptyset\ , \tag{2.48}$$

where Γ_{D} and Γ_{N} denote Dirichlet and Neumann boundaries, respectively.

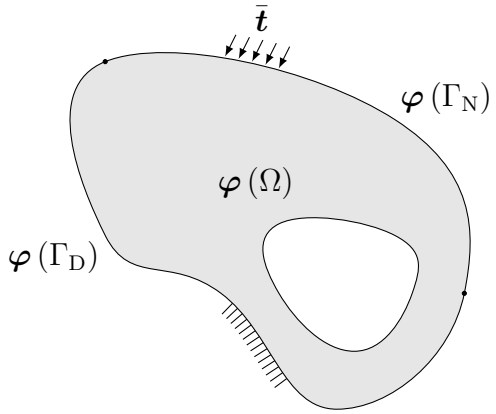


Figure 2.5: Boundary-value problem.

2.4.2 Weak form of equilibrium

To apply the finite element method the balance of linear momentum (2.26) needs to be stated in a variational (weak) form.

2.4.2.1 Principal of virtual work

One way to derive the weak form is the principal of virtual work [53]. To this end, the equilibrium equation is multiplied by a vector-valued test function $\delta\boldsymbol{u}$ and integrated over the deformed domain of the body $\boldsymbol{\varphi}(\Omega)$

$$G(\boldsymbol{u}, \delta\boldsymbol{u}) = \int_{\boldsymbol{\varphi}(\Omega)} (\operatorname{div}\boldsymbol{\sigma} + \boldsymbol{f}) \cdot \delta\boldsymbol{u}\, \mathrm{d}v = 0\ . \tag{2.49}$$

With the help of the product rule

$$\operatorname{div}\boldsymbol{\sigma}\cdot\delta\boldsymbol{u} = \operatorname{div}(\boldsymbol{\sigma}\delta\boldsymbol{u}) - \boldsymbol{\sigma} : \operatorname{grad}\delta\boldsymbol{u} \tag{2.50}$$

and the divergence theorem

$$\int_{\varphi(\Omega)} \operatorname{div}(\boldsymbol{\sigma}\delta\boldsymbol{u})\, \mathrm{d}v = \int_{\varphi(\partial\Omega)} \boldsymbol{\sigma}\delta\boldsymbol{u}\cdot\boldsymbol{n}\, \mathrm{d}a \tag{2.51}$$

the variational problem follows as

$$G(\boldsymbol{u},\delta\boldsymbol{u}) = \int_{\varphi(\Omega)} \boldsymbol{\sigma} : \operatorname{grad}\delta\boldsymbol{u}\, \mathrm{d}v - \int_{\varphi(\Omega)} \boldsymbol{f}\cdot\delta\boldsymbol{u}\, \mathrm{d}v - \int_{\varphi(\partial\Omega)} \boldsymbol{\sigma}\delta\boldsymbol{u}\cdot\boldsymbol{n}\, \mathrm{d}a = 0\ . \tag{2.52}$$

Note, that this operation effectively reduces the continuity requirements on the approximate solution. The test function $\delta\boldsymbol{u}$ is required to vanish on the Dirichlet boundary Γ_{D}, so that only the Neumann part remains in the last part of Equation (2.52)

$$G(\boldsymbol{u},\delta\boldsymbol{u}) = \int_{\varphi(\Omega)} \boldsymbol{\sigma} : \operatorname{grad}\delta\boldsymbol{u}\, \mathrm{d}v - \int_{\varphi(\Omega)} \boldsymbol{f}\cdot\delta\boldsymbol{u}\, \mathrm{d}v - \int_{\varphi(\Gamma_{\mathrm{N}})} \bar{\boldsymbol{t}}\cdot\delta\boldsymbol{u}\, \mathrm{d}a = 0\ , \tag{2.53}$$

where $\bar{\boldsymbol{t}}$ is the prescribed traction introduced in Equation (7.3). To arrive at the final form of the principal of virtual work in the spatial description (deformed configuration), we express the double contraction in the first integral of the above equation in terms of the variation of the Euler-Almansi strain tensor

$$\boldsymbol{\sigma} : \operatorname{grad}\delta\boldsymbol{u} = \boldsymbol{\sigma} : \delta\boldsymbol{e} \tag{2.54}$$

with

$$\delta\boldsymbol{e} = \frac{1}{2}\left[(\operatorname{grad}\delta\boldsymbol{u})^{\mathrm{T}} + \operatorname{grad}\delta\boldsymbol{u}\right] \qquad \text{and} \qquad \boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathrm{T}}\ . \tag{2.55}$$

Thus, the variational form of the balance equation follows as

$$G(\boldsymbol{u},\delta\boldsymbol{u}) = \int_{\varphi(\Omega)} \boldsymbol{\sigma} : \delta\boldsymbol{e}\, \mathrm{d}v - \int_{\varphi(\Omega)} \boldsymbol{f}\cdot\delta\boldsymbol{u}\, \mathrm{d}v - \int_{\varphi(\Gamma_{\mathrm{N}})} \bar{\boldsymbol{t}}\cdot\delta\boldsymbol{u}\, \mathrm{d}a = 0\ . \tag{2.56}$$

The two functions

$$\delta W_{\mathrm{int}}(\boldsymbol{u},\delta\boldsymbol{u}) = \int_{\varphi(\Omega)} \boldsymbol{\sigma} : \delta\boldsymbol{e}\, \mathrm{d}v \quad \text{and} \tag{2.57a}$$

$$\delta W_{\mathrm{ext}}(\boldsymbol{u},\delta\boldsymbol{u}) = \int_{\varphi(\Omega)} \boldsymbol{f}\cdot\delta\boldsymbol{u}\, \mathrm{d}v + \int_{\varphi(\Gamma_{\mathrm{N}})} \bar{\boldsymbol{t}}\cdot\delta\boldsymbol{u}\, \mathrm{d}a\ . \tag{2.57b}$$

are called the *internal* virtual work δW_{int} and *external* virtual work δW_{ext}, respectively [53, 17].

2.4.2.2 Principal of stationary potential energy

As shown in Section 2.3.2, hyperelastic materials are based on the assumption that the elastic behavior of a deformable body can be described by a strain energy function W. The stored elastic energy or elastic potential inside such a body is then expressed as [53]

$$\Pi_{\text{int}}(\boldsymbol{u}) = \int_{\Omega} W(F(\boldsymbol{u}))\, \mathrm{d}V \ . \tag{2.58}$$

For a conservative problem, i.e. where loads do not depend on the deformation [53, 17], the external potential energy reads

$$\Pi_{\text{ext}}(\boldsymbol{u}) = -\int_{\Omega} \boldsymbol{F} \cdot \delta\boldsymbol{u}\, \mathrm{d}V - \int_{\partial\Omega} \boldsymbol{T} \cdot \delta\boldsymbol{u}\, \mathrm{d}A \ . \tag{2.59}$$

Here, $\boldsymbol{F}$ and $\bar{\boldsymbol{T}}$ denote the body force and traction vector in the reference configuration, respectively [53]. The full potential energy is given by

$$\Pi(\boldsymbol{u}) = \Pi_{\text{int}}(\boldsymbol{u}) + \Pi_{\text{ext}}(\boldsymbol{u}) \ . \tag{2.60}$$

Constraints such as Dirichlet conditions or frictionless contact conditions can also be formulated in terms of constraint potentials, as will be shown in the following chapters. Accordingly, the weak forms of such constraint potentials may be obtained using the principle of stationary potential energy. To arrive at a solution, the potentials need to be minimized with respect to the deformation. To this end, the directional derivative [53, 17] is applied

$$\min_{\arg \boldsymbol{u}} \Pi(\boldsymbol{u}) \quad \rightarrow \quad \delta\Pi(\boldsymbol{u}, \delta\boldsymbol{u}) = D_{\delta\boldsymbol{u}}[\Pi(\boldsymbol{u})] = \frac{d}{d\varepsilon}\, \Pi(\boldsymbol{u} + \varepsilon\delta\boldsymbol{u})\big|_{\varepsilon=0} = 0 \tag{2.61}$$

The directional directive is not only convenient for the principle of stationary energy. It is also used extensively to obtain linearized quantities for the solution of nonlinear problems, as described in the next section.

2.4.3 Linearized weak form

Hyperelastic material models and contact constraints (of inequality type) render the boundary value problem given in Equation (7.3) nonlinear. An efficient scheme to solve nonlinear elasticity problems is the Newton-Raphson method. Therefore, the weak form obtained in the previous section needs to be linearized.

2.4.3.1 Newton-Raphson method

The Newton-Raphson method iteratively solves a, possibly vector valued, nonlinear function $\boldsymbol{G}$ by utilizing a Taylor expansion

$$\boldsymbol{G}(\boldsymbol{u}_{ex}) \approx \boldsymbol{G}\left(\boldsymbol{u}^i\right) + D_{\Delta\boldsymbol{u}}\boldsymbol{G}\left(\boldsymbol{u}^i\right) + \ldots = \boldsymbol{0} \ , \tag{2.62}$$

where the directional derivative for an intermediate solution $\boldsymbol{u}^i$ at iteration i is defined as

$$D_{\Delta\boldsymbol{u}}\boldsymbol{G}\left(\boldsymbol{u}^i\right) = \frac{d}{d\varepsilon}\, \boldsymbol{G}\left(\boldsymbol{u}^i + \varepsilon\Delta\boldsymbol{u}^i\right)\big|_{\varepsilon=0} \ . \tag{2.63}$$

The Taylor expansion is truncated after the second term leading to the equation

$$D_{\Delta \boldsymbol{u}} \boldsymbol{G}\left(\boldsymbol{u}^{\boldsymbol{i}}\right) = -\boldsymbol{G}\left(\boldsymbol{u}^{i}\right) \ , \tag{2.64}$$

which is solved repeatedly until a convergence criterion is fulfilled. An important characteristic of the Newton-Raphson method is its quadratic convergence close to the solution point $\boldsymbol{u}_{ex}$ [10].

2.4.3.2 Linearized internal virtual work

As opposed to pressure loads, which are not in the focus of this work, and contact conditions, which will be addressed in Chapter 6, the external virtual work (2.57b) does not depend on the deformation. Therefore, we will now focus on the linearization of the internal virtual work

$$\delta W_{\text{int}}\left(\boldsymbol{u}, \delta \boldsymbol{u}\right) = \int\limits_{\boldsymbol{\varphi}(\Omega)} \boldsymbol{\sigma}\left(\boldsymbol{u}\right) : \delta \boldsymbol{e}\left(\boldsymbol{u}\right) \, \mathrm{d}v = \int\limits_{\Omega} \boldsymbol{S}\left(\boldsymbol{E}\left(\boldsymbol{u}\right)\right) : \delta \boldsymbol{E}\left(\boldsymbol{u}\right) \, \mathrm{d}V \ . \tag{2.65}$$

To arrive at the linearization of Equation (2.57a), the internal virtual work is first linearized in the material (undeformed) configuration and then pushed to the deformed configuration

$$D_{\Delta \boldsymbol{u}} \delta W_{\text{int}}\left(\boldsymbol{u}, \delta \boldsymbol{u}, \Delta \boldsymbol{u}\right) = \int\limits_{\Omega} \left(\boldsymbol{S} : D_{\Delta \boldsymbol{u}} \delta \boldsymbol{E} + \delta \boldsymbol{E} : D_{\Delta \boldsymbol{u}} \boldsymbol{S}\right) \, \mathrm{d}V \tag{2.66}$$

$$= \int\limits_{\boldsymbol{\varphi}(\Omega)} \left(\operatorname{grad} \delta \boldsymbol{u} : \operatorname{grad} \Delta \boldsymbol{u} \boldsymbol{\sigma} + \operatorname{grad} \delta \boldsymbol{u} : \boldsymbol{c} : \operatorname{grad} \Delta \boldsymbol{u}\right) \, \mathrm{d}v \ . \tag{2.67}$$

This greatly simplifies the linearization, since in the material configuration the domain boundaries do not depend on the deformation [17].

Chapter 3

Contact mechanics for frictionless problems

This chapter gives a short overview of the field of computational contact mechanics with a focus on frictionless problems. Contact problems are nonlinear by nature since the final configuration of the contact interface is not known beforehand. An exception are tied contact conditions applied to linear elastic bodies [93, 103]. However, this thesis deals with normal (frictionless) contact constraints, which are of inequality type. While there is a considerable amount of closed form solutions for contact problems involving simplified geometries (e.g. [51] and [41]) the vast majority of problems of engineering interest can only be solved approximately. Such approximations are generally obtained using the finite element method. Classically, contact is enforced along contact elements assigned to the surface of the bulk discretization of the bodies under consideration. Most of the available contact element formulations can be assigned to one of the three categories: the *node-to-node*, *point-to-segment* and *segment-to-segment* formulations [71, 133, 64]. While contact can occur between numerous bodies, it is normally treated by considering unique pairs of contact bodies. For every pair a master and a slave side are identified, where the numerical integration of the contact contributions is performed on the slave side [133]. In most cases the final contact interface is not known in advance. Therefore, potential contact element (*master-slave*) pairs need to be identified using a global search. In a second step, discrete points on the slave surface are projected onto the corresponding master surface. The positions and number of these points depend on the choice of the contact element discretization. The result of the projections is the (discrete) minimum distance–or *gap*–between the contacting bodies. This stage is also called local search or closest point projection [66]. Contact is detected whenever the gap becomes less or equal to zero. In this case contact constraints need to be enforced by a suitable method. Possible constraint formulations include Lagrange multipliers, the penalty method, the barrier method, Nitsche's method and combinations thereof [71, 10, 133, 138].

3.1 Contact kinematics for normal contact

The fundamental measure of normal (frictionless) contact is the minimum distance–or normal gap g_{N}–between the contacting bodies. Using the normal gap, we can express the non-

penetration condition

$$g_{\mathrm{N}} = (\boldsymbol{x}_1 - \boldsymbol{x}_2) \cdot \boldsymbol{n}_2 \geq 0 \ . \tag{3.1}$$

Here, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ denote current coordinates of a pair of contacting bodies (Figure 3.1). The normal vector $\boldsymbol{n}_2$ is defined on the contact boundary Γ_2 of the second body.

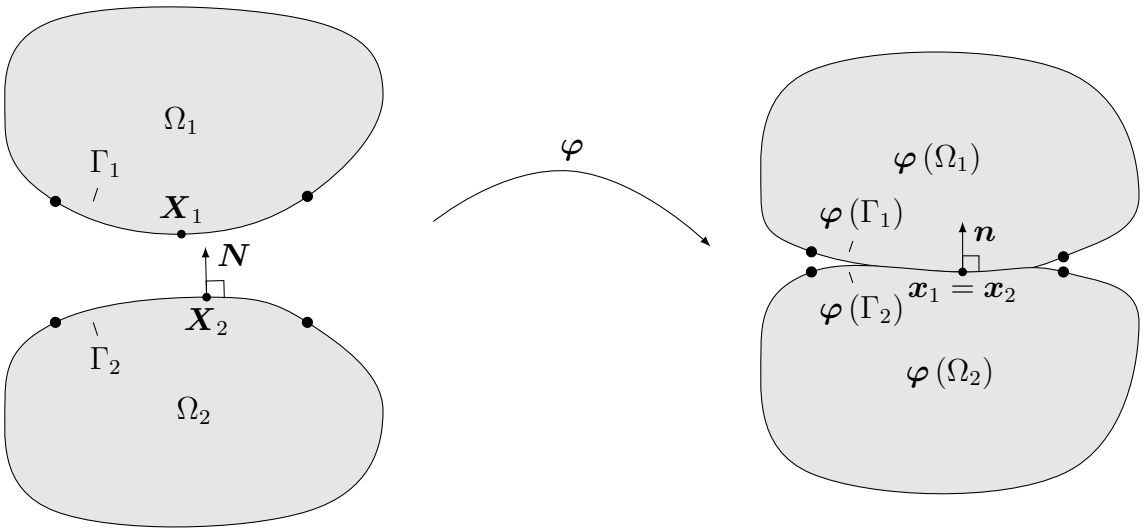


Figure 3.1: Two bodies coming into contact. Left: Initial configuration. Right: Deformed configuration.

3.1.1 Normal gap function

The normal gap g_{N} is computed by performing a closest point projection from a point $\boldsymbol{x}_1$ on the surface of body 1 onto the surface Γ_2 of the associated body 2 [133, 64]. The aim of the closest point projection is to find the local (convective) coordinates $\boldsymbol{\xi}_2$ of a point $\bar{\boldsymbol{x}}_2(\boldsymbol{\xi}_2)$ on Γ_2, which minimizes the distance to $\boldsymbol{x}_1$

$$||\boldsymbol{x}_1 - \bar{\boldsymbol{x}}_2(\boldsymbol{\xi}_2)|| \rightarrow \min_{\boldsymbol{\xi}_2 \in \Gamma_2} \ . \tag{3.2}$$

The resulting point $\bar{\boldsymbol{x}}_2$ and its associated normal vector $\bar{\boldsymbol{n}}_2$ are then used to obtain the normal gap as follows

$$g_{\mathrm{N}} = [\boldsymbol{x}_1 - \bar{\boldsymbol{x}}_2(\boldsymbol{\xi}_2)] \cdot \bar{\boldsymbol{n}}_2(\boldsymbol{\xi}_2) \ . \tag{3.3}$$

The normal vector can be computed using the tangent vectors $\bar{\boldsymbol{x}}_{2,(\cdot)}$ at the point $\bar{\boldsymbol{x}}_2$, as depicted in Figure 3.2

$$\bar{\boldsymbol{n}}_2(\boldsymbol{\xi}_2) = \frac{\bar{\boldsymbol{x}}_{2,1}(\boldsymbol{\xi}_2) \times \bar{\boldsymbol{x}}_{2,2}(\boldsymbol{\xi}_2)}{\|\bar{\boldsymbol{x}}_{2,1}(\boldsymbol{\xi}_2) \times \bar{\boldsymbol{x}}_{2,2}(\boldsymbol{\xi}_2)\|} \ . \tag{3.4}$$

The closest point projection and its numerical realization within this thesis is discussed in more detail in Chapter 6.

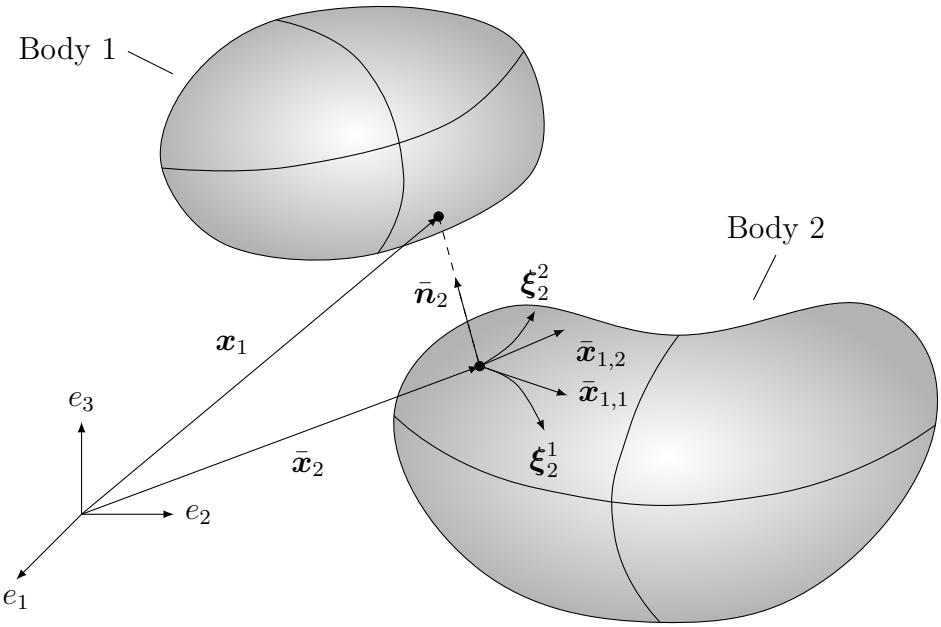


Figure 3.2: Closest point projection. The point $\boldsymbol{x}_1$ is projected onto the surface of body 2 to find $\bar{\boldsymbol{x}}_2$ and its associated local basis.

3.1.2 Variation of the normal gap

In order to solve frictionless contact problems using a finite element formulation, the variation of the gap g_N is needed [133]

$$\begin{aligned}\delta g_\mathrm{N} &= D_{\delta \boldsymbol{u}} \left\{ [\boldsymbol{x}_1 - \bar{\boldsymbol{x}}_2 (\boldsymbol{\xi}_2)] \cdot \bar{\boldsymbol{n}}_2 (\boldsymbol{\xi}_2) \right\} \\ &= \left[\delta \boldsymbol{u}_1 - \delta \boldsymbol{u}_2 - \bar{\boldsymbol{x}}_{2,i} \delta \xi_2^i \right] \cdot \bar{\boldsymbol{n}}_2 + [\boldsymbol{x}_1 - \bar{\boldsymbol{x}}_2] \cdot \delta \bar{\boldsymbol{n}}_2 \ . \end{aligned} \tag{3.5}$$

The first term of Equation (3.5) can be simplified by noting that the dot product of the normal and any of its tangential vectors is zero

$$\boldsymbol{n}_2 \cdot \boldsymbol{x}_{2,i} = 0 \ . \tag{3.6}$$

Furthermore, using Equation (3.4) and the fact that a unit vector and its derivative are orthogonal (see [133, 64])

$$\boldsymbol{n} \cdot \delta \boldsymbol{n} = 0 \ , \tag{3.7}$$

it follows that the second term of Equation (3.5) completely vanishes. Hence the final statement for the variation of the normal gap reads

$$\delta g_\mathrm{N} = [\delta \boldsymbol{u}_1 - \delta \bar{\boldsymbol{u}}_2] \cdot \bar{\boldsymbol{n}}_2 \ . \tag{3.8}$$

Note, that to linearize the variation of the gap one has to start from Equation (3.5), since the terms that have vanished may still have non-zero tangents.

3.2 Constraint conditions

Frictionless contact problems can be classified as optimization problems with inequality constraints. For an exact mathematical treatment, we resort to the Karush-Kuhn-Tucker (KKT) conditions for normal contact, known from optimization theory [74]. The constraints, also called Hertz-Signorini-Moreau conditions in the solid mechanics community [133], read as follows

$$g_{\mathrm{N}} \geq 0 \qquad \text{No penetration} \tag{3.9a}$$

$$t_{\mathrm{N}} \leq 0 \qquad \text{Compression only} \tag{3.9b}$$

$$g_{\mathrm{N}} t_{\mathrm{N}} = 0 \qquad \text{Complementary condition .} \tag{3.9c}$$

The KKT conditions state that bodies shall not interpenetrate, i.e. the normal gap g_{N} must not become negative. Also, only compressive normal tractions (pressures) t_{N} shall occur at the interface. While Equation (3.9a) holds for all contact phenomena, Equation (3.9b) follows from the frictionless contact model used here. Different other constitutive models are available for the normal contact traction, which can account for e.g. adhesive and cohesive effects ([133]). To obtain numerical solutions, problems containing inequality constraints like the KKT conditions are usually transformed into a series of simpler problems with only equality constraints [133].

3.3 Contact weak form

The weak form for a pair of elastic bodies coming into contact can be stated as

$$G_{\mathrm{C}}\left(\boldsymbol{u}, \delta\boldsymbol{u}\right) = \sum_{i=1}^{2} \left[G_i\left(\boldsymbol{u}_i, \delta\boldsymbol{u}_i\right) - \delta W_{i,\mathrm{C}}\left(\boldsymbol{u}_i, \delta\boldsymbol{u}_i\right) \right] \ , \tag{3.10}$$

where G_i is the standard weak form of body i given in Equation (2.53). $\delta W_{i,\mathrm{C}}$ denotes the virtual contact work for body i. The total virtual contact work reads

$$\begin{aligned} \delta W_{\mathrm{C}} &= \delta W_{1,\mathrm{C}}\left(\boldsymbol{u}_1, \delta\boldsymbol{u}_1\right) + \delta W_{2,\mathrm{C}}\left(\boldsymbol{u}_2, \delta\boldsymbol{u}_2\right) \\ &= \int\limits_{\boldsymbol{\varphi}\left(\Gamma_{1,\mathrm{C}}\right)} \boldsymbol{t}_1 \cdot \delta\boldsymbol{u}_1 \, \mathrm{d}a_1 + \int\limits_{\boldsymbol{\varphi}\left(\Gamma_{2,\mathrm{C}}\right)} \boldsymbol{t}_2 \cdot \delta\boldsymbol{u}_2 \, \mathrm{d}a_2 \ , \end{aligned} \tag{3.11}$$

where $\boldsymbol{t}_i$ and $\mathrm{d}a_i$ denote the contact traction and an infinitely small area segment of the boundary of body i, respectively. Using the local equilibrium between the two bodies

$$\boldsymbol{t}_1 \, \mathrm{d}a_1 = -\boldsymbol{t}_2 \, \mathrm{d}a_2 \ , \tag{3.12}$$

the virtual contact work can be written compactly as

$$\delta W_{\mathrm{C}} = \int\limits_{\boldsymbol{\varphi}(\Gamma_{\mathrm{C}})} \boldsymbol{t} \cdot \left(\delta\boldsymbol{u}_{\mathrm{S}} - \delta\boldsymbol{u}_{\mathrm{M}}\right) \, \mathrm{d}a \ . \tag{3.13}$$

At this point, the contact integral is performed only over the boundary of body 1, in the following denoted as the *slave* boundary. Accordingly, the surface of body 2 will be denoted as the *master* surface. The contact traction $\boldsymbol{t}$ can be decomposed into its normal and tangential components

$$\boldsymbol{t} = t_{\mathrm{N}}\boldsymbol{n}_{\mathrm{M}} + t_{\mathrm{T}}^{j}\boldsymbol{x}_{\mathrm{M},j} \ . \tag{3.14}$$

Since the tangential tractions are zero $t_{\mathrm{T}}^{i} = 0$ for frictionless contact, the virtual contact work follows as

$$\delta W_{\mathrm{C}} = \int\limits_{\boldsymbol{\varphi}(\Gamma_S)} t_{\mathrm{N}}\boldsymbol{n}_{\mathrm{M}} \cdot (\delta\boldsymbol{u}_{\mathrm{S}} - \delta\boldsymbol{u}_{\mathrm{M}}) \, \mathrm{d}a \ . \tag{3.15}$$

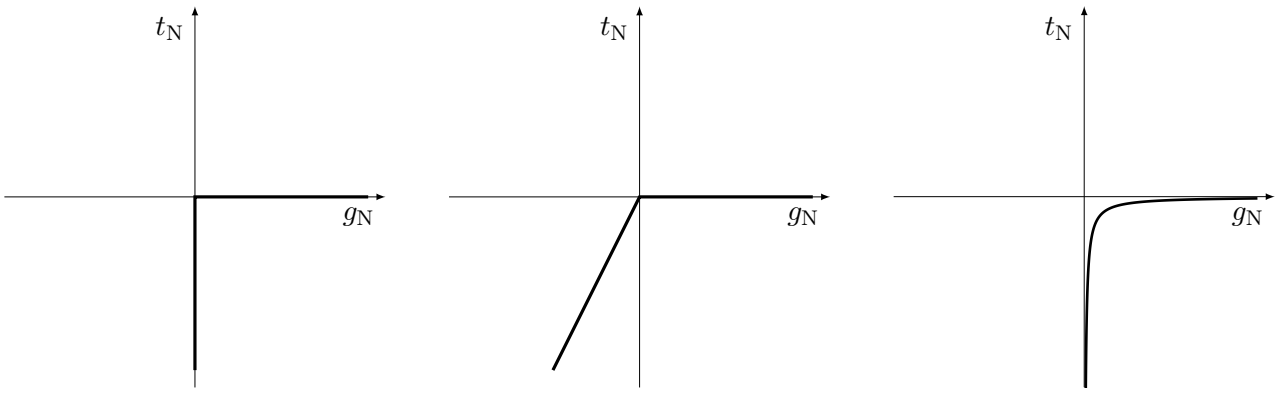
In the case of hyperelasticity, the virtual contact work can also be obtained by minimization of a constraint potential

$$\delta W_{\mathrm{C}} = D_{\delta\boldsymbol{u}}\Pi_{\mathrm{C}} = \delta\Pi_{\mathrm{C}} \ . \tag{3.16}$$

The following section will outline some of the most common constraint formulations in potential description.

3.4 Constraint formulations

Many formulations are available to enforce contact for the frictionless case. Their choice is generally governed by their achievable accuracy, (nonlinear) solution behavior and implementational complexity. The following sections outline some of the most established constraint formulations. Figure 3.3 depicts the relation between normal gap g_{N} and normal traction t_{N} for some of these formulations.



(a) Lagrange Multiplier **(b)** Penalty Method **(c)** Barrier Method

Figure 3.3: Schematic comparison of different methods to enforce contact constraints.

3.4.1 Lagrange multipliers

Most contact formulations can be derived from the method of Lagrange multipliers (LMs) [133], which enforce the KKT conditions directly (Figure 3.3a). To this end, an additional field variable λ_N, the Lagrange multiplier, is introduced to form the constraining potential for frictionless contact

$$\Pi_\mathrm{C}^\mathrm{LM} = \int_{\Gamma_\mathrm{C}} \lambda_\mathrm{N} g_\mathrm{N} \,\mathrm{d}A \ . \tag{3.17}$$

Lagrange multipliers are usually applied with an *active-set* strategy. Here active multipliers are identified before and kept fixed during load step and the load step is solved until the active set does not change anymore [133]. An alternative are semi-smooth Newton methods utilized by e.g. Popp et al. [92]. The Lagrange multipliers for normal contact can be identified as the normal traction t_N introduced earlier [133]. While Lagrange multipliers enforce the KKT conditions exactly, they have several shortcomings. They introduce additional unknowns, whose number is constantly changing during contact iterations (*active set*). This makes LMs harder to handle in a numerical implementation. Furthermore, LMs lead to a saddle-point structure of the linear system of equations, which restricts the choice of numerical solvers. Finally, due to their exact fulfilment of the non-regular KKT conditions, LMs behave comparatively stiff, which sometimes leads to solution problems [133]. An interesting alternative to standard Lagrange multipliers are bi-orthogonal multiplier spaces introduced by Wohlmuth et al. [132, 131] and further developed by Popp et al. [94]. The spaces allow the condensation of the Lagrange multiplier from the system of equations, reaping the benefits of LMs while getting rid of most of its shortcomings. Unfortunately, their formulation for finite elements of arbitrary order within a fictitious domain setting, as used in this thesis, is not straightforward and computationally expensive.

3.4.2 Penalty method

The most frequently used alternative to Lagrange multipliers is the penalty method, which does not lead to additional unknowns and allows for a much simpler implementation. To this end, the penalty method applies a regularization of the multipliers reading

$$\lambda_\mathrm{N} = \frac{1}{2} \epsilon_\mathrm{N} g_\mathrm{N} \ , \tag{3.18}$$

leading to

$$\Pi_\mathrm{C}^\mathrm{P} = \frac{1}{2} \int_{\Gamma_\mathrm{C}} \epsilon_\mathrm{N} H\left(-g_\mathrm{N}\right) g_\mathrm{N}^2 \,\mathrm{d}A \ . \tag{3.19}$$

Here, ϵ_N is the so-called penalty parameter for normal contact and H is the Heaviside function

$$H\left(x\right) = \begin{cases} 0.0 & \text{if } x < 0 \\ 1.0 & \text{if } x \geq 0 \ . \end{cases} \tag{3.20}$$

The Heaviside function takes care that only regions where penetration occurs contribute to the contact constraint. Conceptually, the penalty method can be understood as a system of distributed springs with a stiffness ϵ_N that only act once bodies interpenetrate. Consequently, the penalty method leaves non-physical residual penetrations. Luenberger et al. [74] show that the original solution is only obtained in the limit case when the penalty parameter ϵ_N tends to infinity. However, too large penalty parameters may lead to ill-conditioned systems and failure of the nonlinear solution procedure. Nonetheless, due to the regularization (Figure 3.3b) the penalty method generally behaves more robustly during Newton iterations than standard LMs [133]. Also note that Equation (3.19) is the simplest penalty formulation for normal contact. Several alternatives can be found e.g. in the monograph by Yastrebov [138].

3.4.3 Augmented methods

Augmented (Lagrange) methods have been developed to preserve the simplicity of the penalty method, while alleviating its shortcomings [133]. In its basic form the constraint formulation reads

$$\Pi_\mathrm{C}^\mathrm{AL} = \int_{\Gamma_\mathrm{C}} \left[\bar{\lambda}_\mathrm{N} g_\mathrm{N} \; + \frac{1}{2} \epsilon_\mathrm{N} H\left(-g_\mathrm{N}\right) g_\mathrm{N}^2 \right] \mathrm{d}A \; . \tag{3.21}$$

Here, the Lagrange multipliers $\bar{\lambda}_\mathrm{N}$ are augmented forces with do not introduce additional unknowns. They are computed via an augmentation procedure like the Uzawa algorithm, see for instance [10, 133]. Thus, the Lagrange multipliers are treated as history variables. The basic idea of the Uzawa algorithm is to fix the augmented forces $\bar{\lambda}_\mathrm{N}$ during Newton iterations and update them in an outer loop over augmentations a according to the formula

$$\bar{\lambda}_\mathrm{N}^{a+1} = \bar{\lambda}_\mathrm{N}^{a} + \epsilon_\mathrm{N} g_\mathrm{N} \tag{3.22}$$

until the maximum penetration is below a given threshold. This approach allows for a relatively small penalty parameter to avoid ill-conditioning and preserves the convergence characteristics of the Newton method. However, a significant number of augmentations is generally necessary to minimize the penetration to the desired level. Therefore, Zavarise et al. presented update procedures [142, 141] that have better convergence characteristics than the original Uzawa scheme Equation (3.22).
Another method, which is closely related to augmented approaches is the *shifted penalty method* introduced by Zavarise [143]. Here, the author introduces a history variable–the shift–which is added to the local gap. Similar to the Uzawa algorithm, the shift is fixed during Newton iterations and then updated in an outer loop. The update formula is derived based on a minimization of the gap function (penetration).

3.4.4 Barrier method

Barrier or interior penalty methods account for another group of algorithms [133], which modify the original problem (3.17) to avoid additional unknowns for the contact constraint

$$\Pi_\mathrm{C}^\mathrm{B} = \int_{\Gamma_\mathrm{C}} \epsilon_\mathrm{B} H\left(-g_\mathrm{N}\right) b\left(g_\mathrm{N}\right) \, \mathrm{d}A \; , \tag{3.23}$$

where ϵ_B is the barrier parameter and $b\left(g_\mathrm{N}\right)$ is the barrier function. Different formulations are possible for $b\left(g_\mathrm{N}\right)$ [133] such as

$$b\left(g_\mathrm{N}\right) = -\frac{1}{g_\mathrm{N}} \tag{3.24}$$

$$b\left(g_\mathrm{N}\right) = -\ln\left[\min\left\{1, -g_\mathrm{N}\right\}\right] \ . \tag{3.25}$$

Unlike the penalty method, barrier methods do not allow any penetration. Hence, the solution has to be feasible at all times (Figure 3.3c). This poses some challenges for the nonlinear solution procedure since special care has to be taken that no penetrations are present at the beginning of a simulation, like e.g. in a press fit problem. Also, it has to be made sure that no penetration occurs during the Newton iterations. Similar to the penalty method the exact solution is only reached for an infinite barrier parameter ϵ_B. Hence, ill-conditioning can become a problem for large parameter values. However, barrier methods can also be cast into an augmented formulation to reduce ill-conditioning and reduce the final gap as shown by Kloosterman [61].
At this point the cross-constraint method introduced by Zavarise and Wriggers [144] deserves mention. Here, the authors combine penalty and barrier methods to allow for a smoother transition for the contact state, leading to a more robust penalty scheme. This improvement comes at the cost of an increased number of parameters, which have to be set by the user.

3.4.5 Nitsche's method

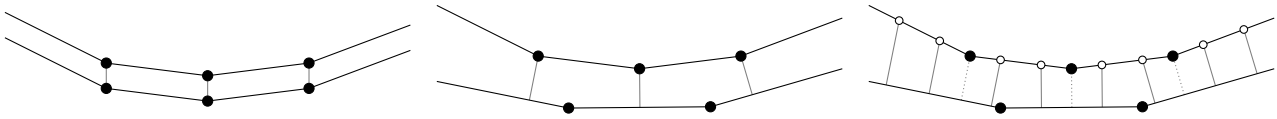
Nitsche's method has received a lot of attention for the enforcement of Dirichlet boundaries conditions [48, 102], domain decomposition (mesh tying) [103] as well as contact mechanics [136, 105, 22, 31]. The basic idea is to start from a standard Lagrange multiplier formulation (3.17) and identify the multipliers with normal tractions

$$\Pi_\mathrm{C}^\mathrm{NI} = \int_{\Gamma_\mathrm{C}} \frac{1}{2}\left[-\left(t_\mathrm{N}^1 + t_\mathrm{N}^2\right) g_\mathrm{N} + \epsilon_\mathrm{N} g_\mathrm{N}\right] \mathrm{d}A \ . \tag{3.26}$$

Since the first term in Equation (3.26) is subtracted from the final bi-linear form, a penalty term is introduced to preserve coercivity [133, 103]. The normal contact tractions occurring on the pair of considered contact bodies are denoted by t_N^1 and t_N^2, respectively. Nitsche's method does not introduce additional unknowns, as the pressures are formulated in terms of the primary solution. However, the accuracy of the final solution depends on the choice of penalty parameter. Furthermore, for geometrically nonlinear problems the stress tensor needs to be linearized. This has to be done for every employed material formulation and thus poses a significant effort for implementation.

3.5 Contact discretizations

Most discrete contact formulations can be classified into one of the following three cases. Early discretizations were based on so-called *node-to-node* contact elements (Figure 3.4a), for which the nodal positions of adjacent bodies had to match at the interface [35, 86, 60]. Hence, these formulations are also only applicable for small deformations.



(a) Node-to-node **(b)** Point-to-segment **(c)** Segment-to-segment

Figure 3.4: Comparison of different contact discretizations.

Because of the limited applicability of this approach, *node-to-segment* (or more general *point-to-segment*) contact elements were developed (Figure 3.4b) allowing the nodes to slide along the whole contact interface [47, 135]. However, non-smooth surfaces, which typically arise in linear FE meshes for the h-version, pose a challenge for this kind of formulation. For problems involving large sliding, nodes can get stuck at re-entrant corners. To overcome this issue, *segment-to-segment* (Figure 3.4c) and mortar-type methods have been developed, which enforce contact constraints in a weak (integral) sense [130, 131, 96, 95, 33, 34, 91, 93].

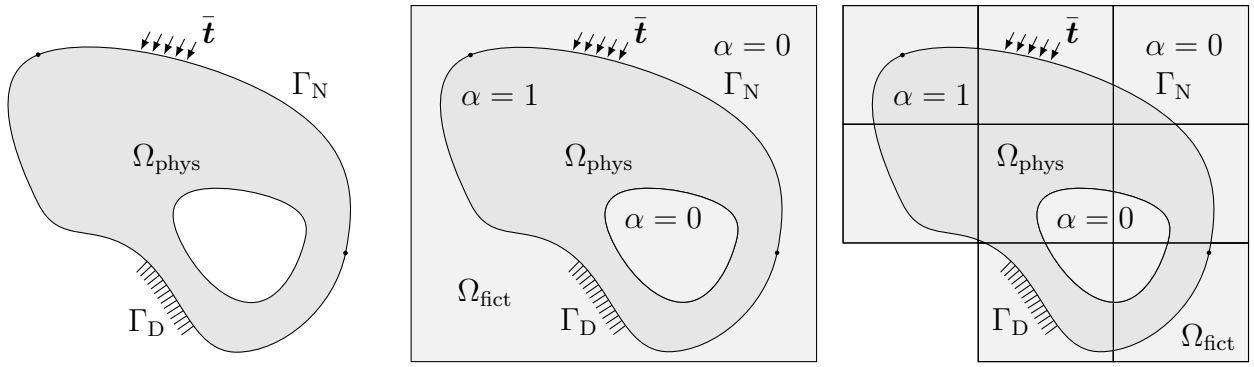
Chapter 4

Finite cell method for nonlinear problems in elasticity

The aim of this chapter is to explain how nonlinear problems in elasticity can be solved using a variant of the finite element method (FEM), which combines the benefits of high-order finite elements with a fictitious domain approach, namely the *finite cell method* (FCM).
The FEM is state of the art for solid mechanics problems in most engineering disciplines. It is based on the weak form of the partial differential equation (PDE) under consideration. The FEM introduces a finite (discrete) space of shape functions to approximate the original solution. Hence, the accuracy of the discrete solution depends on the number of shape functions and their type. The h-version of the FEM employs linear or quadratic shape functions. It controls the discretization error by reducing the size (h) of the shape function support (element size). The p-version utilizes high-order polynomial shape functions and elevates the polynomial order to reduce discretization errors. Hence, coarser and fewer elements are necessary for the p-version. An optimal combination of both version is the hp-approach, where small low order elements are utilized in regions of low regularity (singularities, material interfaces) and large high-order elements are used where the solution is smooth. The hp-version allows for an optimal trade-off between numerical cost (number of unknowns) and achievable accuracy.
A major bottleneck in practical finite element simulations is the generation of a suitable mesh (discretization). Here, a considerable amount of time is spent on preparing the geometric (CAD) model. CAD models provided by designers often contain too many or flawed geometric features or have failures like gaps between or overlaps of surfaces. While geometric flaws often do not pose a problem for visualization purposes, they are a real challenge for most meshing algorithms. There are many reliable, automatic algorithms available to create quality h-FEM meshes. This is not the case however for the p-version. To fully exploit its potential, sophisticated algorithms and data structures, as well as engineering experience, are necessary. To the author's knowledge, there is no robust automatic meshing algorithm available for the p-version applicable to three-dimensional solid geometries. Hence, while meshing is hard for the h-version of the FEM, it is even more challenging for the p- and hp-version.
The FCM is a high-order fictitious domain approach, which avoids labor-intensive meshing by recovering the original geometry at the integration level. To this end, the FCM utilizes special quadrature schemes for discontinuous integrands. Furthermore, boundary conditions are enforced in an integral sense whenever the boundary of a body does not coincide with a mesh boundary.

4.1 Fictitious domain approach

In the last years, fictitious domain approaches have gained much momentum in the field of solid mechanics [89, 26, 126, 83, 20]. Their main idea is to extend the physical domain Ω_{phys} of an arbitrarily shaped body (Figure 4.1a) by a fictitious domain Ω_{fict}, such that the union of both domains has a much simpler shape (Figure 4.1b). This resulting domain Ω can be easily meshed by Cartesian-shaped finite elements, here called *cells* (Figure 4.1c). Note that cells which do not contain any physical material can be omitted.



(a) Physical domain **(b)** Fictitious extension **(c)** Discretization

Figure 4.1: Basic idea of the fictitious domain methods: (a) The physical domain Ω_{phys} of a boundary value problem (Section 2.4.1) is embedded (b) into a fictitious domain Ω_{fict}. (c) The resulting domain Ω can be (c) simply meshed using Cartesian finite elements.

4.1.1 Linear elasticity

To illustrate the fictitious domain concept of the FCM, we consider the weak form of equilibrium for linear elasticity [56, 9] (see also Section 2.4.2). The bi-linear form $a(\boldsymbol{u}, \delta\boldsymbol{u})$ is split into the contributions of the physical Ω_{phys} and fictitious Ω_{fict} domain, respectively,

$$a(\boldsymbol{u}, \delta\boldsymbol{u}) = \int_{\Omega_{\text{phys}}} (\boldsymbol{B}\delta\boldsymbol{u})^{\mathrm{T}} \boldsymbol{C} (\boldsymbol{B}\boldsymbol{u}) \, \mathrm{d}V + \int_{\Omega_{\text{fict}}} (\boldsymbol{B}\delta\boldsymbol{u})^{\mathrm{T}} \boldsymbol{0} (\boldsymbol{B}\boldsymbol{u}) \, \mathrm{d}V \ , \tag{4.1}$$

where $\boldsymbol{B}$ is the linear strain operator, $\boldsymbol{C}$ is the linear elasticity tensor and $\boldsymbol{u}$ and $\delta\boldsymbol{u}$ are applicable test and trial functions, respectively [56]. Using an indicator function $\alpha(\boldsymbol{X})$, the two integrals of equation (4.1) can be combined into a single term:

$$a(\boldsymbol{u}, \delta\boldsymbol{u}) = \int_{\Omega} (\boldsymbol{B}\delta\boldsymbol{u})^{\mathrm{T}} \alpha\boldsymbol{C} (\boldsymbol{B}\boldsymbol{u}) \, \mathrm{d}V \ . \tag{4.2}$$

The indicator function $\alpha(\boldsymbol{X})$ is defined as

$$\alpha(\boldsymbol{X}) = \begin{cases} 1.0 & x \in \Omega_{\text{phys}} \\ \varepsilon \ll 1.0 & x \in \Omega_{\text{fict}} \end{cases} \ . \tag{4.3}$$

Note that the value of $\alpha(\boldsymbol{X})$ inside the fictitious domain is generally not set to zero but to a very small value, to avoid severe ill-conditioning of the resulting system of equations. Typical choices lie in the range of $[10^{-12}, 10^{-6}]$ (see e.g. [26, 112]). The linear form $f(\delta\boldsymbol{u})$ can be split in the same manner, leading to

$$f(\delta\boldsymbol{u}) = \int_{\Omega} \alpha\delta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{f}\,\mathrm{d}V + \int_{\Gamma_N} \alpha\delta\boldsymbol{u}^{\mathrm{T}}\bar{\boldsymbol{t}}\,\mathrm{d}A. \tag{4.4}$$

Here $\boldsymbol{f}$ denotes a body force and $\bar{\boldsymbol{t}}$ is a prescribed boundary traction (see also Section 2.4.1).

4.1.2 Nonlinear elasticity

In a general nonlinear setting involving large strain hyperelasticity a solution to Equation (2.64) is sought

$$D_{\Delta u}\boldsymbol{G}\left(\boldsymbol{u}, \delta\boldsymbol{u}, \Delta\boldsymbol{u}\right) = \delta W_{\mathrm{ext}}\left(\boldsymbol{u}, \delta\boldsymbol{u}\right) - \delta W_{\mathrm{int}}\left(\boldsymbol{u}, \delta\boldsymbol{u}\right) \ . \tag{4.5}$$

Then, assuming that no external load depends on the deformation and following Section 2.4.3.2, the formulation for the fictitious domain approach reads

$$D_{\Delta u}\boldsymbol{G}\left(\boldsymbol{u}, \delta\boldsymbol{u}, \Delta\boldsymbol{u}\right) = \int_{\boldsymbol{\varphi}(\Omega)} \alpha\ \left(\operatorname{grad}\delta\boldsymbol{u} : \operatorname{grad}\Delta\boldsymbol{u}\boldsymbol{\sigma} + \operatorname{grad}\delta\boldsymbol{u} : \boldsymbol{c} : \operatorname{grad}\Delta\boldsymbol{u}\right)\ \mathrm{d}v \tag{4.6}$$

$$\delta W_{\mathrm{ext}}\left(\boldsymbol{u}, \delta\boldsymbol{u}\right) = \int_{\boldsymbol{\varphi}(\Omega)} \alpha\,\boldsymbol{f}\cdot\delta\boldsymbol{u}\ \mathrm{d}v + \int_{\boldsymbol{\varphi}(\Gamma_{\mathrm{N}})} \alpha\,\bar{\boldsymbol{t}}\cdot\delta\boldsymbol{u}\ \mathrm{d}a \tag{4.7}$$

$$\delta W_{\mathrm{int}}\left(\boldsymbol{u}, \delta\boldsymbol{u}\right) = \int_{\boldsymbol{\varphi}(\Omega)} \alpha\,\boldsymbol{\sigma} : \delta\boldsymbol{e}\ \mathrm{d}v\ . \tag{4.8}$$

However, this formulation is problematic when very small values of α are employed, as shown by Schillinger et al. [108, 109]. In this case, unphysical solutions within the fictitious domain can lead to significant contributions to the fictitious linearized part (4.6) as well as the fictitious virtual internal work (4.8). In theory, these contributions should be zero. To overcome this issue, we follow the idea proposed by Schillinger et al. [108] and neglect the deformation within the fictitious domain ($\boldsymbol{u}_{\mathrm{fict}} = \boldsymbol{0} \rightarrow \boldsymbol{F}_{\mathrm{fict}} = \boldsymbol{I} \rightarrow \boldsymbol{\sigma}_{\mathrm{fict}} = \boldsymbol{0}$). This is equivalent to repeatedly performing linear computations in the fictitious part. Following this approach, the convergence characteristics typical for high-order methods can be preserved [113]. The resulting formulation reads:

$$\begin{aligned} D_{\Delta u}\boldsymbol{G}\left(\boldsymbol{u}, \delta\boldsymbol{u}, \Delta\boldsymbol{u}\right) = &\int_{\boldsymbol{\varphi}\left(\Omega_{\mathrm{phys}}\right)} \left(\operatorname{grad}\delta\boldsymbol{u} : \operatorname{grad}\Delta\boldsymbol{u}\boldsymbol{\sigma} + \operatorname{grad}\delta\boldsymbol{u} : \boldsymbol{c} : \operatorname{grad}\Delta\boldsymbol{u}\right)\ \mathrm{d}v \\ &+ \int_{\Omega_{\mathrm{fict}}} \left(\boldsymbol{B}\delta\boldsymbol{u}\right)^{\mathrm{T}} \alpha\boldsymbol{\mathcal{C}}\left(\boldsymbol{B}\boldsymbol{u}\right)\ \mathrm{d}V \end{aligned} \tag{4.9}$$

$$\delta W_{\mathrm{ext}}\left(\boldsymbol{u}, \delta\boldsymbol{u}\right) = \int_{\boldsymbol{\varphi}\left(\Omega_{\mathrm{phys}}\right)} \boldsymbol{f}\cdot\delta\boldsymbol{u}\ \mathrm{d}v + \int_{\boldsymbol{\varphi}\left(\Gamma_{N,\mathrm{phys}}\right)} \alpha\,\bar{\boldsymbol{t}}\cdot\delta\boldsymbol{u}\ \mathrm{d}a \tag{4.10}$$

$$\delta W_{\text{int}}\left(\boldsymbol{u}, \delta\boldsymbol{u}\right) = \int\limits_{\boldsymbol{\varphi}\left(\Omega_{\text{phys}}\right)} \boldsymbol{\sigma} : \delta\boldsymbol{e} \, \mathrm{d}v + \int\limits_{\Omega_{\text{fict}}} \mathbf{0} \, \mathrm{d}V \ . \tag{4.11}$$

4.2 Discretization

The previous section introduced the nonlinear weak form for the FCM (4.11). To find an approximate solution using the finite element method, this continuous weak form needs to be discretized. Therefore, the solution $\boldsymbol{u} = [u, v, w]^{\mathrm{T}}$ is approximated by a linear combination of shape functions $\boldsymbol{N}$ and coefficients $\tilde{\boldsymbol{u}}$

$$\boldsymbol{u}\left(\boldsymbol{X}\right) \approx \boldsymbol{u}^{\mathrm{h}}\left(\boldsymbol{X}\right) = \boldsymbol{N}\left(\boldsymbol{X}\right)\tilde{\boldsymbol{u}} \ . \tag{4.12}$$

The shape function and coefficient matrices have the following structure

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{N}_u & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{N}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{N}_w \end{bmatrix} = \begin{bmatrix} N_u^1 & \dots & N_u^k & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & N_v^1 & \dots & N_v^l & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & N_w^1 & \dots & N_w^m \end{bmatrix} , \tag{4.13}$$

$$\tilde{\boldsymbol{u}} = \left[u^1, \dots, u^k, v^1, \dots, v^l, w^1, \dots, w^m\right]^{\mathrm{T}} \ . \tag{4.14}$$

The index bound k, l and m are the number of modes for the components of the displacement solution $\boldsymbol{u} = [u, v, w]^{\mathrm{T}}$. This is because high-order finite elements allow for different ansatz orders for different solution components (e.g. u, v, w) [121]. Hence, each block of $\boldsymbol{N}$ might have a different size so that the interleaved structure known from low-order discretizations does not generally apply [56].
While the approximation for the solution increment $\Delta\boldsymbol{u}$ directly follows the solution $\boldsymbol{u}$, the test function $\delta\boldsymbol{u}$ is deliberately chosen to use the same functions $\boldsymbol{N}$

$$\Delta\boldsymbol{u}\left(\boldsymbol{X}\right) \approx \Delta\boldsymbol{u}^{\mathrm{h}}\left(\boldsymbol{X}\right) = \boldsymbol{N}\left(\boldsymbol{X}\right)\Delta\tilde{\boldsymbol{u}} \tag{4.15a}$$

$$\delta\boldsymbol{u}\left(\boldsymbol{X}\right) \approx \delta\boldsymbol{u}^{\mathrm{h}}\left(\boldsymbol{X}\right) = \boldsymbol{N}\left(\boldsymbol{X}\right)\delta\tilde{\boldsymbol{u}} \ . \tag{4.15b}$$

This approach is known as *Bubnov-Galerkin* method and is the standard practice for the finite element method for problems in solid mechanics [56].

4.2.1 Discretized weak form

Employing Equation 4.12 and 4.15 the following discrete system of equations is derived

$$\left(\boldsymbol{K}_k^{\mathrm{C},i} + \boldsymbol{K}_k^{\sigma,i}\right)\Delta\boldsymbol{u}_k^i = \boldsymbol{G}_k^i \ . \tag{4.16}$$

Here, $\boldsymbol{K}_k^{\mathrm{C},i}$ and $\boldsymbol{K}_k^{\sigma,i}$ are the global consistent and geometric tangential stiffness matrices, respectively, at load step i and iteration k. $\Delta\boldsymbol{u}_k^i$ is the incremental displacement and $\boldsymbol{G}_k^i$ is the residual given as

$$\boldsymbol{G}_k^i = \boldsymbol{f}_k^{\mathrm{ext},i} - \boldsymbol{f}_k^{\mathrm{int},i}, \tag{4.17}$$

where $\boldsymbol{f}_k^{\text{ext},i}$ and $\boldsymbol{f}_k^{\text{int},i}$ are the global discrete vectors of external and internal virtual work, respectively. All these matrices and vectors are formed by assembling quantities, which are obtained on the element level via numerical integration

$$\boldsymbol{K}^{(\cdot)} = \mathbf{A}_{e=1}^{n_{\text{elem}}} \left(\boldsymbol{K}_e^{(\cdot)} \right) \ , \tag{4.18a}$$

$$\boldsymbol{f}^{(\cdot)} = \mathbf{A}_{e=1}^{n_{\text{elem}}} \left(\boldsymbol{f}_e^{(\cdot)} \right) \ . \tag{4.18b}$$

Here, $\mathbf{A}$ denotes the assembly operator [56] and $\boldsymbol{K}^{(\cdot)}$ and $\boldsymbol{f}^{(\cdot)}$ denote element matrices and vectors, respectively. Note that in the following element quantities are implied whenever quadrature is involved.

4.2.2 Transition to matrix notation

So far all quantities involved in the linearized weak form of equilibrium (4.11) have been formulated using second- and fourth-order tensors (stress, strain tensors and elasticity tensors, respectively). In order to provide a more efficient implementation, most discrete quantities will be formulated in matrix (Voigt) notation by utilizing tensor symmetries [56, 17]. By doing this, the number of components for every discrete term can be reduced significantly, which leads to fewer operations during numerical evaluations.
The Cauchy stress tensor $\boldsymbol{\sigma}$ is a symmetric second-order tensor, which can be represented as a vector with six components

$$\bar{\boldsymbol{\sigma}} = [\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}]^T \ . \tag{4.19}$$

Similarly, the variation of the Euler-Almansi strain tensor $\delta\boldsymbol{e}$ can be written as

$$\delta\bar{\boldsymbol{e}} = [e_{11}, e_{22}, e_{33}, 2e_{12}, 2e_{13}, 2e_{23}]^T \ . \tag{4.20}$$

Here, the factors in the last three terms have been introduced to preserve the scalar identity of Equation (2.57a)

$$\int_{\boldsymbol{\varphi}(\Omega)} \boldsymbol{\sigma} : \delta\boldsymbol{e} \, \mathrm{d}v = \int_{\boldsymbol{\varphi}(\Omega)} \delta\bar{\boldsymbol{e}}^T \bar{\boldsymbol{\sigma}} \, \mathrm{d}v \ . \tag{4.21}$$

4.2.3 Discretized external and internal virtual work

The residual $\boldsymbol{G}$ is defined as the difference between external and internal work (4.17). Here, the vector of external virtual work is only defined for the physical part of the domain

$$\boldsymbol{f}^{\text{ext}} = \int_{\boldsymbol{\varphi}\left(\Omega_{\text{phys}}\right)} \boldsymbol{N}^{\mathrm{T}} \boldsymbol{f} \, \mathrm{d}v + \int_{\boldsymbol{\varphi}\left(\Gamma_{N,\text{phys}}\right)} \boldsymbol{N}^{\mathrm{T}} \boldsymbol{t} \, \mathrm{d}v, \tag{4.22}$$

where the first term stands for the virtual work introduced in Ω_{phys} by a source term $\boldsymbol{f}$. The second term represents the natural boundary conditions for an applied traction $\boldsymbol{t}$ along $\Gamma_{N,\text{phys}}$.

The discrete vector of internal virtual work is defined as

$$\boldsymbol{f}^{\text{fict}} = \int\limits_{\boldsymbol{\varphi}\left(\Omega_{\text{phys}}\right)} \boldsymbol{B}^{\mathrm{T}} \bar{\boldsymbol{\sigma}} \, \mathrm{d}v \ , \tag{4.23}$$

where $\bar{\boldsymbol{\sigma}}$ is the Cauchy stress in vector notation (4.19) and is the spatial version of the well known strain operator [17]

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{N}_{u,x} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{N}_{v,y} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{N}_{w,z} \\ \boldsymbol{N}_{u,y} & \boldsymbol{N}_{v,x} & \boldsymbol{0} \\ \boldsymbol{N}_{u,z} & \boldsymbol{0} & \boldsymbol{N}_{w,x} \\ \boldsymbol{0} & \boldsymbol{N}_{v,z} & \boldsymbol{N}_{w,y} \end{bmatrix} \ . \tag{4.24}$$

Here, $\boldsymbol{N}_{i,j}$ are the spatial derivatives of the i^{th} (solution) component of $\boldsymbol{N}$ with respect to the current position component $j \in \{x, y, z\}$, e.g.

$$\boldsymbol{N}_{i,x} = \frac{\partial \boldsymbol{N}_i}{\partial x} = \frac{\partial \boldsymbol{N}_i}{\partial X}\frac{\partial X}{\partial x} + \frac{\partial \boldsymbol{N}_i}{\partial Y}\frac{\partial Y}{\partial x} + \frac{\partial \boldsymbol{N}_i}{\partial Z}\frac{\partial Z}{\partial x} \ . \tag{4.25}$$

4.2.4 Consistent stiffness matrix

The consistent part of the discrete tangential stiffness matrix is obtained as

$$\boldsymbol{K}^{\mathrm{C}} = \int\limits_{\boldsymbol{\varphi}\left(\Omega_{\text{phys}}\right)} \boldsymbol{B}^{\mathrm{T}} \bar{\boldsymbol{c}} \boldsymbol{B} \, \mathrm{d}v + \int\limits_{\Omega_{\text{fict}}} \boldsymbol{B}_0^{\mathrm{T}} \alpha \bar{\boldsymbol{\mathcal{C}}} \boldsymbol{B}_0 \, \mathrm{d}V \ , \tag{4.26}$$

where $\bar{\boldsymbol{c}}$ and $\bar{\boldsymbol{\mathcal{C}}}$ are the spatial and linear elasticity tensors in matrix notation (see also sections 2.3.2 and 2.3.1). In the current case of isotropic elasticity the fourth-order elasticity tensors can be transformed into symmetric 6 by 6 matrices. Hence, e.g. the spatial elasticity tensor can be written as [17]

$$\bar{\boldsymbol{c}} = \begin{bmatrix} c_{1111} & c_{1122} & c_{1133} & c_{1112} & c_{1113} & c_{1123} \\ & c_{2222} & c_{2233} & c_{2212} & c_{2213} & c_{2223} \\ & & c_{3333} & c_{3312} & c_{3313} & c_{3323} \\ & & & c_{1212} & c_{1213} & c_{1223} \\ & \text{sym.} & & & c_{1313} & c_{1323} \\ & & & & & c_{2323} \end{bmatrix} \ . \tag{4.27}$$

The initial strain operator $\boldsymbol{B}_0$ has the same structure as the spatial operator $\boldsymbol{B}$ (4.24) and is obtained by performing all derivatives with respect to the initial configuration (i.e. $j \in \{X, Y, Z\}$).

4.2.5 Geometric stiffness matrix

The geometric tangential stiffness matrix is defined as [9, 17]

$$\boldsymbol{K}^{\sigma} = \int\limits_{\boldsymbol{\varphi}(\Omega_{\text{phys}})} \nabla\tilde{\boldsymbol{N}}^{T}\tilde{\boldsymbol{\sigma}}\nabla\tilde{\boldsymbol{N}}\,\mathrm{d}v\,, \tag{4.28}$$

where the operators $\nabla\tilde{\boldsymbol{N}}$ and $\tilde{\boldsymbol{\sigma}}$ are defined as

$$\nabla\tilde{\boldsymbol{N}} = \begin{bmatrix} \nabla\boldsymbol{N}_u & 0 & 0 \\ 0 & \nabla\boldsymbol{N}_v & 0 \\ 0 & 0 & \nabla\boldsymbol{N}_w \end{bmatrix} \quad \text{and} \quad \tilde{\boldsymbol{\sigma}} = \begin{bmatrix} \boldsymbol{\sigma} & 0 & 0 \\ 0 & \boldsymbol{\sigma} & 0 \\ 0 & 0 & \boldsymbol{\sigma} \end{bmatrix}. \tag{4.29}$$

The shape function gradients $\nabla\boldsymbol{N}_{(\cdot)}$ have the following structure:

$$\nabla\boldsymbol{N}_{(\cdot)} = \begin{bmatrix} N^1_{(\cdot),x} & \dots & N^n_{(\cdot),x} \\ N^1_{(\cdot),y} & \dots & N^n_{(\cdot),y} \\ N^1_{(\cdot),z} & \dots & N^n_{(\cdot),z} \end{bmatrix}, \tag{4.30}$$

where n is the number of modes of the current field $(\cdot)$.

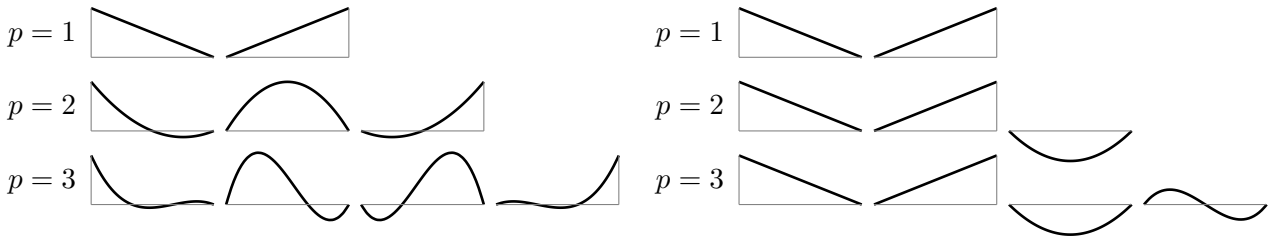
4.3 The p-version of the finite element method

The p-version of the FEM utilizes polynomial shape functions of higher order [7, 121]. The approximation quality is improved by elevating the polynomial degree of the shape functions, as opposed to reducing the element size as it is done in the h-version. High-order finite elements allow for highly accurate results while naturally alleviating locking phenomena and providing robustness against mesh distortion in large strain applications. Another key feature of high-order finite elements is the achievable exponential convergence for smooth problems. The most prominent shape functions used within the p-version are Lagrange and integrated Legendre polynomials, see Figure 4.2. In contrast to the Lagrange basis, the hierarchic basis obtained from using integrated Legendre polynomials allows for a local adaption of the polynomial order, while preserving full continuity. The integrated Legendre basis also leads to better conditioned systems of linear equations due to its orthogonality property.

4.3.1 Hierarchic shape functions

In the one-dimensional case, the hierarchic basis of the p-version are constructed as follows

$$N^{1\text{D}}_1(r) = \tfrac{1}{2}(1-r) \tag{4.31a}$$
$$N^{1\text{D}}_2(r) = \tfrac{1}{2}(1+r) \tag{4.31b}$$
$$N^{1\text{D}}_i(r) = \phi_{i-1}(r), \quad i = 3,\dots,p+1\,, \tag{4.31c}$$



(a) Lagrange basis **(b)** Hierarchic basis

Figure 4.2: Comparison of some Lagrange and hierarchic integrated Legendre basis functions in 1D.

where the first two modes are the linear shape functions known from the h-version and ϕ_i are integrated Legendre polynomials [6] defined as

$$\phi_j(r) = \sqrt{\frac{2j-1}{2}} \int_1^r P_{j-1}(x)dx \tag{4.32}$$

$$= \frac{1}{\sqrt{(4j-2)}} \left(P_j(r) - P_{j-2}(r)\right), \quad j = 2, 3, \ldots . \tag{4.33}$$

The Legendre polynomials P_j can be computed using Bonnet's recursion formula

$$(n+1)\, P_{j+1}(x) = (2n+1)\, xP_n(x) - nP_{n-1}(x) \tag{4.34}$$

with

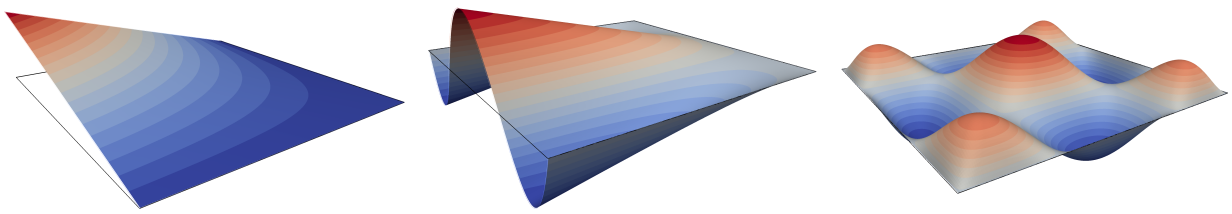
$$P_0(x) = 1, \quad P_1(x) = x\ . \tag{4.35}$$

The two- and three-dimensional ansatz spaces are constructed as tensor products of the one-dimensional basis, forming the so-called tensor product space:

$$\mathbf{N}^{2\mathrm{D}}(r,s) = \mathbf{N}^{1\mathrm{D}}(r) \otimes \mathbf{N}^{1\mathrm{D}}(s) \tag{4.36a}$$

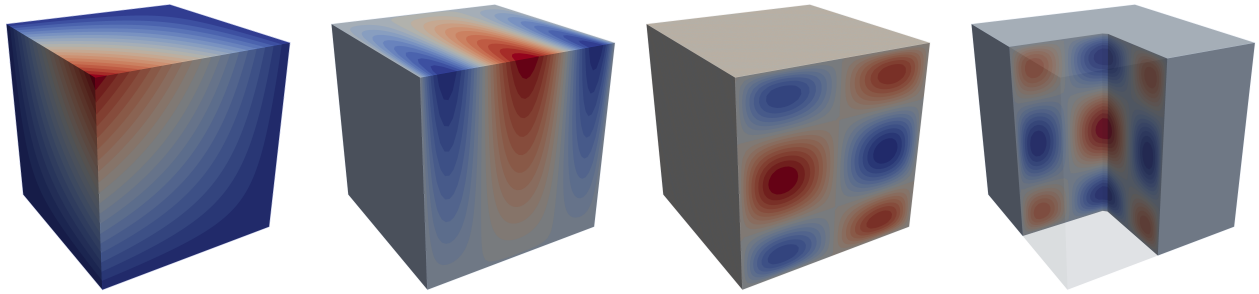
$$\mathbf{N}^{3\mathrm{D}}(r,s,t) = \mathbf{N}^{1\mathrm{D}}(r) \otimes \mathbf{N}^{1\mathrm{D}}(s) \otimes \mathbf{N}^{1\mathrm{D}}(t). \tag{4.36b}$$

Each shape function of these bases can be associated to a topological component (node, edge, face, volume) of the underlying element, see Figures 4.3 and 4.4.



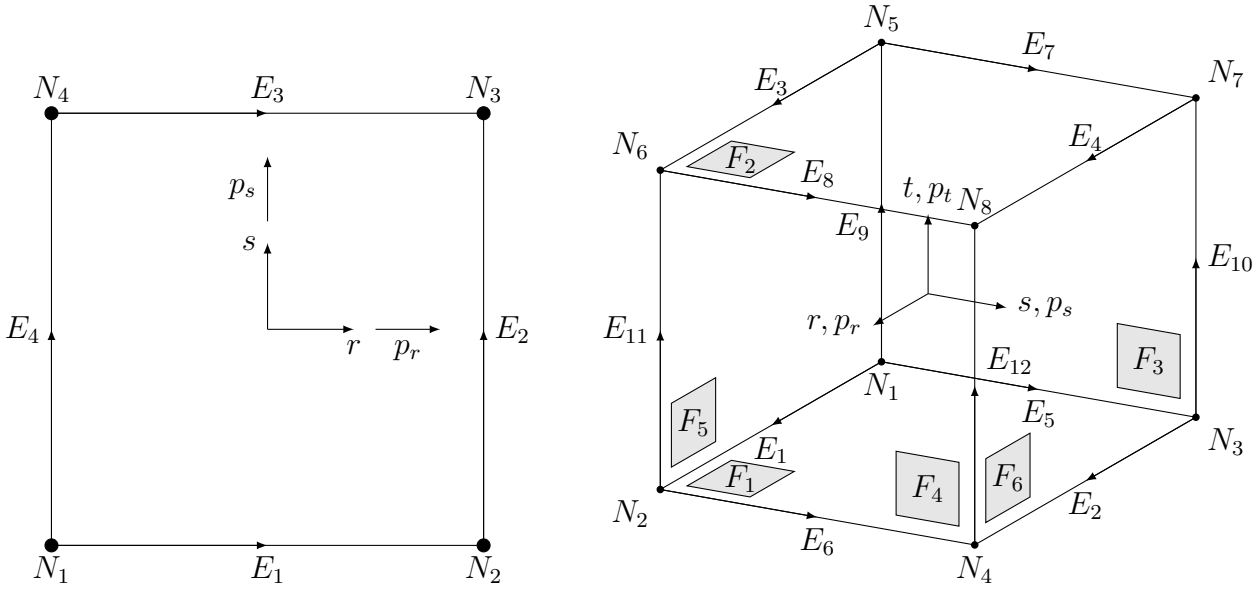
Nodal mode Edge mode Internal mode

Figure 4.3: Representative two-dimensional mode shapes for $p = 4$.



Nodal mode Edge mode Face mode Internal mode

Figure 4.4: Representative three-dimensional mode shapes for $p = 4$.



(a) Standard quadrilateral **(b)** Standard hexahedral

Figure 4.5: Numbering of nodes, edges and faces on standard elements.

Furthermore, the tensor product space can be reduced by removing some of the face and volume modes leading to the so-called trunk space. This formulation is similar to the serendipity spaces [9] known from Lagrange finite elements. To illustrate the construction of the different kinds of modes, consider the standard hexahedral element depicted in Figure 4.5b:

- The **nodal modes** correspond the standard tri-linear basis functions known from low-order formulations. Each mode has a value of one at its associated node i and is zero at all other nodes.

$$N_{1,1,1}^{\mathrm{N}_i}(r,s,t) = \frac{1}{8}(1 + r_i r)(1 + s_i s)(1 + t_i t) \qquad i = 1, \ldots, 8 \tag{4.37}$$

  Here (r_i, s_i, t_i) denote the local coordinates $(\pm 1, \pm 1, \pm 1)$ of node i.

- Starting from $p = 2$ each edge has $p_d - 1$ **edge modes**, where p_d is the polynomial order in the local direction of the edge ($d \in \{r, s, t\}$). These modes are a combination of a

one-dimensional high-order shape function (4.31c) along the edge and two linear blending modes, e.g., the modes for edge 1 read

$$N^{\mathrm{E}_1}_{i,1,1}(r,s,t) = \frac{1}{4}(1-s)(1-t)\,\phi_i(r) \quad . \tag{4.38}$$

- The **face modes** of each edge are a combination of two high-order shape functions (4.31c) and a linear mode. Hence, they are zero at all nodes and edges. For face 1 the modes are defined as

$$N^{\mathrm{F}_1}_{i,j,1}(r,s,t) = \frac{1}{2}(1-t)\,\phi(r)_i\,\phi(s)_j \quad . \tag{4.39}$$

  Depending on whether the tensor or the trunk space is used, the polynomial degrees i and j for the local directions r and s, respectively, have different bounds

tensor space	**trunk space**
$i = 2,\ldots,p_r$	$i = 2,\ldots,p_r-2$
$j = 2,\ldots,p_s$	$j = 2,\ldots,p_s-2$
	$i+j = 4,\ldots,\max(p_r,p_s)$

- The **volume (internal) modes** are constructed using three high-order shape functions (4.31c). They vanish on all nodes, edges and faces

$$N^{\mathrm{V}}_{i,j,k}(r,s,t) = \phi_i(r)\,\phi_j(s)\,\phi_k(t) \quad . \tag{4.40}$$

  Similar to the face modes, the choice of the ansatz space determines the bounds for the polynomial order in each local direction.

tensor space	**trunk space**
$i = 2,\ldots,p_r$	$i = 2,\ldots,p_r-2$
$j = 2,\ldots,p_s$	$j = 2,\ldots,p_s-2$
$k = 2,\ldots,p_t$	$k = 2,\ldots,p_t-2$
	$i+j+k = 6,\ldots,\max(p_r,p_s,p_t)$

The resulting spaces can be further modified to yield anisotropic variants by choosing different polynomial degrees for each local direction. This has been shown to be beneficial for problems involving thin-walled structures [99]. Here, the numerical effort can be greatly reduced by setting different polynomial degrees for the in-plane and the thickness directions. Most of the research for the FCM, including this work, focuses on Cartesian shaped elements. However, there are also polynomial spaces for tetrahedra developed by Wassouf et al. [129], which have been utilized for the FCM by Varduhn and co-workers [125].

4.3.2 Hierarchic refinement strategies

High-order finite elements show exponential rates of convergence when applied to smooth problems. Such problems do not incorporate singular effects like concentrated loads or re-entrant corners. Unfortunately, most practical problems contain some form of singularity, either of physical nature or introduced by model simplifications. To reduce the corresponding discretization errors, meshes are classically refined towards singularities to "capture" singular effects (errors) within small elements. If high-order finite elements are used, the polynomial degree can also be reduced gradually towards the singularity. By doing that, singularity-induced oscillations (similar to Gibbs phenomenon) are damped out [8]. The combined adaptation of element sizes and polynomial degrees to improve approximation accuracy is the fundamental idea of the class of *hp*-methods [4]. If applied correctly, *hp*-methods can recover exponential convergence rates even for singular problems. Mesh adaption is either done a priori based on engineering experience or during simulation, where the latter needs to be steered using some form of error estimator [120]. While *hp*-methods have superior approximation properties, the implementation of such methods is anything but trivial, especially in three dimensions.

One way to implement mesh adaptation is to remesh those parts of the domain, which contain large error contributions, if necessary followed by a projection of the old solution onto the new mesh. This requires a highly sophisticated meshing engine, which allows for fully automatic mesh adaptation. Since such engines are not yet available for three-dimensional *hp*-discretizations, this approach is mostly used for low-order meshes involving triangular and tetrahedral elements.

Another possibility for refinement is to replace elements containing large error contributions by smaller elements with an appropriate polynomial order. This method does not require the meshing engine mentioned earlier, but introduces hanging nodes, see Figure 4.6. Here, the

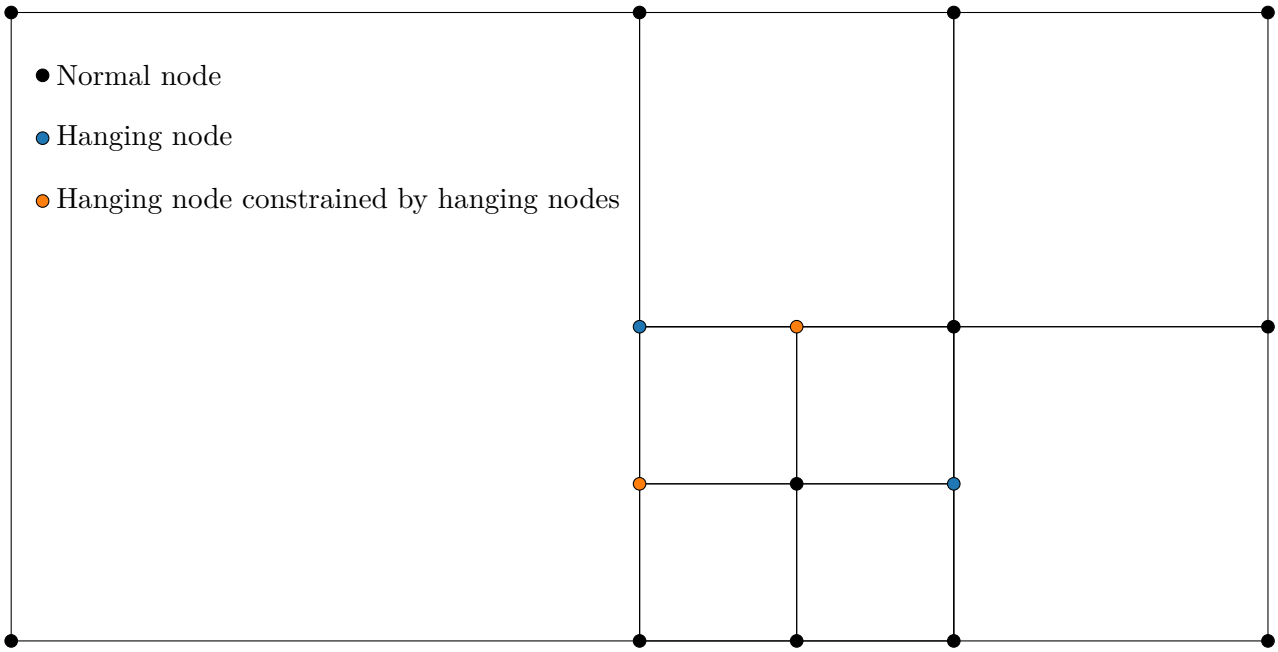
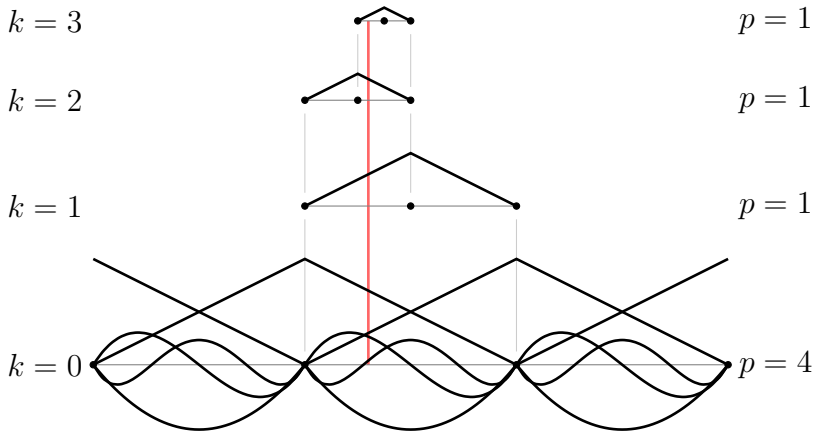


Figure 4.6: Hanging nodes introduced by refine-by-replacement approach.

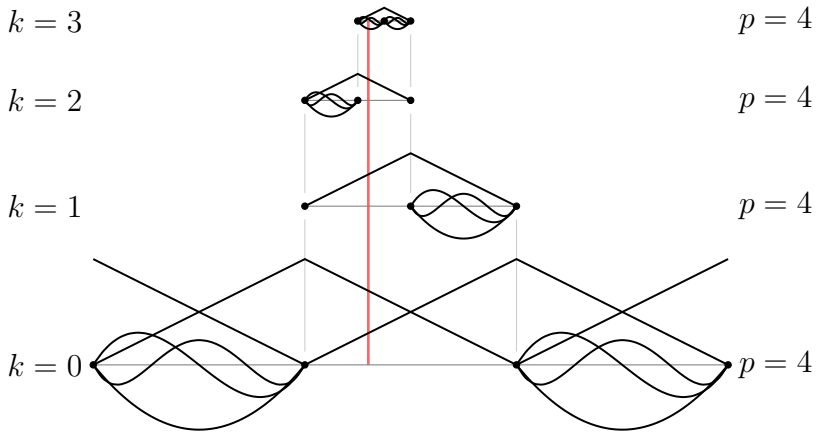
shape functions between adjacent elements become incompatible due to the refinement. To enforce compatibility the shape functions need to be constrained correctly. This is especially complicated if several levels of refinement are utilized, and modes need to be constrained in terms of already constraint modes.
A much simpler formulation can be obtained by applying the idea of *refinement-by-superposition*. Here, a coarse high-order base mesh is superposed by finer overlay mesh in regions of interest.

$$\boldsymbol{u} = \boldsymbol{u}_b + \boldsymbol{u}_o \tag{4.41}$$

Building upon the work of Rank [100] and Schillinger [111], Zander et al. [140, 139] introduced a hierarchical refinement scheme, which superposes high-order overlay meshes: the multi-level *hp*-method (Figure 4.7b). Here, compatibility is ensured by deactivating the degrees of freedom at the boundary of the overlay meshes. Furthermore, higher modes on lower levels are deactivated to avoid linear dependencies.



(a) *hp-d* method [111]

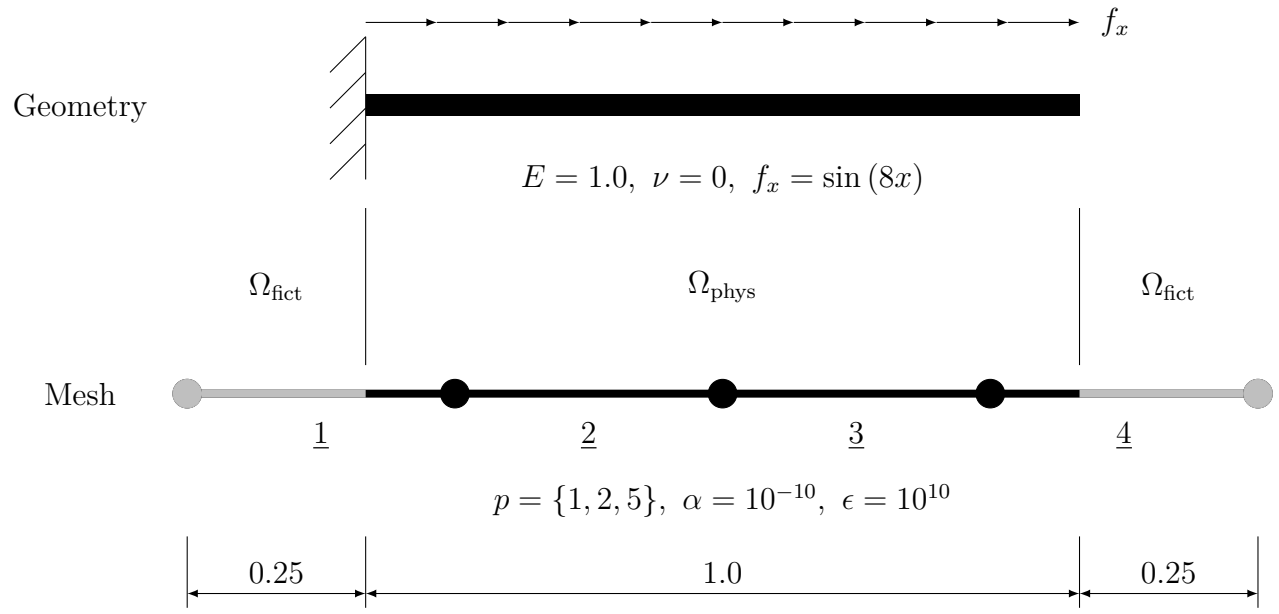


(b) Multi-level *hp* method [140].

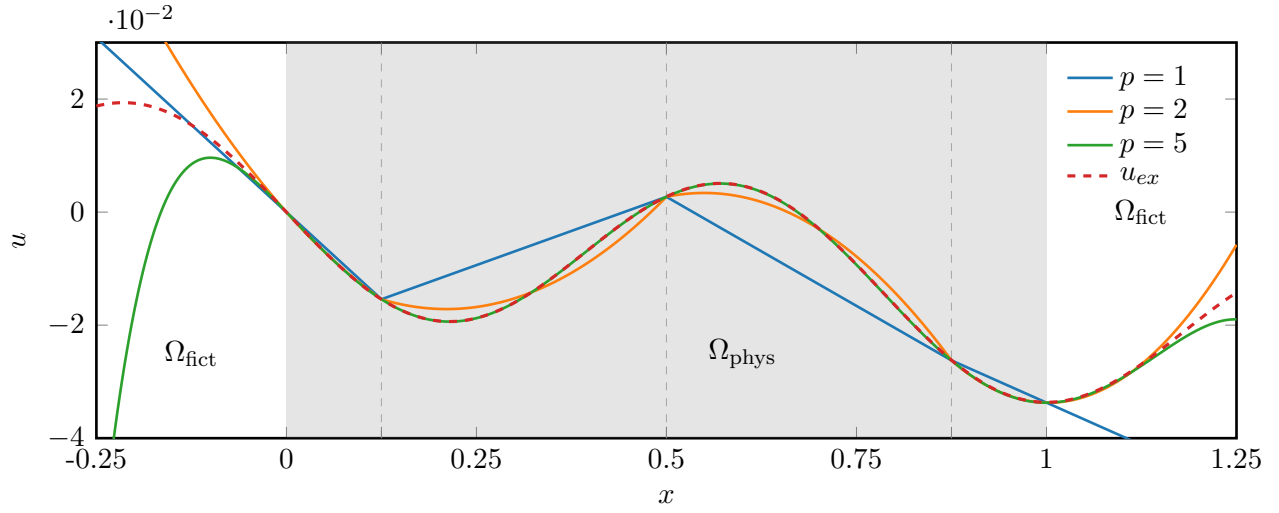
Figure 4.7: Comparison of hierarchic refinement schemes in 1D for a base mesh of $p = 4$ and 4 levels of refinement.

4.4 Multiple meshes

The essence of fictitious domain approaches is to disregard the physical boundaries of the problem geometry for discretization. This works well, when the solution can be represented within the physical domain Ω_{phys} and is free to smoothly extend into the fictitious domain Ω_{fict}, as shown in Figure 4.8.



(a) Geometrical setup and discretization



(b) Computed displacements vs. analytical solution

Figure 4.8: One-dimensional linear elastic problem embedded in a slightly larger mesh of 4 elements. The Dirichlet boundary condition is enforced using a weak penalty constraint (Section 4.6.1).

These requirements are usually met for the analysis of smooth problems involving a single body. However, in the presence of material interfaces, plasticity and for cases, where multiple bodies closely interact (e.g. contact), further steps are necessary to correctly represent the

solution. Material interfaces introduce weak discontinuities (kinks) into the solution field. The nonconforming meshes employed by fictitious domain approaches cannot correctly capture these discontinuities, when the interfaces pass through the elements (see Figures 4.9 and 4.10).

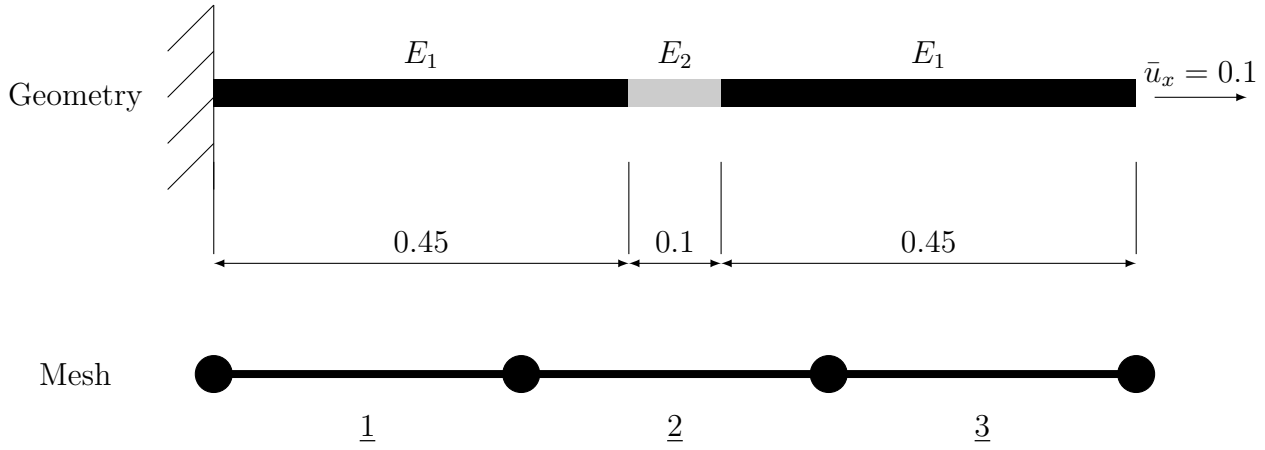


Figure 4.9: One-dimensional linear elastic problem involving regions of different stiffness. Choosing $E_1 \neq E_2$ introduces a material interface, while choosing $E_2 \to 0$ results in a gap. The problem is loaded by a prescribed displacement $\bar{u}_x$. The mesh, containing 3 elements, conforms with the outer bounds of the geometry. The region with stiffness E_2 is fully contained in element 2.

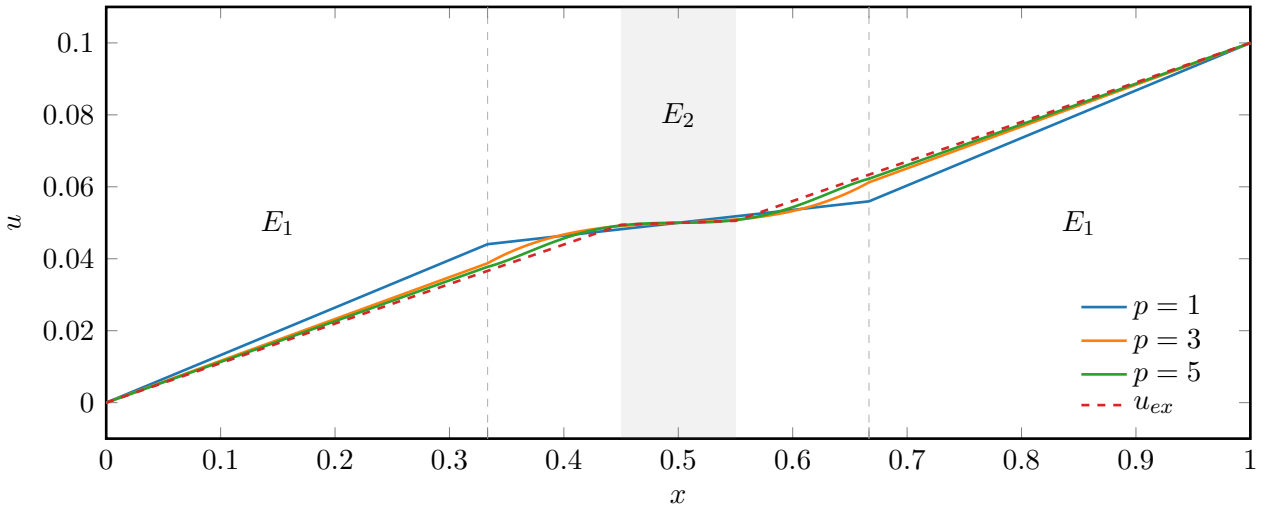


Figure 4.10: Displacement solution for a problem involving a material interface ($E_1 = 1.0$, $E_2 = 10.0$).

The same problem can arise for multiple bodies embedded in a common mesh. In this case the solution needs to be able to represent jumps in the displacement field (strong discontinuities). In the case of high-order discretizations embedded discontinuities can introduce severe oscillations (see Figure 4.11).
Different strategies are available to overcome this problem, one of which is to refine the discretization towards the interface. This only reduces the problem, unless the refinement

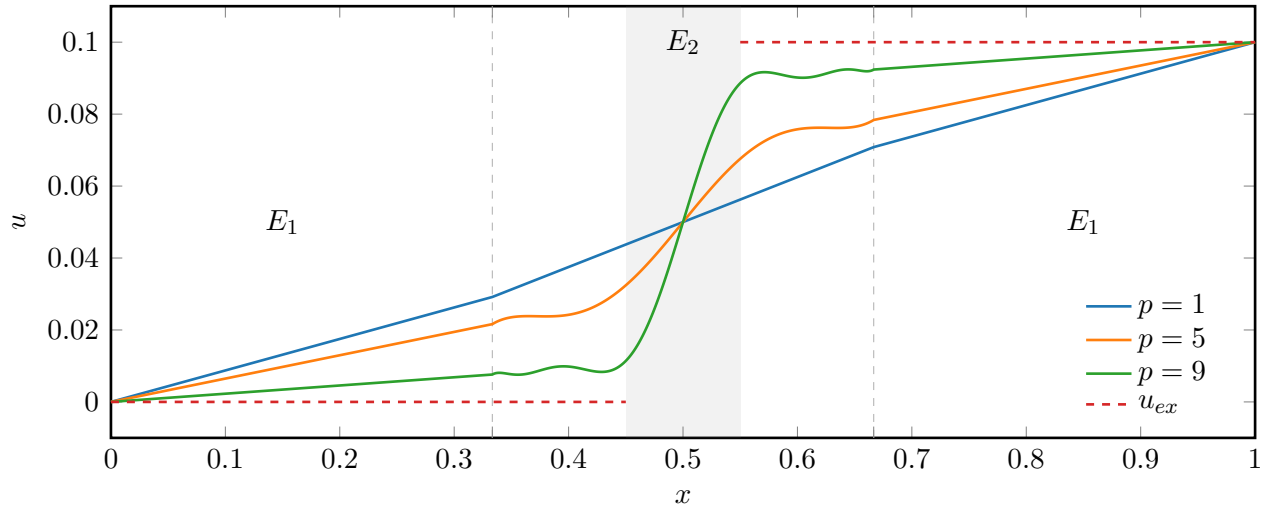


Figure 4.11: Displacement solution for a problem involving a gap ($E_1 = 1.0$, $E_2 = 1.0^{-10}$).

captures the interface exactly [28]. While no explicit boundary representation is necessary for this approach the refinement effort can be greatly disproportionate to the possible improvements. A more efficient strategy is to enrich the discretization at the interface with modes that are able to represent the weak discontinuity, see e.g. [57, 98, 31]. This is an idea employed by partition-of-unity methods [5], generalized FEMs [118] and extended FEMs [79]. To be able to construct these enrichment modes, the interface needs to be either known explicitly or represented by a distance field. Furthermore, the construction of enrichment modes for arbitrary interfaces in three dimensions is not straightforward.
Another alternative, which is employed in this work, is to embed every physical body or material domain in its own mesh. Hence, the solution fields of the distinct meshes do not need to represent any discontinuities and can rely on the property of *smooth extension*. Compatibility between the physical solutions of the meshes is enforced by applying weak constraints to either couple the meshes (Section 4.6.2) or enforce contact conditions (Chapter 6) [103, 16, 28].

4.5 Numerical integration of discontinuous integrands

In order to integrate the discrete weak forms introduced in Section 4.2.1, numerical quadrature rules can be employed

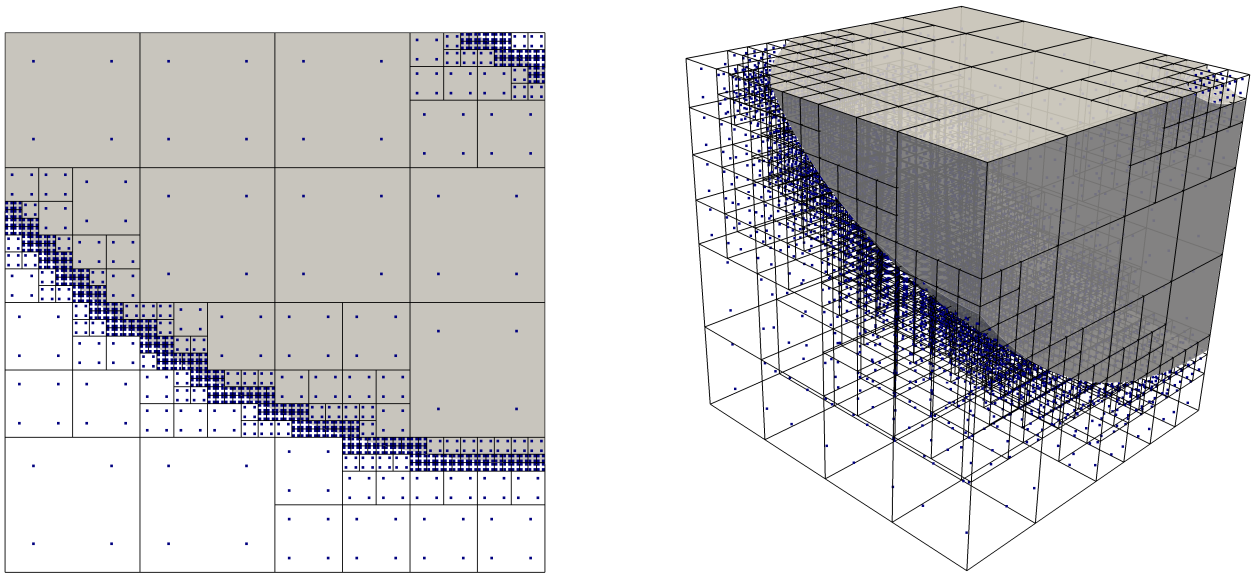
$$\int_{\Omega} \boldsymbol{F}(\boldsymbol{x}) \; \mathrm{d}V \approx \sum_{i=1}^{n_{\mathrm{qp}}} \boldsymbol{F}(\boldsymbol{x}(\boldsymbol{\xi}_i))\, w_i \det \boldsymbol{J}(\boldsymbol{\xi}_i) \tag{4.42}$$

Here, the continuous integral of a possibly vector-valued function $\boldsymbol{F}(\boldsymbol{x})$ is approximated as a sum of function evaluations over n_{qp} quadrature points $\boldsymbol{\xi}_i$ with associated weights w_i. Since these quadrature rules are generally formulated for an index domain $\boldsymbol{\xi} \in [-1, 1]^d$ of dimension d, the numerical integral is mapped to the real domain by multiplying with the determinant of the Jacobian $\boldsymbol{J}$ at the specific integration points $\boldsymbol{\xi}_i$. Cells that are fully contained inside the physical domain can be treated like standard continuum elements using Gauss-Legendre quadrature rules [56, 9].

In cells that are cut by the domain boundary, the integrand is not continuous anymore. Hence, Gauss-Legendre quadrature, which assumes C^∞ polynomials, cannot be applied without introducing a significant integration error [101]. This is a challenge not only for fictitious domain approaches but also for formulations like the generalized (GFEM) and the extended (XFEM) finite element method. Therefore, different schemes have been developed to integrate discontinuous integrands correctly. Some of these include composed integration based on space-trees [26] and exact partitioning [68, 69, 39, 127] as well as custom quadratures obtained by moment fitting [81, 119, 58].

4.5.1 Composed integration based on space-trees

The classical approach to integrate discontinuous integrands within the finite cell method employs the partitioning of the integration-domain using a quadtree or octree [26] in two and three dimensions, respectively. Therefore, the integration domain is recursively refined towards the domain boundary, see Figure 4.13. In each of the resulting cells a standard quadrature



(a) Quadtree **(b)** Octree

Figure 4.12: Example of a domain partitioned using space-trees. Integration points are shown for a quadrature order of $q = 2$.

scheme is applied (see Equation (4.42)) so that the final quadrature rule reads

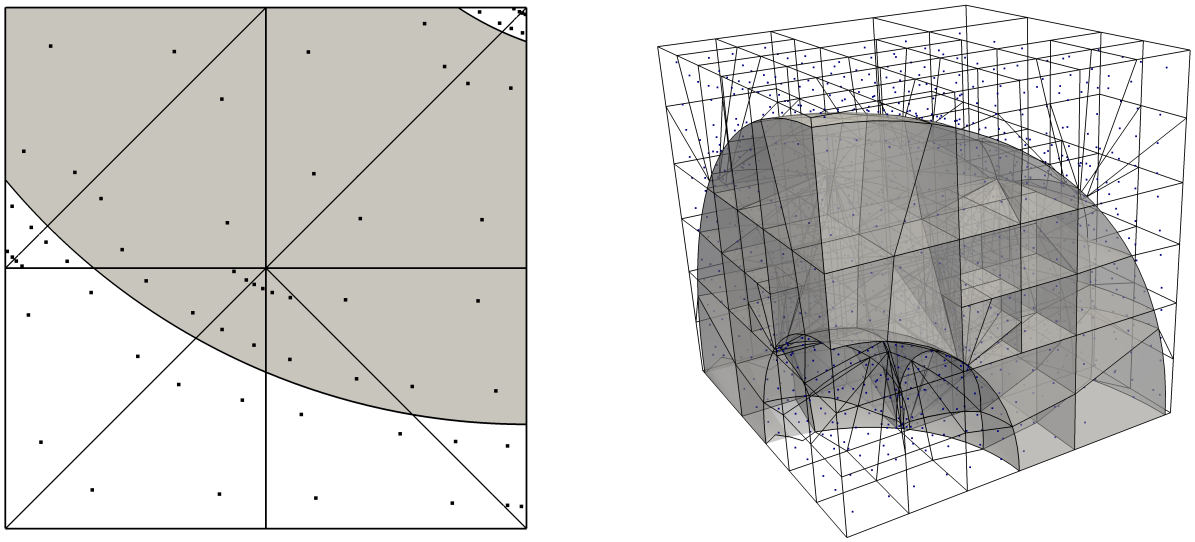
$$\int_\Omega \boldsymbol{F}(\boldsymbol{x})\ \mathrm{d}V \approx \sum_{j=1}^{n_{\mathrm{sc}}} \sum_{i=1}^{n_{\mathrm{qp}}} \boldsymbol{F}(\boldsymbol{x}(\boldsymbol{\xi}_i))\, w_i \det J_j^{\mathrm{sc}}(\boldsymbol{\xi}_i) \det J(\boldsymbol{\xi}_i)\ . \tag{4.43}$$

The index spaces of the sub-cells and the original cell are connected by the additional mapping $\det J_j^{\mathrm{sc}}(\boldsymbol{\xi}_i)$.
In most cases, a quadrature based on a space-tree decomposition is only exact for an infinite number of refinements. However, the integration error is confined to the smallest cut sub-cells

and can be controlled by the number of refinements. Numerical quadrature based on space-tree decompositions represents a very robust scheme that can be applied to all geometric models that provide an inside-outside (*point-in-membership*) test. While this can also be stated for a uniform grid decomposition [65], the space-tree approach leads to far fewer integration points. The number of integration points can be decreased even further by only considering either the physical or the fictitious domain in full resolution. For a thorough overview of possible modifications the reader is referred to [24].

4.5.2 Composed integration using exact geometric partitioning

Composed integration rules can also utilize domain decompositions based on exact geometric partitioning to minimize the number of integration points and the integration error. Here, cut cells are decomposed into sub-cells with curved boundaries, which coincide with the geometric boundary of the domain. This approach has gained a lot of attention in recent years, which can be largely attributed to the rising interest in fictitious domain methods and extended finite element methods. The basic requirement for this to work is to be able to compute ray-intersections with the geometric boundary. In the case of B-Rep geometries (Section 5.1), which are state of the art in industrial CAD applications, decomposition algorithms have been presented by Nadal et al. [82] and Kudela et al. [68, 69]. Another interesting class of geometric models are level-sets (Section 5.1). These are implicit geometry descriptions, which can be numerically described as continuous fields. Here, noteworthy contributions include Fries et al. [39], Verhoosel et al. [127] and Lehrenfeld et al. [72].



(a) Exact two-dimensional partitioning [68] **(b)** Smart-Octree [69]

Figure 4.13: Domain partitioned exactly using blended segments. Integration points are shown for a quadrature order of $q = 2$.

4.5.3 Numerical quadrature rules based on moment fitting

A third alternative to numerically integrate cell containing discontinuous integrands is to use custom quadratures obtained from moment fitting [58]. The basic idea it to find a set of integration points $\{\boldsymbol{\xi}\}$ and weights $\{w\}$ that exactly integrates all the high-order products for the finite element matrices (see Equations (4.26), (4.28), (4.22) and (4.23)). In its most general form the moment fitting approach tries to determine the optimal number of points, their position and the respective weights. This leads to a nonlinear problem with a non-symmetric, generally non-square system

$$\begin{bmatrix} f_1(\boldsymbol{\xi}_1) & \dots & f_1(\boldsymbol{\xi}_n) \\ \vdots & \ddots & \vdots \\ f_m(\boldsymbol{\xi}_1) & \dots & f_m(\boldsymbol{\xi}_n) \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \int f_1 \, dV \\ \vdots \\ \int f_m \, dV \end{bmatrix} . \tag{4.44}$$

Here, f_i are products of polynomial functions, which are more generic and much cheaper to evaluate than actual finite element matrices. Joulaian et al. [58] show that integrated Legendre functions work very well within this method. The integrals of the right-hand side of Equation (4.44) are the *moments*, which give the method its name. To simplify this problem, the positions of the integration points can be fixed, based on standard Gauss-Legendre or Gauss-Lobatto abscissas. This way, the problem becomes linear and only depends on the integration weights. What remains, is the question of how to efficiently integrate the moments in Equation (4.44). One possibility is to transform the volume integrals of the moments into surface integrals using the divergence theorem

$$\int_\Omega f_i \, dV = \int_\Omega \nabla \cdot \boldsymbol{g}_i \, dV = \int_\Gamma \boldsymbol{g}_i \cdot \boldsymbol{n} \, \mathrm{d}A \, , \tag{4.45}$$

where $\boldsymbol{g}_i$ are so called antiderivatives and $\boldsymbol{n}$ is the unit outward normal on the surface of the geometric body under consideration. The antiderivatives can be constructed following Müller et al. [81]

$$\boldsymbol{g}_i = \frac{1}{3} \begin{bmatrix} \int f_i \, dx \\ \int f_i \, dy \\ \int f_i \, dz \end{bmatrix} \, , \tag{4.46}$$

where the individual integrals can be easily computed in closed form. Furthermore, using the methods introduced in the previous chapter to automatically recover the surface of a cut cell, the moment fitting approach can be robustly applied to a wide range of geometric models.

4.6 Weakly enforced Dirichlet constraints

Boundary conditions play a crucial role in the analysis of mechanical problems. Since the FCM employs a fictitious domain approach, geometric boundaries might not coincide with element (cell) edges. In that case, boundary conditions must be enforced in a weak (integral) sense.

For Neumann boundary conditions, the integral formulation follows *naturally* from the weak form (see Equation (2.57b))

$$f_{\mathrm{N}}\left(\delta \boldsymbol{u}\right) = \int_{\Gamma_{\mathrm{N}}} \delta \boldsymbol{u}^{\mathrm{T}} \bar{\boldsymbol{t}} \, \mathrm{d}a \ . \tag{4.47}$$

This also means that homogeneous Neumann boundary conditions are fulfilled automatically. The following sections will introduce different methods to enforce Dirichlet (essential) boundary conditions as well as formulations to couple meshes. The latter group offers interesting possibilities to model multi-material bodies and weak discontinuities in general [28] (see also Section 4.4) .
Consequently, all of these formulations will lead to additional quantities in the discretized weak form, namely $\boldsymbol{K}^{\mathrm{D}}$, $\boldsymbol{f}^{\mathrm{ext,D}}$ and $\boldsymbol{f}^{\mathrm{int,D}}$

$$\left[\boldsymbol{K}^{\mathrm{C}} + \boldsymbol{K}^{\sigma} + \boldsymbol{K}^{\mathrm{D}}\right] \Delta \bar{\boldsymbol{u}} = \boldsymbol{f}^{\mathrm{ext}} - \boldsymbol{f}^{\mathrm{int}} + \boldsymbol{f}^{\mathrm{ext,D}} - \boldsymbol{f}^{\mathrm{int,D}} \ . \tag{4.48}$$

The contributions of the weak boundary conditions are obtained by numerically integrating along the related surfaces. However, all discrete quantities are constructed using the shape functions of the embedding elements. To this end, the coordinates of the integration points inside a segment are mapped into the associated element, as shown for the two-dimensional case in Figure 4.14. This involves a backward mapping from the undeformed global space to

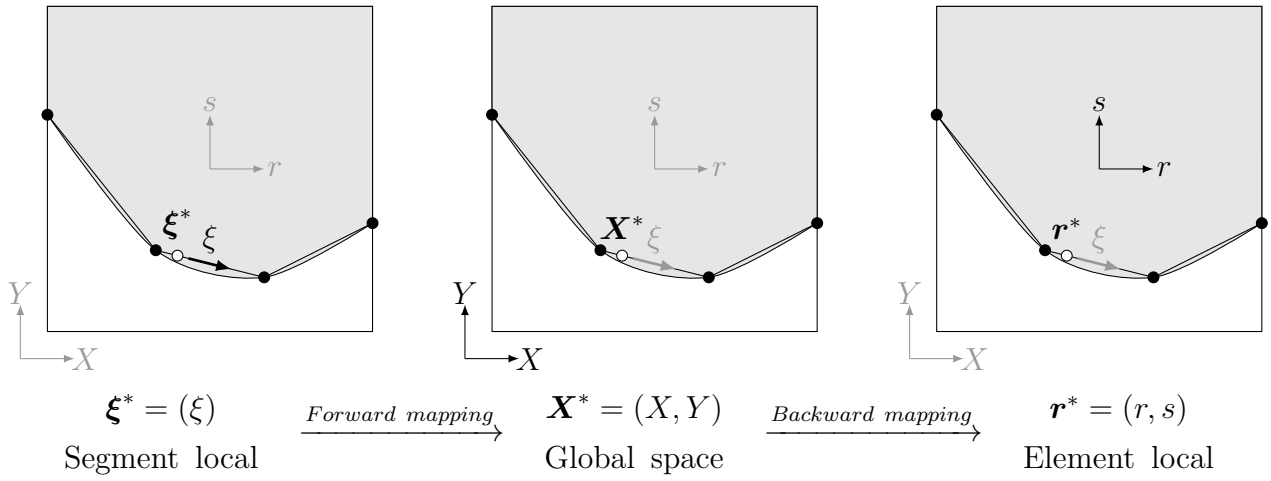


Figure 4.14: Coordinate transformations in the case of a geometric model described by linear segments: Coordinates of integration points positioned in the local space of the segments (a) are first mapped forward to the undeformed global space (b) and then mapped backward to local space of the element (c).

the element local space. Thanks to the Cartesian structure of the FCM meshes, this mapping can be performed efficiently. Note that, to perform a numerical integration on the bodies' surfaces, parametric boundary descriptions are necessary. Obtaining these descriptions might represent an additional step in setting up a model. However, surfaces can be extracted much easier from geometric models, than volumetric meshes. The previous chapter introduced some methods to recover such surface descriptions in an automated manner.

4.6.1 Essential boundary conditions

Dirichlet boundary conditions prescribe values $\bar{\boldsymbol{u}}$ for the displacement solution $\boldsymbol{u}$ on the (initial) boundary Γ_{D}. For a displacement-based formulation this can be written in a strong statement as

$$\boldsymbol{u}(\boldsymbol{X}) = \bar{\boldsymbol{u}}(\boldsymbol{X}) \quad \forall \boldsymbol{X} \in \Gamma_{\mathrm{D}} \tag{4.49}$$

The corresponding weak formulation can be obtained from the minimization of a constraining potential

$$\Pi_{\mathrm{D}}(\boldsymbol{u}) = \int_{\Gamma_{\mathrm{D}}} G(\bar{\boldsymbol{u}} - \boldsymbol{u}) \, \mathrm{d}A \quad \rightarrow \quad \min_{\boldsymbol{u}} , \tag{4.50}$$

where G represents the distinct constraint formulation. The method of Lagrange multipliers (LM) can exactly enforce the weak constraints. The corresponding constraining function reads

$$G = \boldsymbol{\lambda} \cdot (\bar{\boldsymbol{u}} - \boldsymbol{u}) \ . \tag{4.51}$$

As stated in Section 3.4.1, the Lagrange multiplier method has several properties, which complicate their usage. First, the method introduces additional unknowns, leading to more complicated data structures and which in turn may pose severe problems for an implementation within existing finite element codes. Furthermore, Lagrange multipliers generate a system of equations with a saddle point structure, restricting the choice of applicable solvers [10, 133]. A formulation that avoids these shortcomings is the penalty method. It can be recovered by identifying the LM with a penalty function

$$\boldsymbol{\lambda} = \tfrac{1}{2}\epsilon (\bar{\boldsymbol{u}} - \boldsymbol{u}) \ . \tag{4.52}$$

The resulting potential does not introduce any additional unknowns and hence maintains the original matrix structure

$$\Pi_{\mathrm{P}}^{\mathrm{D}}(\boldsymbol{u}) = \tfrac{1}{2} \int_{\Gamma_{\mathrm{D}}} \epsilon \, \delta\boldsymbol{u} (\bar{\boldsymbol{u}} - \boldsymbol{u}) \, \mathrm{d}A \quad \rightarrow \quad \min_{\boldsymbol{u}} \ . \tag{4.53}$$

The penalty method is very popular because of its simple implementation and robust behavior. Nonetheless, the penalty method slightly modifies the original problem and only recovers the solution of the original problem for an infinite penalty parameter ϵ [74], However, it leads to good results for reasonable parameter choices, i.e. without introducing severe conditioning problems [113, 112]. The variation and its linearization can be obtained in a straight-forward manner as

$$\delta\Pi_{\mathrm{P}}^{\mathrm{D}}(\boldsymbol{u}, \delta\boldsymbol{u}) = \int_{\Gamma_{\mathrm{D}}} \epsilon \, \delta\boldsymbol{u} \cdot (\bar{\boldsymbol{u}} - \boldsymbol{u}) \, \mathrm{d}A \ , \tag{4.54}$$

$$\Delta\delta\Pi_{\mathrm{P}}^{\mathrm{D}}(\boldsymbol{u}, \delta\boldsymbol{u}, \Delta\boldsymbol{u}) = \int_{\Gamma_{\mathrm{D}}} \epsilon \, \delta\boldsymbol{u} \cdot \Delta\boldsymbol{u} \, \mathrm{d}A \ . \tag{4.55}$$

The continuous formulations in Equations (4.54) and (4.55) can be discretized by introducing the approximated solution $\boldsymbol{u}^{\mathrm{h}}$ (Equation (4.12)). Then the contributions to the tangential stiffness matrix, the external and internal virtual work read

$$\boldsymbol{K}_{\mathrm{P}}^{\mathrm{D}} = \int_{\Gamma_{\mathrm{D}}} \epsilon\, \boldsymbol{N}^{\mathrm{T}} \boldsymbol{N}\, \mathrm{d}A \ , \tag{4.56a}$$

$$\boldsymbol{f}_{\mathrm{P}}^{\mathrm{ext,D}} = \int_{\Gamma_{\mathrm{D}}} \epsilon\, \boldsymbol{N}^{\mathrm{T}} \bar{\boldsymbol{u}}\, \mathrm{d}A \ , \tag{4.56b}$$

$$\boldsymbol{f}_{\mathrm{P}}^{\mathrm{int,D}} = \int_{\Gamma_{\mathrm{D}}} \epsilon\, \boldsymbol{N}^{\mathrm{T}} \boldsymbol{u}\, \mathrm{d}A \ . \tag{4.56c}$$

Here, $\boldsymbol{N}$ denotes the matrix of shape functions (4.13) introduced in Section 4.2.

4.6.2 Mesh coupling

The strong form for coupling two displacement fields $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ spanned by different discretizations (meshes) is stated as

$$\boldsymbol{u}^1(\boldsymbol{X}) = \boldsymbol{u}^2(\boldsymbol{X}) \quad \forall \boldsymbol{X} \in \Gamma_{\mathrm{I}} \ , \tag{4.57}$$

where Γ_{I} denotes the interface boundary shared by both meshes. A constraining function using Lagrange multipliers can be stated as

$$G = \boldsymbol{\lambda}\left(\boldsymbol{u^1} - \boldsymbol{u^2}\right) . \tag{4.58}$$

Again, the Lagrange multiplier field can be regularized by imposing a penalty function leading to the following weak constraining potential

$$\Pi_{\mathrm{P}}^{\mathrm{I}}\left(\boldsymbol{u}^1, \boldsymbol{u}^2\right) = \tfrac{1}{2}\epsilon \int_{\Gamma_{\mathrm{I}}} \left(\boldsymbol{u}^1 - \boldsymbol{u}^2\right)^2 \mathrm{d}A \quad \rightarrow \quad \min_{\boldsymbol{u}^1, \boldsymbol{u}^2} \ , \tag{4.59}$$

which results in the following variation

$$\delta\Pi_{\mathrm{P}}^{\mathrm{I}}\left(\boldsymbol{u}^1, \delta\boldsymbol{u}^1, \boldsymbol{u}^2, \delta\boldsymbol{u}^2\right) = \epsilon \int_{\Gamma_{\mathrm{I}}} \delta\boldsymbol{u}^1 \cdot \left(\boldsymbol{u}^1 - \boldsymbol{u}^2\right) - \delta\boldsymbol{u}^2 \cdot \left(\boldsymbol{u}^1 - \boldsymbol{u}^2\right) \mathrm{d}A \ . \tag{4.60}$$

The corresponding linearization reads

$$\Delta\delta\Pi_{\mathrm{P}}^{\mathrm{I}}\left(\boldsymbol{u}^1, \delta\boldsymbol{u}^1, \Delta\boldsymbol{u}^1, \boldsymbol{u}^2, \delta\boldsymbol{u}^2, \Delta\boldsymbol{u}^2\right) = \int_{\Gamma_{\mathrm{I}}} \delta\boldsymbol{u}^1 \cdot \left(\Delta\boldsymbol{u}^1 - \Delta\boldsymbol{u}^2\right) - \delta\boldsymbol{u}^2 \cdot \left(\Delta\boldsymbol{u}^1 - \Delta\boldsymbol{u}^2\right) \mathrm{d}A \ . \tag{4.61}$$

These weak coupling formulations can be discretized using approximate solutions, leading to contributions to the tangential stiffness as well as the internal virtual work, which read

$$\boldsymbol{K}_{\mathrm{P}}^{\mathrm{I}} = \int_{\Gamma_{\mathrm{I}}} \epsilon\, \tilde{\boldsymbol{N}}^{\mathrm{T}} \tilde{\boldsymbol{N}}\, \mathrm{d}A \ , \tag{4.62a}$$

$$\boldsymbol{f}_{\mathrm{P}}^{\mathrm{int,I}} = \int_{\Gamma_{\mathrm{I}}} \epsilon\, \tilde{\boldsymbol{N}}^{\mathrm{T}} \tilde{\boldsymbol{u}}\, \mathrm{d}A \ . \tag{4.62b}$$

Here, the combined shape function matrix and discrete solution vector are defined as

$$\tilde{\boldsymbol{N}} = \begin{bmatrix} \boldsymbol{N}_1 & -\boldsymbol{N}_1 \\ -\boldsymbol{N}_2 & \boldsymbol{N}_2 \end{bmatrix} \tag{4.63a}$$

and

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \hat{\boldsymbol{u}}_1 \\ \hat{\boldsymbol{u}}_2 \end{bmatrix} \ . \tag{4.63b}$$

Note that there is no contribution to the external virtual work, as no (externally) prescribed value is involved in the formulation.

Chapter 5

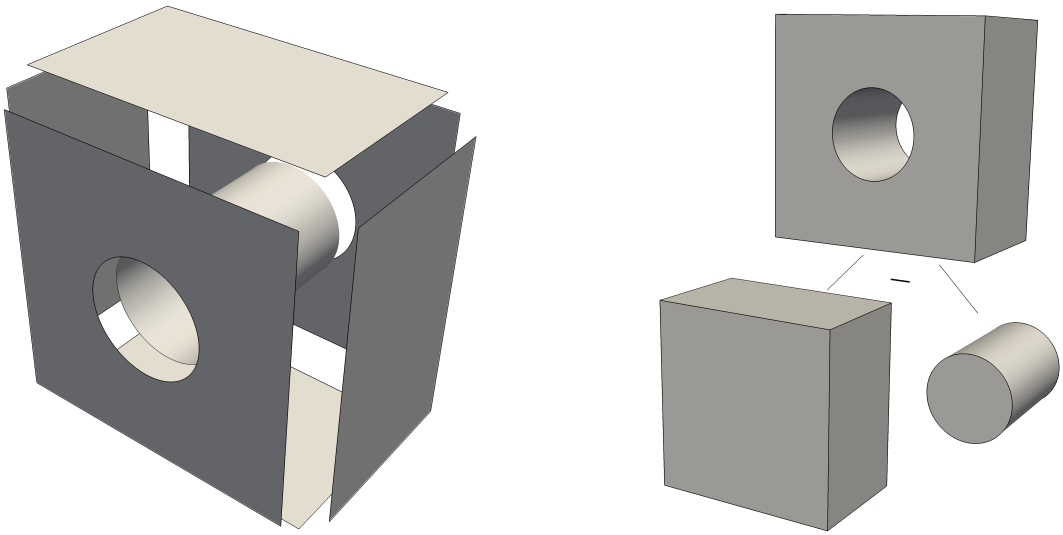
Recovery of embedded surfaces*

The FCM, outlined in the previous chapter, can be applied for any geometry description that provides an inside–outside (point-in-membership) test, including implicitly defined descriptions like e.g. CSG models, CT-data and level sets [115, 88]. Thus, a parametric boundary description may not be available from the beginning. Such a description is necessary, however, to numerically integrate Neumann boundary conditions, weak Dirichlet boundary conditions or contact constraints. Therefore, this chapter introduces an algorithm to *automatically* recover an explicit boundary surface that is conforming to the employed discretization and thus suitable for analysis. This algorithm is an extension of the well-known Marching Squares/Cubes (MS, MC) algorithm [73]. To improve the quality of recovered surfaces, the linear approximations can be transformed into high-order (curved) representations, see also [123, 39]. This is closely related to a p-extension known from finite elements. Thus, a generic formulation is introduced to recover high-order surfaces for implicit geometric models using polynomial Lagrange curves and triangles in two- and three-dimensional settings respectively (Appendix A). In the context of fictitious domain methods, like the FCM, automatic surface recovery is not only useful for the application of boundary conditions on embedded interfaces. The algorithms mentioned earlier can also be used to obtain the physical surface element-wise within a fictitious domain approach, to e.g. quadrature rules with the help of moment fitting (Chapter 4) and for post-processing with controllable surface resolutions.

5.1 Geometric models

Different geometric models are available to represent physical models in a form suitable for computational processing. The prevalent models used in Computer Aided Design (CAD) are Boundary Representations (B-Rep) and Constructive Solid Geometries (CSG) as shown in Figures 5.1a and 5.1b. B-Rep models *explicitly* represent three-dimensional bodies by their surfaces [76]. This is especially advantageous for visualization purposes, and normally provides all the information a designer needs. Unfortunately, these models pose some problems, when it comes to creating finite element meshes. B-Reps models are often non-watertight or contain modeling errors, which are unproblematic for visualization but may lead to failure of

* The major part of this chapter has been submitted for publication in T. Bog, N. Zander, S. Kollmannsberger, and E. Rank, "Weak imposition of frictionless contact constraints on automatically recovered high-order, embedded interfaces using the finite cell method," *Computational Mechanics*, submitted, Jan. 2017.



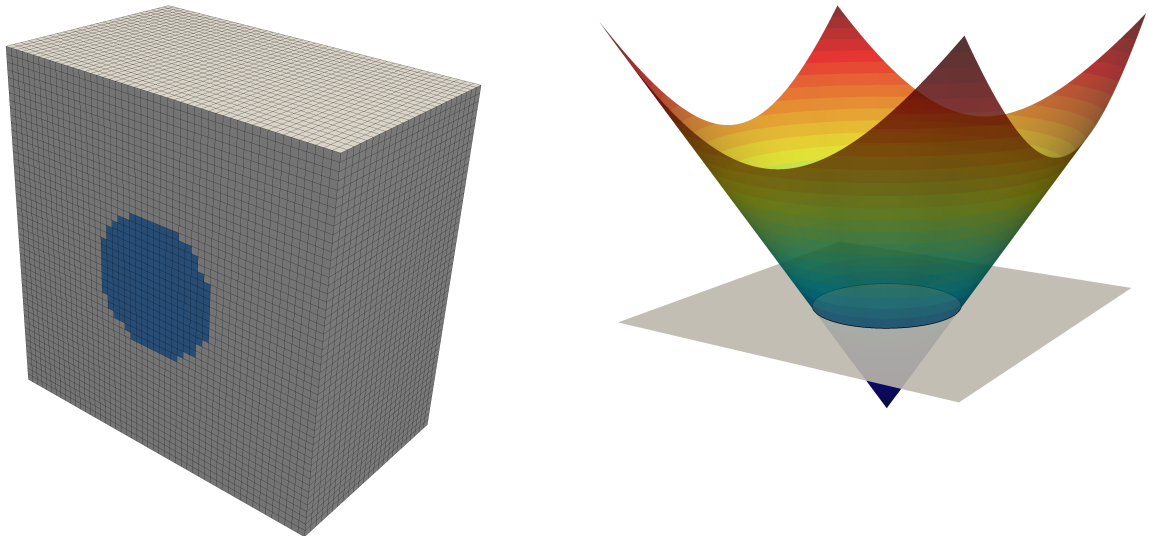
(a) Boundary Representation (B-Rep) **(b)** Constructive Solid Geometry (CSG)

Figure 5.1: Prominent geometric models in Computer Aided Design (CAD).

meshing algorithms [40] or insufficient mesh quality. In fact, a study carried out by Sandia National Laboratories [23] determined that the major part of the time spent on the creation of simulation model is spent on geometry cleanup and simplification. In contrast, CSG-models are constructed from primitives using Boolean operations [76]. Consequently, these models are always valid. Since point-in-membership tests can be performed very efficiently, CSG models are well suited for fictitious domain simulation approaches like the finite cell method [128] (Chapter 4). Unfortunately, CSG models only represent geometries *implicitly*. Hence, body surfaces are generally not directly available. While, CAD kernels that employ a procedural modeling approach based on CSG can simply generate a boundary representation, these descriptions again suffer from the same shortcomings as original B-Rep models. An interesting approach to combine the ideas of B-Rep and CSG to create of more powerful volumetric geometric description was recently presented by Massarwi and Elber [78].

Voxel models obtained from CT scans represent another implicit geometric representations (Figure 5.2a). These models can be thought of as a stack of images defining geometric objects in 3D-space. The images encode the density of the original geometry using different gray-values for each voxel. With a growing interest in bio-mechanical simulation, voxel models are also becoming more important in the finite element community. Again, fictitious domain methods like the FCM are optimally suited to directly perform simulations on these kind of models [137, 104]. Surface recovery for voxel models is well-established, e.g. using the Marching Cubes algorithm [73], which recovers iso-surfaces using a linear triangulation. However, since voxels only represent a zeroth order approximation of the geometry, the surfaces resulting from the Marching Cubes algorithm still contain the characteristic "block-wise" structure. To overcome this problem, Theisel et al. [123] presented an approach to recover iso-surfaces using G^1-continuous patches.

Another alternative to represent physical bodies with a volumetric field are *level-set* methods [115, 88]. These methods span a discrete scalar field in 3D-space and implicitly describe



(a) Voxel model **(b)** Level-set representation of a hole

Figure 5.2: Implicit geometric descriptions.

geometric bodies by a distinct iso-value, usually the zero level-set (see Figure 5.2b).
Level-sets can be constructed to have the property of signed distance near the boundary, which makes them attractive for simulating discontinuities using the eXtended Finite Element Method (XFEM) [79, 13]. Depending on the choice of shape function, level-sets allow for smooth representation of geometric bodies. Verhoosel et al. [127] used a convolution approach to transform CT-based voxel models of trabecular bones into level-sets described by tri-variate NURBS-patches.

5.2 Recovery of linear surface approximations

Linear surface approximations represent the simplest description for geometric boundaries. In the case of B-Rep geometries, CAD-kernels can easily provide such descriptions by triangulating the respective surfaces. For implicit geometries like CSG, Voxel and Level set models the Marching Cubes is a well established algorithm.

5.2.1 The Marching Cubes algorithm

The MC algorithm [73, 18, 77] recovers the iso-surface of a scalar field by producing a triangular surface mesh (line segments in 2D). To this end, the volumetric domain is discretized into a grid of cubes, in the following denoted as recovery grid. For every cube that is cut by the domain boundary, a part of the total surface is recovered. The decision about what topology this mesh should have is based on the values of the scalar field evaluated at the corner vertices of the cube. Using these values, the surface topology is established using a lookup table containing 256 cases. By exploiting inherent symmetries, these 256 cases can be reduced to 16 fundamental cases.

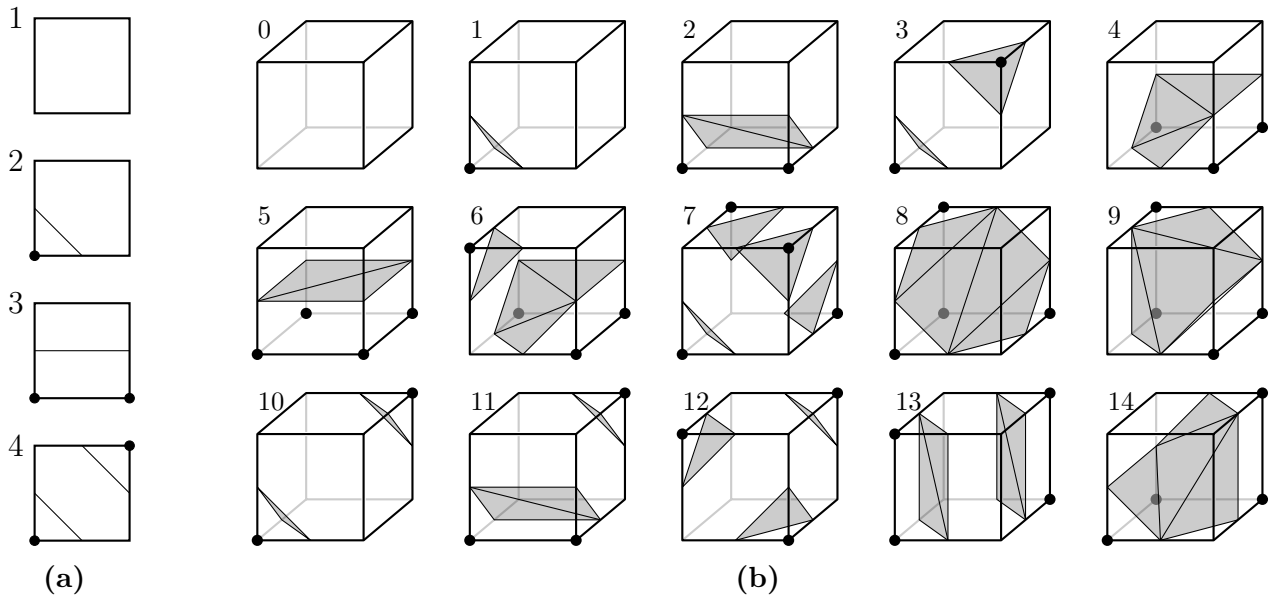


Figure 5.3: Fundamental (a) Marching Squares and (b) MC cases following [77].

Each vertex of the resulting triangular surface mesh lies on an edge of the recovery grid. The final vertex position is obtained by linearly interpolating the coordinates of the edge vertices with their scalar field values. Thus, the MC algorithm can be summarized as follows:

1. Define recovery grid (cubes)
2. Determine MC case for every cube
3. Linearly interpolate intersections using edge values
4. Create triangles

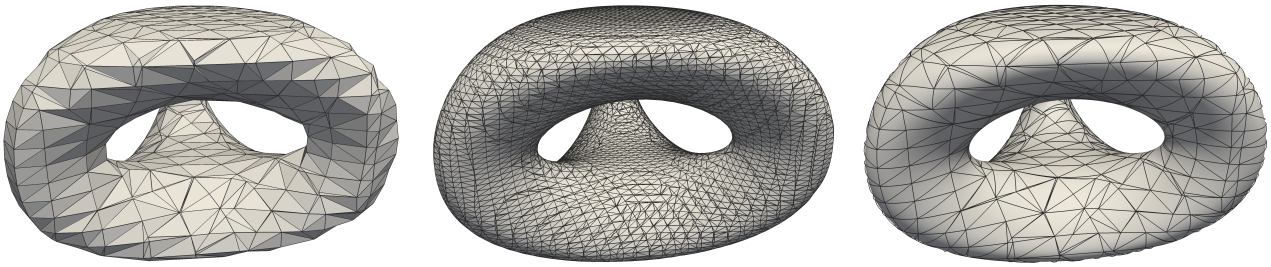
5.2.2 Ray intersections

The original MC algorithm works well for geometric models, which are defined by continuous scalar fields. To apply it also to CSG or B-Rep models, which are widely used in computer-aided engineering (CAE), the interpolation step needs to be extended to a ray-surface intersection. CSG models generally only provide the information whether a point lies inside or outside of the model [128]. Therefore, we utilize an inexact line search (*regula falsi*) between the respective edge vertices to find the ray-surface intersection within a given tolerance. In the case of B-Rep models, the ray-surface intersection is a standard procedure provided by most CAD systems. The intersection is generally obtained using Newton's method to minimize the distance between a point on the ray and a point on the intersecting surface [114].

5.3 High-order surface recovery for smooth surfaces

In this section we will discuss cases where the complete surface of a body is described by a smooth surface, i.e. without sharp edges or corners. The MC algorithm recovers the surface

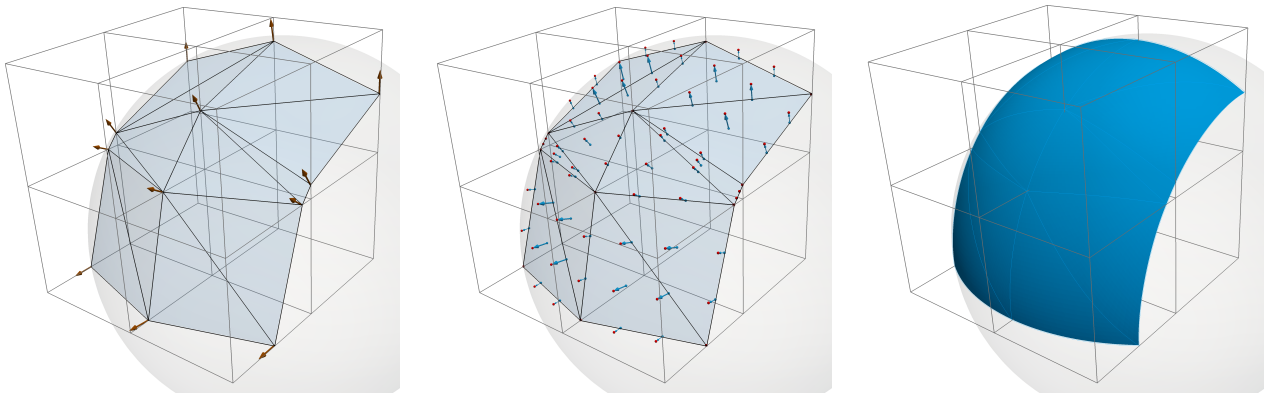
of a geometric body using linear triangles. Therefore, to accurately represent the surface of curved geometries, a large number of triangles is needed (Figures 5.4a and 5.4b).



(a) 1668 linear triangles **(b)** 17408 linear triangles **(c)** 1668 triangles of order $p = 4$

Figure 5.4: Boundary recovered for an implicit geometry using linear and high-order triangles.

This has a direct impact on the computational cost of a finite cell simulation and limits the achievable accuracy. To improve the resulting surface description, we propose to transform linear segments obtained from the MC algorithm into high-order barycentric Lagrange triangles [32] (curves in 2D), as shown in Figure 5.4c. In our work, this high-order recovery is done for every cut element in the finite cell mesh. The surface of such an element is first approximated by linear segments using a recovery grid with a given resolution. Afterwards, interpolation directions are computed at the vertices of the triangles (Figure 5.5a), where three different cases need to be considered, so that the resulting high-order surface does not contain any gaps.



(a) **(b)** **(c)**

Figure 5.5: High-order boundary reconstruction of spherical surface using a recovery grid of resolution $2 \times 2 \times 2$.
a) Search directions at segment vertices. b) Ray-surface intersection using interpolated search directions at seed points. c) Reconstruction using high-order triangles.

For those triangle vertices that do not touch the boundary of an element, the interpolation direction is computed by averaging the normals of the adjacent segments:

$$\boldsymbol{r}_{\text{avg}} = \frac{\sum_{i=1}^{n_{\text{seg}}} \boldsymbol{n}_i}{\left\| \sum_{i=1}^{n_{\text{seg}}} \boldsymbol{n}_i \right\|} \,. \tag{5.1}$$

The vector $\boldsymbol{r}_{\text{avg}}$ is the averaged interpolation direction and $\boldsymbol{n}_i$ is the normal vector of the i^{th} adjacent linear segment. For vertices that touch an edge of an element, the interpolation direction needs to be co-linear with that edge and has to point out of the domain:

$$\boldsymbol{r}_{\text{edge}} = \frac{\boldsymbol{t}_{\text{edge}} \cdot \boldsymbol{n}_{\text{avg}}}{\|\boldsymbol{t}_{\text{edge}} \cdot \boldsymbol{n}_{\text{avg}}\|} \cdot \boldsymbol{t}_{\text{edge}} \ . \tag{5.2}$$

Here, $\boldsymbol{t}_{\text{edge}}$ is the vector spanned by the edge. The interpolation directions for triangle vertices touching a face of an element are projected onto that (planar) face. This is achieved by using the averaged search direction $\boldsymbol{n}_{\text{avg}}$ and subtracting the component orthogonal to the face:

$$\boldsymbol{r}_{\text{face}} = \boldsymbol{n}_{\text{avg}} - \frac{\boldsymbol{n}_{\text{avg}} - \boldsymbol{n}_{\text{face}}}{\|\boldsymbol{n}_{\text{face}}\|} \cdot \boldsymbol{n}_{\text{face}} \ , \tag{5.3}$$

where, $\boldsymbol{n}_{\text{face}}$ is the normal vector of a face. In the next step, seed points are distributed in the index space of each linear segment, and a search direction is computed for every seed point by interpolating the search directions computed at the triangle vertices (Figure 5.5b). Using these search directions the control points of the final high-order segments are, depending on the geometrical model at hand, either computed by interpolation or by the ray-surface intersection presented in the previous section. Once the control points are obtained the high-order segments can be constructed (Figure 5.5c). A similar approach has been employed by Fries et al. [39] as an intermediate step in the high-order decomposition of level set geometries.
A major advantage of the Lagrange segments is that they can be easily deformed by modifying control point coordinates in an *updated-Lagrange* [12] fashion. This makes geometric operations during contact simulations, such as *closest-point-projections* [66], much cheaper, since the high-order ansatz spaces of the contact bodies do not have to be repeatedly evaluated.

5.3.1 Element conforming recovery

To reduce integration errors for weakly enforced boundary or interface conditions, boundary segments should not overlap element boundaries. The MC algorithm makes this possible without additional cost by choosing the resolution of the recovery grid to be a multiple of the mesh FCM grid resolution. In the case of a refined FCM-discretization, where elements have different sizes, the algorithm can be applied element-wise. This is the approach used in this work.

5.3.2 Sharp geometric features

CSG- or B-Rep-models often include sharp geometric features, such as edges or vertices, which are not correctly recovered by the MC algorithm and its high-order extension. Possible alternatives are the Dual Contouring [59, 106] and the Dual Marching Cubes [107] algorithms, which solve this problem at the cost of higher algorithmic complexity and computational effort. However, in the majority of cases the active contact interface does not contain sharp features, so that the algorithms presented here are widely applicable.

Chapter 6

Weakly enforced contact constraints*

S This chapter presents an approach to weakly enforce normal (frictionless) contact constraints in the framework of the FCM. All steps involved in a contact simulation will be described in detail, whenever they need special consideration to the fictitious domain approach employed by the FCM. Some researchers have already investigated contact problems involving embedded interfaces. However, they either restricted their investigations to low-order finite element formulations or simplified unilateral formulations. In [124] Tur et al. enforce contact on embedded interfaces using a stabilized Lagrange formulation in conjunction with linear and quadratic hexahedra. The stabilization parameter is found iteratively using a recovered stress field. The contact interface is obtained as the surface of a volumetric triangulation used to numerically integrate cut elements. Andreykiv et al. [2] employ distributed Lagrange multipliers and a level set function between a conformingly meshed implant and a bone, which is treated with a fictitious domain approach. Most noteworthy regarding the FCM is the contribution by Konyukhov et al. [65], which presents three formulations to enforce contact constraints using the FCM. Their Cell-surface-to-analytical-surface (CSTAS) contact element is closely related to the formulation presented in this chapter.

As in classical contact simulations, the contacting bodies are grouped in pairs of master and slave bodies. Similar to *segment-to-segment* and *mortar* formulations, the numerical integration of the weak contact contributions is performed on the surface of the slave bodies. The necessary parametric surfaces may either be available directly, constructed manually using a CAD system or can be automatically recovered. A methodology for the latter case has been introduced in Chapter 5.

Once parametric surface descriptions are available for all bodies, a global search is performed to identify potential contact pairs. In the framework of the FCM these pairs are made up of a slave and a master boundary segment together with their respective embedding elements. In most cases, contact occurs only locally. Therefore, the global search is an important step to reduce the computational effort for a contact simulation.

During the local search, closest point projections are performed on every integration point of the slave surface to compute the gap value. When a negative gap value (penetration) is detected, the local contributions of the contact weak form need to be computed and added to the global system of equations.

* The major part of this chapter has been submitted for publication in T. Bog, N. Zander, S. Kollmannsberger, and E. Rank, "Weak imposition of frictionless contact constraints on automatically recovered high-order, embedded interfaces using the finite cell method," *Computational Mechanics*, submitted, Jan. 2017.

The contact formulation to be presented here is closely related to the weakly enforced Dirichlet conditions and the (Dirichlet) mesh coupling constraints presented in Chapter 4. In fact, these constraints can be considered as tied contact formulations. Unlike in the case of Dirichlet constraints, it is not always clear from the beginning, what shape contact interface will have eventually. The interface depends on the deformation, which renders the contact problem nonlinear.

6.1 Global search

The aim of the global search is to identify a set potential contact pairs efficiently and to avoid unnecessary work on the local level. Within this work we denote contact pairs as a combination of two geometric segments, and their associated elements, each associated with either the slave or the master surface of the contact bodies [133].
The global search used in computational contact mechanics is also known as collision detection, in the computer graphics and computer vision communities. Collision detection plays an important role in many practical applications, which is why a large number of algorithms are available [30, 54]. The right choice of algorithm depends on the run time complexity, memory requirements and implementation effort the user is willing to accept. For an overview of collision detection algorithms, the reader is referred to Ericson [30] and Wriggers et al. [133]. Within the scope of this work, a brute force algorithm based on bounding spheres is employed. This algorithm is robust, easy to implement and has only low memory requirements. However, it has a theoretical complexity of $\mathcal{O}\left(n^2\right)^{\dagger}$, which does not make it a good candidate for explicit or real-time simulations. The basic idea of the brute force search is to center bounding spheres, or circles in 2D, at the current (deformed) center points $\boldsymbol{x}_{(\cdot)}^{\text{center}}$ of each segment of the slave and master surfaces. (Figure 6.1). The radius of each sphere is either provided by the user or can be computed based on the segment size. For the latter case, one possibility is to use the maximum distance between the undeformed center point $\boldsymbol{X}_{(\cdot)}^{\text{center},i}$ of a segment i and all its n_{CP} control points $\boldsymbol{X}_{\text{CP}}^{j}$

$$r_i = \max\left(\left\|\boldsymbol{X}_{(\cdot)}^{\text{center},i} - \boldsymbol{X}_{\text{CP}}^{j}\right\|\right), \quad j \in \{1, \ldots, n_{\text{CP}}\} \ . \tag{6.1}$$

To find potential contact pairs, each master sphere is checked against each slave sphere for overlap. The overlap Δr^{ij} of a pair $i - j$ is simply computed as

$$\Delta r^{ij} = \left(r_{\text{M}}^{i} + r_{\text{S}}^{j}\right) - \left\|\boldsymbol{x}_{\text{M}}^{\text{center},i} - \boldsymbol{x}_{\text{S}}^{\text{center},j}\right\| \ . \tag{6.2}$$

Whenever a positive overlap is detected, the respective master and slave segments are stored with their embedding elements. In the case of a self-contact simulation master and slave surfaces contain the same elements. Here, contact segments will be paired with themselves and their direct neighbors, when using large bounding spheres. This can be avoided, by additionally ensuring that contact pairs do not have any initial (undeformed) overlaps

$$\Delta R^{ij} = \left(r_{\text{M}}^{i} + r_{\text{S}}^{j}\right) - \left\|\boldsymbol{X}_{\text{M}}^{\text{center},i} - \boldsymbol{X}_{\text{S}}^{\text{center},j}\right\| \ . \tag{6.3}$$

[†]The big O notation describes how the computational run time is affected by the number of inputs n.

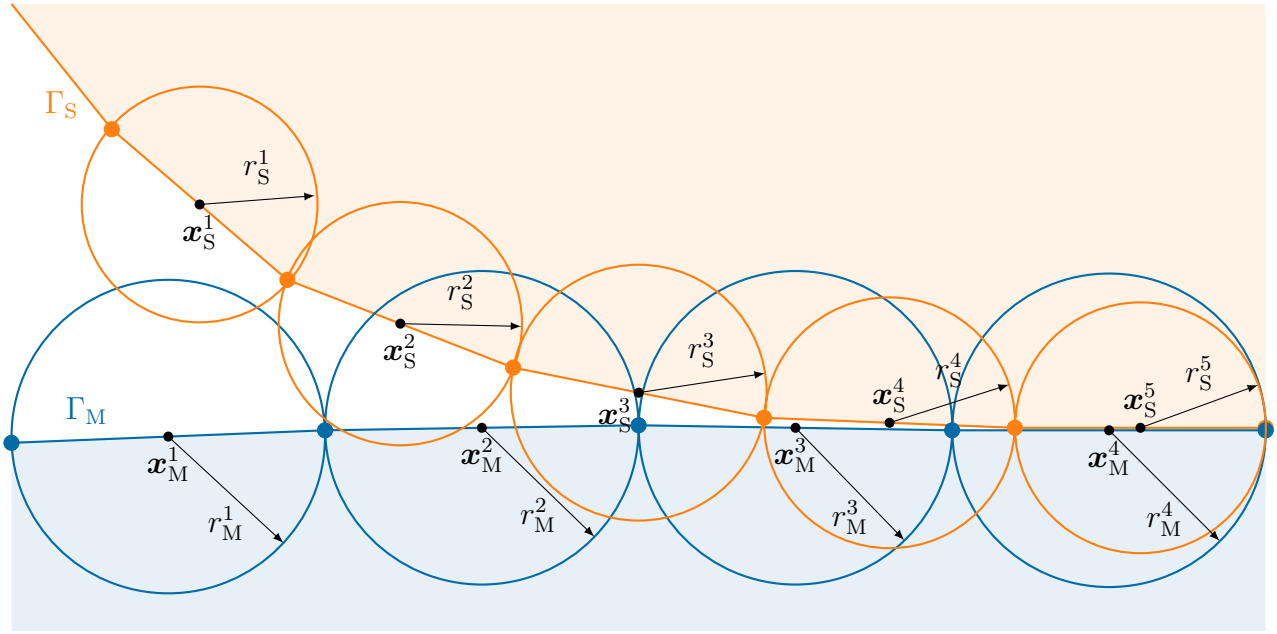


Figure 6.1: Basic idea of bounding sphere collision detection.

6.2 Local search

The local search tries to find the closest point projection (CPP) $\boldsymbol{x}_\mathrm{M}$ of a deformed slave point $\boldsymbol{x}_\mathrm{S}$ onto the master surface $\Gamma_{\mathrm{C,M}}$ (Chapter 3). Using the projection result, the associated surface normal and the gap at $\boldsymbol{x}_\mathrm{M}$ can be computed as

$$\boldsymbol{n}_\mathrm{M} = \frac{\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}}{\|\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}\|} \tag{6.4}$$

$$g = (\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}) \cdot \boldsymbol{n}_\mathrm{M} \ . \tag{6.5}$$

In the scope of this work, the local search is carried out during the numerical integration of the contact contributions on the slave boundary. Hence, a CPP is performed for every quadrature point $\boldsymbol{\xi}_\mathrm{S}^i$ of a slave segment. The associated undeformed slave point $\boldsymbol{X}_\mathrm{S}$ is obtained by a forward-mapping from the current segment at $\boldsymbol{\xi}_\mathrm{S}^i$ to the global space. Using the slave element associated with the current segment, obtained in the global search (Section 6.1), the deformed slave point $\boldsymbol{x}_\mathrm{S}$ is computed as

$$\boldsymbol{x}_\mathrm{S}\left(\boldsymbol{\xi}_\mathrm{S}^i\right) = \boldsymbol{X}_\mathrm{S}\left(\boldsymbol{\xi}_\mathrm{S}^i\right) + \boldsymbol{u}_e\left(\boldsymbol{r}_e\left(\boldsymbol{X}_\mathrm{S}\left(\boldsymbol{\xi}_\mathrm{S}^i\right)\right)\right) \ . \tag{6.6}$$

Here, $\boldsymbol{u}_e$ denotes the displacements associated with an element e at $\boldsymbol{X}_\mathrm{S}^i$ and $\boldsymbol{r}_e$ are the parametric coordinates of the element at $\boldsymbol{X}_\mathrm{S}$. These mappings have been introduced in Chapter 4. As a result of the global search (Section 6.1) the CPP for a given slave point only needs to be performed on a limited number of master segments. On an arbitrarily shaped (master) segment, the CPP can be formulated as the following minimization problem [66]

$$F\left(\boldsymbol{x}_\mathrm{M}\left(\boldsymbol{\xi}_\mathrm{M}\right)\right) = \frac{1}{2}\left(\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}\left(\boldsymbol{\xi}_\mathrm{M}\right)\right) \cdot \left(\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}\left(\boldsymbol{\xi}_\mathrm{M}\right)\right) \rightarrow \min \ , \tag{6.7}$$

were $\boldsymbol{\xi}_\mathrm{M}$ denotes the parametric coordinates of the master surface. To solve this minimization problem Newtons method is employed to find the zeros of the residual

$$\boldsymbol{F}' = \frac{\partial F}{\partial \boldsymbol{\xi}_\mathrm{M}} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{M},1} \cdot (\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}) \\ \boldsymbol{x}_{\mathrm{M},2} \cdot (\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}) \end{bmatrix} = \boldsymbol{0} \ . \tag{6.8}$$

Therefore, the linearization of Equation (6.8) is necessary, which reads

$$\begin{aligned} \boldsymbol{F}'' &= \begin{bmatrix} \frac{\partial^2 F}{\partial \xi^1_\mathrm{M} \partial \xi^1_\mathrm{M}} & \frac{\partial^2 F}{\partial \xi^1_\mathrm{M} \partial \xi^2_\mathrm{M}} \\ \frac{\partial^2 F}{\partial \xi^2_\mathrm{M} \partial \xi^1_\mathrm{M}} & \frac{\partial^2 F}{\partial \xi^2_\mathrm{M} \partial \xi^2_\mathrm{M}} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{x}_{\mathrm{M},1} \cdot \boldsymbol{x}_{\mathrm{M},1} - \boldsymbol{x}_{\mathrm{M},11} \cdot (\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}) & \boldsymbol{x}_{\mathrm{M},1} \cdot \boldsymbol{x}_{\mathrm{M},2} - \boldsymbol{x}_{\mathrm{M},12} \cdot (\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}) \\ \boldsymbol{x}_{\mathrm{M},2} \cdot \boldsymbol{x}_{\mathrm{M},1} - \boldsymbol{x}_{\mathrm{M},21} \cdot (\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}) & \boldsymbol{x}_{\mathrm{M},2} \cdot \boldsymbol{x}_{\mathrm{M},2} - \boldsymbol{x}_{\mathrm{M},22} \cdot (\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}) \end{bmatrix} \ . \end{aligned} \tag{6.9}$$

Note that the terms $\boldsymbol{x}_{\mathrm{M},ij} \cdot (\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M})$ can incur a significant computational overhead due to the second derivatives $\boldsymbol{x}_{\mathrm{M},ij}$. Therefore, these terms can be neglected at the price of a few more (cheaper) iterations. Newton's method iteratively updates the parametric coordinates $\boldsymbol{\xi}_\mathrm{M}$ starting from an initial guess until convergence is achieved. A robust approach to get an initial guess is to distribute points in the index space of the master segment in a tensor product structure. Then all points are mapped forward, and the point whose mapping lies closest to the projection point $\boldsymbol{x}_\mathrm{S}$ is chosen as an initial guess.
The projection can be applied to all high-order surfaces (and curves in 2D). This also includes high-order barycentric triangles (Appendix A), if the parametric Cartesian coordinates ξ^1 and ξ^2 are identified with the barycentric coordinates τ_2 and τ_3, respectively. However, there are some simple geometric formulations such as lines and planar triangles for which closed form solutions are available (see Appendix B).
Note that while the deformed slave point $\boldsymbol{x}_\mathrm{S}$ is fixed throughout the projection iterations, the current master point $\boldsymbol{x}_\mathrm{M}$ and its derivatives need to be repeatedly evaluated. This involves a significant number of forward mappings from the master boundary, backward mappings to the embedding element and evaluations of the displacement solution and its derivatives. Especially the latter part can become very expensive for high-order finite elements. Therefore, within this work the segments of the master surface are explicitly deformed in an Updated Lagrange fashion. This is straightforward since we employ Lagrange segments of arbitrary order to represent boundaries (see Chapter 5). The updated control point coordinates in a segment at a solution step $i+1$ can be computed as

$$\boldsymbol{x}^{i+1}_\mathrm{CPP} = \boldsymbol{x}^{i}_\mathrm{CPP} + \boldsymbol{u}^{i}_e \left(\boldsymbol{r}_e \left(\boldsymbol{X}_\mathrm{CPP} \left(\boldsymbol{\xi}_\mathrm{CPP} \right) \right) \right) \ . \tag{6.10}$$

where the same mappings as in Equation (6.6) are utilized.

6.2.1 Existence and uniqueness

In some cases, especially for linear boundaries, there might be no solution to the CPP since the boundary contains kinks (Figure 6.2a). In this case projections to lower dimensional geometric entities, such as edges and points, need to be performed [66, 62]. The use of high-order segments greatly alleviates this problem, since the resulting normal field is much smoother. There might

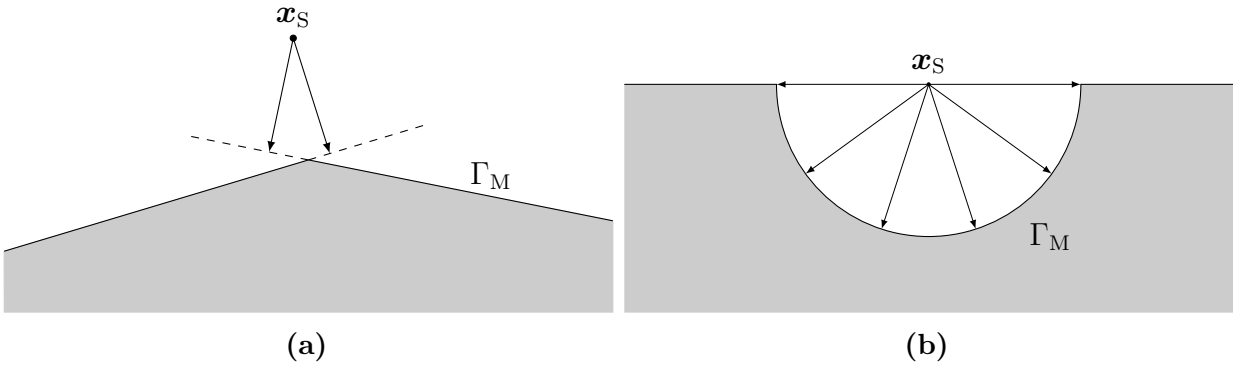


Figure 6.2: Critical cases for closest point projections.

also be cases where the resulting projection is not uniquely defined (Figure 6.2b). However, these problems get less severe once both contact boundaries close up on each other during the solution process [133].

6.3 Constraint enforcement

This section covers the formulation of the weakly enforced, frictionless (normal) contact constraints. To this end, the constraint potential is introduced and developed into a discretized linearized weak form. Furthermore, the numerical integration of the contact contributions as well as some simplifications of the general formulation are discussed.

6.3.1 Constraint potential

The constraint potential along a contact boundary Γ_C for the penalty regularization of normal contact is given by [133]

$$W_\mathrm{C}^\mathrm{N}(\boldsymbol{u}) = \frac{1}{2}\int_{\Gamma_\mathrm{C}} \epsilon_\mathrm{N} H(-g)\, g^2 \mathrm{d}A \ . \tag{6.11}$$

Here ϵ_N is the penalty parameter for the normal contact traction and H is the Heaviside function defined as

$$H(x) = \begin{cases} 0.0 & \text{if } x < 0 \\ 1.0 & \text{if } x \geq 0 \ . \end{cases} \tag{6.12}$$

The gap function g between two bodies is defined as

$$g = (\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{M}) \cdot \boldsymbol{n}_\mathrm{M} \ , \tag{6.13}$$

where $\boldsymbol{x}_\mathrm{S}$ and $\boldsymbol{x}_\mathrm{M}$ denote the coordinates of a point on the slave surface and its projection onto the master surface, respectively [133]. $\boldsymbol{n}_\mathrm{M} = [n_x,\ n_y,\ n_z]$ is the normal vector on the master surface evaluated at $\boldsymbol{x}_\mathrm{M}$. Both coordinates $\boldsymbol{x}_\mathrm{M}$ and $\boldsymbol{x}_\mathrm{S}$ are given in the current configurations [17] and the existence of a closest point projection is assumed [66].

6.3.2 Weak form of the contact constraint

Too obtain a weak formulation leading to a finite element discretization we compute the variation of (6.11), which is equivalent to computing the directional derivative with respect to the variation of the deformation $\delta\boldsymbol{u}$:

$$\delta W_{\mathrm{C}}^{\mathrm{N}}\left(\boldsymbol{u},\delta\boldsymbol{u}\right)=D_{\delta\boldsymbol{u}}\left[W_{\mathrm{C}}^{\mathrm{N}}\left(\boldsymbol{u}\right)\right]=\int_{\Gamma_{\mathrm{C}}}\epsilon_{\mathrm{N}}H\left(-g\right)g\;\delta g\,\mathrm{d}A=0\;. \tag{6.14}$$

Here, $\delta\left(\cdot\right)$ and $D_{(\cdot)}$ denote the variation and the directional derivative, respectively. The variation of the gap function δg (Section 3.1) reads

$$\delta g=\left(\delta\boldsymbol{x}_{\mathrm{S}}-\delta\boldsymbol{x}_{\mathrm{M}}\right)\cdot\boldsymbol{n}_{\mathrm{M}}=\left(\delta\boldsymbol{u}_{\mathrm{S}}-\delta\boldsymbol{u}_{\mathrm{M}}\right)\cdot\boldsymbol{n}_{\mathrm{M}}\;, \tag{6.15}$$

where the fact is utilized that the initial configuration does not depend on the displacement

$$\delta\boldsymbol{x}_{(\cdot)}=\underbrace{\delta\boldsymbol{X}_{(\cdot)}}_{=0}+\delta\boldsymbol{u}_{(\cdot)}\;. \tag{6.16}$$

Note that a dependence of the integration domain on the deformation has been neglected at this point, i.e. we assume that the contact boundary does not change its length significantly.

6.3.3 Linearized weak form of the contact constraints

Since the final gap function is not known in advance, the resulting boundary value problem is rendered nonlinear even in the case of linear elasticity and small deformations. Therefore, the full Newton method will be applied to solve the global system of equations. To this end, the linearization of (6.14) is needed, which can be obtained using the directional derivative with respect to a displacement increment $\Delta\boldsymbol{u}$

$$\begin{aligned}\Delta\delta W_{\mathrm{C}}^{\mathrm{N}}\left(\boldsymbol{u},\delta\boldsymbol{u},\Delta\boldsymbol{u}\right)&=D_{\delta\boldsymbol{u}}\left[\delta W_{\mathrm{C}}^{\mathrm{N}}\left(\boldsymbol{u},\delta\boldsymbol{u}\right)\right]\\&=\int_{\Gamma_{\mathrm{C}}}\epsilon_{\mathrm{N}}H\left(-g\right)\left[\Delta g\;\delta g+g\;\Delta\delta g\right]\mathrm{d}A=0\;.\end{aligned} \tag{6.17}$$

The linearization contains the two additional terms Δg and $\Delta\delta g$. The first term directly follows from Equation (6.15) because of the structural equivalence of variation and linearization

$$\Delta g=\left(\Delta\boldsymbol{x}_{\mathrm{S}}-\Delta\boldsymbol{x}_{\mathrm{M}}\right)\cdot\boldsymbol{n}_{\mathrm{M}}=\left(\Delta\boldsymbol{u}_{\mathrm{S}}-\Delta\boldsymbol{u}_{\mathrm{M}}\right)\cdot\boldsymbol{n}_{\mathrm{M}}\;. \tag{6.18}$$

The second term, the linearized variation of the gap $\Delta\delta g$, is much more involved and lead to very complex terms [133]. To simplify these matters Konyukhov et al. [67, 64] have formulated the linearization in a general, covariant form drawing on the concepts of differential geometry. They state the linearization of the weak form for normal contact as

$$\begin{aligned}&\Delta\delta W_{\mathrm{C}}^{\mathrm{N}}\left(\boldsymbol{u},\delta\boldsymbol{u},\Delta\boldsymbol{u}\right)=D_{\Delta\boldsymbol{u}}\left[\delta W_{\mathrm{C}}^{\mathrm{N}}\left(\boldsymbol{u},\delta\boldsymbol{u}\right)\right]\\&=\int_{\Gamma_{\mathrm{C}}}\epsilon_{\mathrm{N}}H\left(-g\right)\left(\delta\boldsymbol{u}_{\mathrm{S}}-\delta\boldsymbol{u}_{\mathrm{M}}\right)\cdot\left(\boldsymbol{n}_{\mathrm{M}}\otimes\boldsymbol{n}_{\mathrm{M}}\right)\left(\Delta\boldsymbol{u}_{\mathrm{S}}-\Delta\boldsymbol{u}_{\mathrm{M}}\right)\,\mathrm{d}A\end{aligned} \tag{6.19a}$$

$$
\begin{aligned}
&- \int_{\Gamma_\mathrm{C}} \epsilon_\mathrm{N} H\left(-g\right) g \left[\delta \boldsymbol{x}_{\mathrm{M},j} \cdot a^{ij} \left(\boldsymbol{n}_\mathrm{M} \otimes \boldsymbol{x}_{\mathrm{M},i}\right) \left(\Delta \boldsymbol{u}_\mathrm{S} - \Delta \boldsymbol{u}_\mathrm{M}\right)\right. \\
&\qquad \left. + \left(\delta \boldsymbol{u}_\mathrm{S} - \delta \boldsymbol{u}_\mathrm{M}\right) \cdot a^{ij} \left(\boldsymbol{x}_{\mathrm{M},j} \otimes \boldsymbol{n}_\mathrm{M}\right) \Delta \boldsymbol{u}_{\mathrm{M},i}\right] \mathrm{d}A \qquad (6.19\mathrm{b}) \\
&- \int_{\Gamma_\mathrm{C}} \epsilon_\mathrm{N} H\left(-g\right) g \left(\delta \boldsymbol{u}_\mathrm{S} - \delta \boldsymbol{u}_\mathrm{M}\right) \cdot h^{ij} \left(\boldsymbol{x}_{\mathrm{M},i} \otimes \boldsymbol{x}_{\mathrm{M},j}\right) \left(\Delta \boldsymbol{u}_\mathrm{S} - \Delta \boldsymbol{u}_\mathrm{M}\right) \mathrm{d}A \ , \qquad (6.19\mathrm{c})
\end{aligned}
$$

where derivatives with respect to a local parametric coordinate ξ_i inside the contact surface are represented by $(\cdot)_{,i}$

$$
\begin{aligned}
\boldsymbol{x}_{\mathrm{M},i} &= \frac{\partial \boldsymbol{x}_\mathrm{M}}{\partial \xi_i} = \frac{\partial \boldsymbol{x}_\mathrm{M}}{\partial \boldsymbol{X}} \cdot \frac{\partial \boldsymbol{X}}{\partial \xi_i} && (6.20\mathrm{a}) \\
\delta \boldsymbol{x}_{\mathrm{M},i} &= \frac{\partial \delta \boldsymbol{x}_\mathrm{M}}{\partial \xi_i} = \frac{\partial \delta \boldsymbol{u}_\mathrm{M}}{\partial \boldsymbol{X}} \cdot \frac{\partial \boldsymbol{X}}{\partial \xi_i} && (6.20\mathrm{b}) \\
\Delta \boldsymbol{x}_{\mathrm{M},i} &= \frac{\partial \Delta \boldsymbol{x}_\mathrm{M}}{\partial \xi_i} = \frac{\partial \Delta \boldsymbol{u}_\mathrm{M}}{\partial \boldsymbol{X}} \cdot \frac{\partial \boldsymbol{X}}{\partial \xi_i} \ . && (6.20\mathrm{c})
\end{aligned}
$$

The quantities a^{ij} and h^{ij} denote the metric and curvature tensors, respectively, given as

$$
a^{ij} = \frac{1}{a_D} \begin{bmatrix} a_{22} & -a_{21} \\ -a_{12} & a_{11} \end{bmatrix}, \qquad a_D = \det a_{ij} \quad \text{and} \quad a_{ij} = \boldsymbol{x}_{\mathrm{M},i} \cdot \boldsymbol{x}_{\mathrm{M},j} \qquad (6.21)
$$

and

$$
h^{ij} = h_{kl} a^{ik} a^{jl}, \qquad h_{ij} = \boldsymbol{x}_{\mathrm{M},ij} \cdot \boldsymbol{n}_\mathrm{M} \quad \text{and} \quad \boldsymbol{x}_{\mathrm{M},ij} = \frac{\partial^2 \boldsymbol{x}_\mathrm{M}}{\partial \xi_i \partial \xi_j} \ . \qquad (6.22)
$$

Note that the repeated indices i and j in the terms (6.19b) and (6.19c) denote a summation according to Einstein convention [27, 70]. The linearization given in Equations (6.19) can be decomposed into three characteristic parts. The term (6.19a) denotes the *main* part which is dominated by the normal penetration, (6.19b) and (6.19c) denote the *rotational* and the *curvature* parts, respectively, which take into account changes due to the rotation and the curvature of the master boundary during deformation. Konyukhov et al. [62, 63] show that the major contribution of Equation (6.19) stems from the main part (6.19a).

6.3.4 Finite element discretization

To arrive at a formulation suitable for finite element analysis the continuous quantities obtained in the previous section will now be discretized using the spaces introduced in Chapter 4. This will lead to a linear system of equations of the form

$$
\left(\boldsymbol{K}_\Omega + \boldsymbol{K}_\mathrm{C}\right) \Delta \tilde{\boldsymbol{u}} = \boldsymbol{f}_\Omega^\mathrm{ext} - \boldsymbol{f}_\Omega^\mathrm{int} - \boldsymbol{f}_\mathrm{C}^\mathrm{int} \ . \qquad (6.23)
$$

Here $\boldsymbol{K}_\Omega$ and $\boldsymbol{K}_\mathrm{C}$ are the tangent matrices stemming from the domain and the contact boundary, respectively. The discretized external and internal force vectors of the domain

are denoted by $\boldsymbol{f}_\Omega^{\text{ext}}$ and $\boldsymbol{f}_\Omega^{\text{int}}$, respectively. $\boldsymbol{f}_\text{C}^{\text{int}}$ is the contact residual following from the discretization of equation (6.14)

$$\delta W_\text{C}^{\text{N,h}} = \int_{\Gamma_\text{C}} \epsilon_\text{N} H\left(-g^\text{h}\right) g^\text{h}\, \delta g^\text{h}\, \mathrm{d}A \tag{6.24}$$

The variation of the discretized gap function reads

$$\begin{aligned}\delta g^\text{h} &= \left(\delta \boldsymbol{u}_\text{S}^\text{h} - \delta \boldsymbol{u}_\text{M}^\text{h}\right) \cdot \boldsymbol{n}_\text{M} \\ &= \left(\boldsymbol{N}_\text{S} \delta\tilde{\boldsymbol{u}}_\text{S} - \boldsymbol{N}_\text{M} \delta\tilde{\boldsymbol{u}}_\text{M}\right) \cdot \boldsymbol{n}_\text{M}\ ,\end{aligned} \tag{6.25}$$

where $\boldsymbol{N}_{(\cdot)}$ is the matrix of shape functions evaluated for body $(\cdot)$

$$\boldsymbol{N}_{(\cdot)} = \begin{bmatrix} \boldsymbol{N}_u & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{N}_v & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{N}_w \end{bmatrix}_{(\cdot)} = \begin{bmatrix} N_u^1 & \dots & N_u^k & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & N_v^1 & \dots & N_v^l & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & N_w^1 & \dots & N_w^m \end{bmatrix}_{(\cdot)} . \tag{6.26}$$

k, l and m are the number of modes for the components of the displacement solution $\boldsymbol{u} = [u, v, w]^\text{T}$. The coefficients of the variation of the displacement solution $\delta\tilde{\boldsymbol{u}}_{(\cdot)}$ are defined as

$$\delta\tilde{\boldsymbol{u}}_{(\cdot)} = \left[\delta u^1,\ \dots,\ \delta u^k,\ \delta v^1,\ \dots,\ \delta v^l,\ \delta w^1,\ \dots,\ \delta w^m\right]_{(\cdot)}^\text{T} . \tag{6.27}$$

Both the shape functions and the solution coefficients for master and slave bodies can be combined into $\tilde{\boldsymbol{N}}$ and $\delta\tilde{\boldsymbol{u}}$, respectively,

$$\tilde{\boldsymbol{N}} = \left[\boldsymbol{N}_\text{S}\ \ \boldsymbol{N}_\text{M}\right], \qquad \delta\tilde{\boldsymbol{u}} = \begin{bmatrix} \delta\tilde{\boldsymbol{u}}_\text{S} \\ \delta\tilde{\boldsymbol{u}}_\text{M} \end{bmatrix} . \tag{6.28}$$

Inserting these last two quantities into equation (6.24) leads to

$$\delta W_\text{C}^{\text{N,h}} = \delta\tilde{\boldsymbol{u}} \underbrace{\int_{\Gamma_\text{C}} \epsilon_\text{N} H\left(-g^\text{h}\right) g^\text{h} \tilde{\boldsymbol{N}}^\text{T} \boldsymbol{n}_\text{M}^\text{T}\, \mathrm{d}A}_{f_\text{C}^{\text{N,int}}} . \tag{6.29}$$

The tangent matrix of the contact constraints is composed of a main ($\boldsymbol{K}_\text{C}^{\text{N},m}$), a rotational ($\boldsymbol{K}_\text{C}^{\text{N},r}$) and a curvature part ($\boldsymbol{K}_\text{C}^{\text{N},c}$), following from equation (6.19):

$$\Delta\delta W_\text{C}^{\text{N,h}} = \delta\tilde{\boldsymbol{u}} \underbrace{\int_{\Gamma_\text{C}} \epsilon_\text{N} H\left(-g^\text{h}\right) \tilde{\boldsymbol{N}}^\text{T} \boldsymbol{n}_\text{M}^\text{T} \boldsymbol{n}_\text{M} \tilde{\boldsymbol{N}}\, \mathrm{d}A}_{\boldsymbol{K}_\text{C}^{\text{N},m}} \Delta\tilde{\boldsymbol{u}} \tag{6.30a}$$

$$- \delta\tilde{\boldsymbol{u}} \underbrace{\int_{\Gamma_\text{C}} \epsilon_\text{N} H\left(-g^\text{h}\right) g^\text{h}\ a^{ij} \left[\tilde{\boldsymbol{M}}_j^\text{T} \boldsymbol{n}_\text{M}^\text{T} \boldsymbol{x}_{\text{M},i} \tilde{\boldsymbol{N}} + \tilde{\boldsymbol{N}}^\text{T} \boldsymbol{x}_{\text{M},j}^\text{T} \boldsymbol{n}_\text{M} \tilde{\boldsymbol{M}}_i\right] \mathrm{d}A}_{\boldsymbol{K}_\text{C}^{\text{N},r}} \Delta\tilde{\boldsymbol{u}} \tag{6.30b}$$

$$- \delta\tilde{\boldsymbol{u}} \underbrace{\int_{\Gamma_\mathrm{C}} \epsilon_\mathrm{N} H\left(-g^\mathrm{h}\right) g^\mathrm{h}\ h^{ij} \tilde{\boldsymbol{N}}^\mathrm{T} \boldsymbol{x}_{\mathrm{M},i}^\mathrm{T} \boldsymbol{x}_{\mathrm{M},j} \tilde{\boldsymbol{N}}\, \mathrm{d}A}_{\boldsymbol{K}_\mathrm{C}^{\mathrm{N},c}}\ \Delta\tilde{\boldsymbol{u}}\ . \tag{6.30c}$$

Here, the coefficient vector for the linear displacement increments is defined as

$$\Delta\tilde{\boldsymbol{u}} = \begin{bmatrix} \Delta\tilde{\boldsymbol{u}}_\mathrm{S} \\ \Delta\tilde{\boldsymbol{u}}_\mathrm{M} \end{bmatrix}\ , \tag{6.31a}$$

where

$$\Delta\tilde{\boldsymbol{u}}_{(\cdot)} = \left[\Delta u^1,\ \ldots,\ \Delta u^k,\ \Delta v^1,\ \ldots,\ \Delta v^l,\ \Delta w^1,\ \ldots,\ \Delta w^m\right]_{(\cdot)}^\mathrm{T}\ . \tag{6.31b}$$

The matrix $\tilde{\boldsymbol{M}}_i$ contains the shape function derivatives of the master body with respect to the local parametric coordinate ξ_i

$$\tilde{\boldsymbol{M}}_i = \left[\boldsymbol{0}\ \ \boldsymbol{N}_{\mathrm{M},i}\right], \qquad \boldsymbol{N}_{\mathrm{M},i} = \frac{\partial \boldsymbol{N}_\mathrm{M}}{\partial \xi_i} = \frac{\partial N_\mathrm{M}^m}{\partial X^k}\frac{\partial X^k}{\partial \xi_i}\ . \tag{6.32}$$

Note that, just like in the case of weak Dirichlet boundary conditions and mesh constraints mentioned in Chapter 4, the shapes function values $\boldsymbol{N}$ are found and evaluated within the elements embedding the respective contact segment.

6.3.5 Unilateral contact against a rigid plane

The generic formulation, outlined in Section 3.5, can be simplified for the cases, where one of the contacting bodies is rigid. Many such formulations considering contact with analytic surfaces, such as spheres, cylinders or revolution surfaces, can be found in [62] and [64]. An even simpler case, which will be used for one of the numerical examples, is unilateral contact where the master surface is described by a rigid plane (Figure 6.3).
For this case, there is no additional fill-in, since no coupling of otherwise unrelated deformation fields is present. Also, since the rigid plane is chosen as the master surface, the rotational as well as the curvature part of Equation (6.30) vanish. Furthermore, this formulation fully omits global as well as local contact searches, and it nicely shows the connection to weakly enforced Dirichlet boundary conditions. The gap function for this case is formulated as

$$g = \left(x_i - x_i^0\right)\ , \tag{6.33}$$

where x_i is the current ith coordinate ($x_i \in \{x, y, z\}$) of a point on the boundary of the elastic body and x_i^0 is the offset value of the rigid plane in x_i-direction.
With the discrete variation and linearization of the gap function g given as

$$\delta g^\mathrm{h} = \boldsymbol{N}_i \cdot \delta\tilde{\boldsymbol{u}}_i \qquad \text{where} \qquad \delta\tilde{\boldsymbol{u}}_i \in \{\delta\tilde{\boldsymbol{u}},\ \delta\tilde{\boldsymbol{v}},\ \delta\tilde{\boldsymbol{w}}\} \tag{6.34a}$$

$$\Delta g^\mathrm{h} = \boldsymbol{N}_i \cdot \Delta\tilde{\boldsymbol{u}}_i \qquad \text{where} \qquad \Delta\tilde{\boldsymbol{u}}_i \in \{\Delta\tilde{\boldsymbol{u}},\ \Delta\tilde{\boldsymbol{v}},\ \Delta\tilde{\boldsymbol{w}}\} \tag{6.34b}$$

$$\text{and} \qquad \boldsymbol{N}_i \in \{\boldsymbol{N}_u,\ \boldsymbol{N}_v,\ \boldsymbol{N}_w\}\ , \tag{6.34c}$$

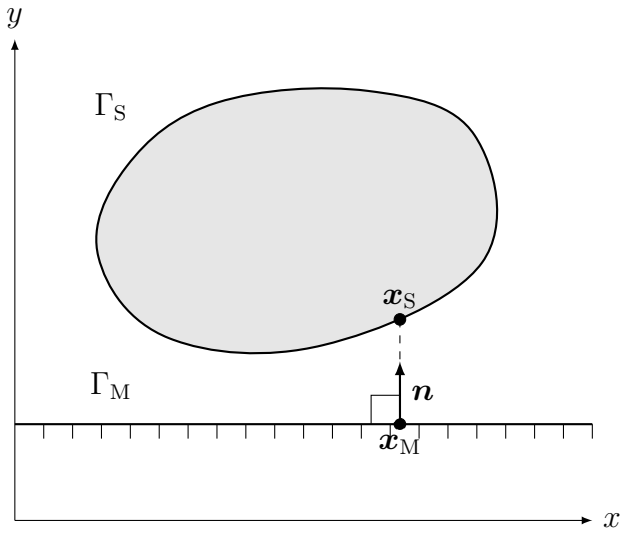


Figure 6.3: Unilateral contact between an elastic body and a rigid plane.

the discrete contact contributions read

$$\boldsymbol{K}_{\mathrm{C}} = \int_{\Gamma_{\mathrm{S}}} \epsilon_{\mathrm{N}} H\left(-g^{\mathrm{h}}\right) \boldsymbol{N}_i^{\mathrm{T}} \boldsymbol{N}_i \, \mathrm{d}A \tag{6.35a}$$

$$\boldsymbol{f}_{\mathrm{C}}^{\mathrm{int}} = \int_{\Gamma_{\mathrm{S}}} \epsilon_{\mathrm{N}} H\left(-g^{\mathrm{h}}\right) \boldsymbol{N}_i^{\mathrm{T}} \left(x_i - x_i^0\right) \, \mathrm{d}A \tag{6.35b}$$

Note that by omitting the Heaviside function in equations (6.35a) to (6.35b), component-wise, homogeneous weak Dirichlet boundary conditions based on the Penalty method are recovered.

6.3.6 Numerical integration

The numerical integration of the contact constraints is performed as a composed integration on the segmentation of the slave surface

$$\int_{\Gamma_{\mathrm{C}}} \boldsymbol{C}\left(\boldsymbol{\xi}\right) \mathrm{d}A \approx \sum_{i_{\mathrm{seg}}=1}^{n_{\mathrm{seg}}} \sum_{i_{\mathrm{qp}}=1}^{n_{\mathrm{qp}}} \boldsymbol{C}\left(\boldsymbol{\xi}_{i_{\mathrm{qp}}}\right) w_{i_{\mathrm{qp}}} \det \boldsymbol{J}_{i_{\mathrm{qp}}} \, . \tag{6.36}$$

Here, $\boldsymbol{C}$ is the integrand of interest stemming from either the discrete variation (6.29) or the discrete linearization (6.30) given in the previous section.
The main quantity of the contact contributions is the gap function g, which is only C^0 continuous along element edges. Hence, to enforce the contact constraints exactly, the slave surfaces would have to be segmented according to the projected master segments [94]. This is very costly, however, especially for three-dimensional applications. Furthermore, the segmentation would also only represent an approximation, since the projections of deformed polynomial edges onto an also deformed polynomial segment are non-rational. Therefore, we omit the segmentation and reduce the integration error by increasing the quadrature order and the

segmentation resolution. Integration points for which no projection can be found on the master segments belonging to the current contact element pair are skipped ([133]). This local *pointwise enforcement* of contact constraints is similar to the weighted integration that is utilized in the FCM for the volumetric parts (Chapter 4). One challenge for contact simulations with refined high-order meshes is the combination of contact segments embedded in sub-elements of different order and refinement level (Figure 6.4).

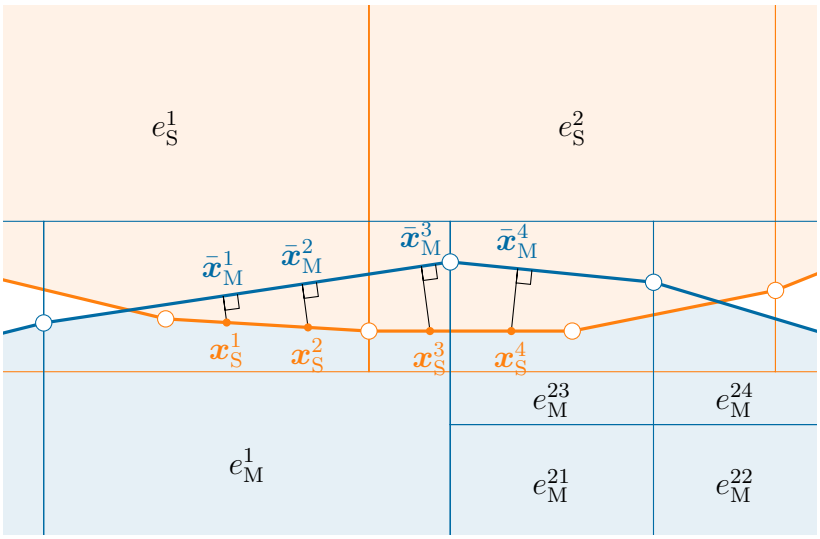


Figure 6.4: Integration point projection from the embedded slave surface $\Gamma_{\mathrm{S}}^{\mathrm{h}}$ to the embedded master surface $\Gamma_{\mathrm{M}}^{\mathrm{h}}$. Contact segments are denoted by solid lines bounded by a white dot. Integration points are denoted by smaller solid points. To efficiently account for the contact vectors and matrices resulting from the coupling of elements with different ansatz spaces, contact pairs are grouped according to their element pairs (e.g. $(e_{\mathrm{S}}^{1}, e_{\mathrm{M}}^{1})$, $(e_{\mathrm{S}}^{2}, e_{\mathrm{M}}^{1})$, $(e_{\mathrm{S}}^{2}, e_{\mathrm{M}}^{23})$).

Hence the discrete vectors and matrices of the contact contributions can have different sizes for each contact pair. Therefore, to efficiently process the contact integrals, we gather all segment pairs belonging to a pair of sub-elements. This way the number of matrix allocations and assembly operations of the contact contributions into the sparse global matrix can be kept to a minimum.

6.4 Numerical examples

The following numerical examples investigate the accuracy of the proposed formulation. Based on the methods presented in the previous sections, these examples will be set up using a common workflow:

1. Physical bodies are represented by CSG-models
2. FCM-discretizations are created based on the bounding box of each body (Chapter 4)
3. Contact interfaces are recovered for every element of a body's mesh (Chapter 5)
4. Weak contact constraints are applied on the recovered interfaces (Section 6.3) .

6.4.1 Unilateral Hertzian contact

As a first example, we consider the frictionless Hertzian contact problem [51] between an infinitely long elastic cylinder and a rigid plane.

6.4.1.1 Hertzian theory

In the Hertzian theory [51], the cylinder is subjected to a point load F leading to a contact zone of width $2b$, see Figure 6.5. Frictionless contact conditions are assumed and the resulting contact width shall be much smaller than the radius of the cylinder. The half width of the

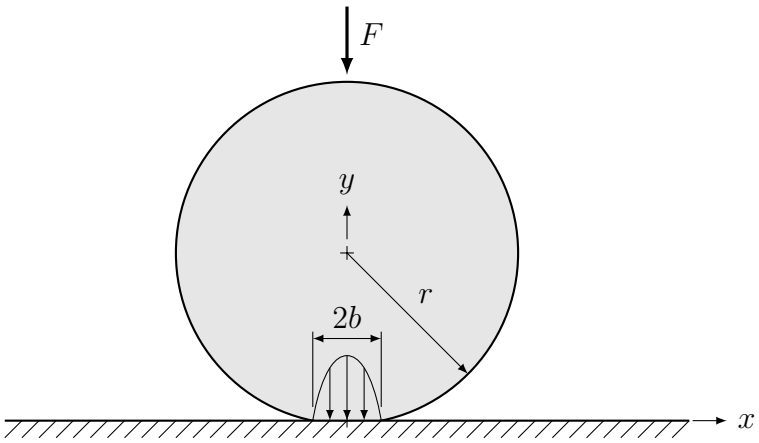


Figure 6.5: 2D Hertzian contact model between an elastic cylinder and a rigid plane.

contact interval b can be computed as

$$b = \sqrt{\frac{4 \cdot F \cdot r}{\pi \cdot \bar{E}}} \quad \text{with} \quad \bar{E} = \frac{E}{1 - \nu^2} \ . \tag{6.37}$$

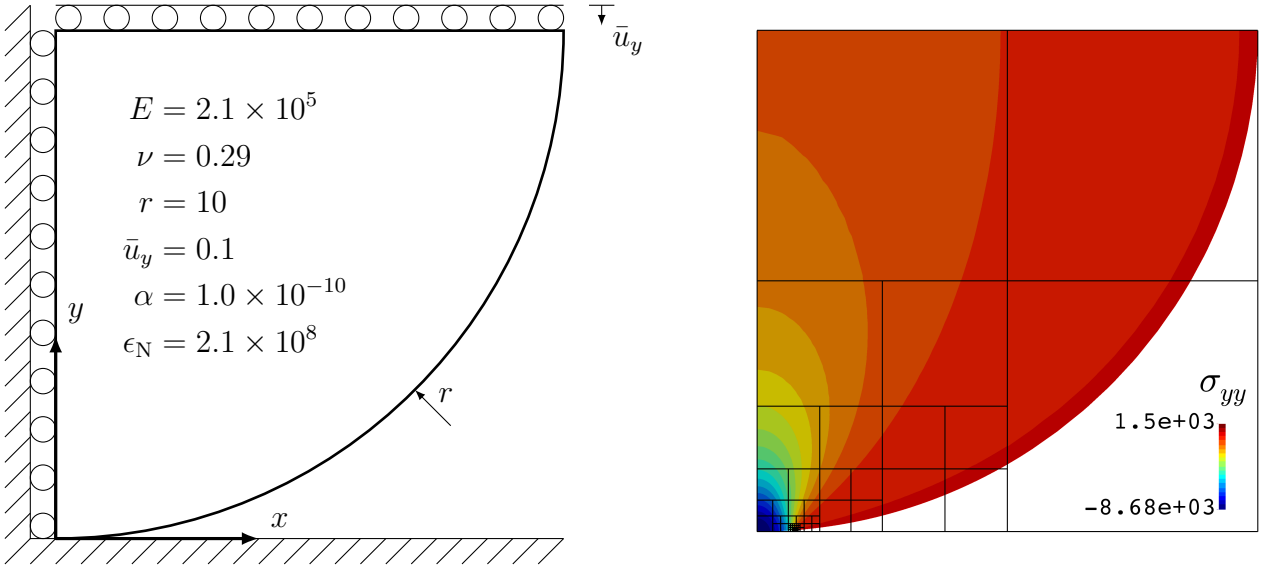
The distribution of the normal contact traction t_N then follows as

$$t_N(x) = t_{\max} \cdot \sqrt{1 - \frac{x^2}{b^2}} \quad \text{with} \quad t_{\max} = \frac{2 \cdot F}{\pi \cdot b} \ . \tag{6.38}$$

Note, that in the Hertzian theory the boundary of the contact body is described by a parabola. An improved theory has been presented by Shtaerman [116], which allows for a general geometric description of the contact boundary.

6.4.1.2 Numerical model

In the numerical model only a quarter of the cylinder is represented, by taking advantage of symmetries (Figure 6.6). The physical domain is modeled using a linear elastic, plane strain material model with thickness one. Symmetry conditions are applied in the vertical plane (y–z). The initial gap g is set to 0. In all following simulations following the cylinder is embedded in a Cartesian grid of size $r \times r$. The point load of the Hertzian model is represented by a uniform prescribed displacement on the horizontal center line of the cylinder, following [36]



(a) Problem setup **(b)** Overkill solution

Figure 6.6: Hertzian contact of an elastic cylinder against a rigid plane. The right figure depicts the vertical stress component σ_{yy} for an overkill solution using a FCM mesh with 2×2 elements of order $p = 10$ and 10 hierarchical refinements towards the analytical contact point.

and [15]. The point load value corresponding to the prescribed displacement can be recovered by integrating the stress component σ_{yy} along the horizontal center line of the cylinder

$$F = 2 \cdot \int_0^r \sigma_{yy}\left(x, y = r\right) \, \mathrm{d}a \ . \tag{6.39}$$

The value for the point load as well as the strain energy, for which no analytical solution is available, have been obtained from an overkill solution (Figure 6.6 and Table 6.1).

Table 6.1: Scalar quantities obtained for the overkill solution.

Strain energy	$\mathcal{U}_{\text{ref}}$	240.9410273412	Nmm
Point load	F	10384.1892800493	N
Contact width	b	0.7593740278	mm
Peak traction	$t_{\max}$	8705.5653378157	Nmm^{-2}

This overkill solution has been computed using an FCM mesh of order $p = 10$, refined ten times towards the theoretical contact boundary with the multi-level *hp*-method, leading to 24002 unknowns. Here, the contact boundary is exactly represented by an arc. All domain integrals are evaluated using quadratures obtained from exact partitioning [68]. The mesh and the vertical stress component σ_{yy} for the overkill solution are depicted in Figure 6.6b.

6.4.1.3 Deviation in strain energy

To assess to quality of our proposed contact formulation, we now look at the relative deviation in the strain energy η

$$\eta = \sqrt{\frac{|\mathcal{U}_{\text{ref}} - \mathcal{U}_{\text{num}}|}{\mathcal{U}_{\text{ref}}}} \cdot 100\,[\%] \quad . \tag{6.40}$$

Here, $\mathcal{U}_{\text{ref}}$ is the reference value obtained with the overkill solution (Table 6.1) and $\mathcal{U}_{\text{num}}$ is the value obtained using different discretizations. These include uniform mesh ($h-$) refinements for ansatz orders of $p = 1$ and $p = 2$, as well as refined meshes using the multi-level hp approach. All studies start from an initial mesh with 2×2 elements. The number of elements in the uniformly refined meshes is set to 2^i with $i \in \{1, \dots, 7\}$ in both x- and y-direction. The multi-level hp meshes are refined towards the analytical contact boundary k times ($k = 1, \dots, 6$). In all of the computations, the contact interface is recovered using the approach introduced in Section 5. The polynomial orders of the resulting boundary curves are set to the ansatz order of the corresponding base mesh. To rule out any domain integration errors, all volumetric integrals have been evaluated exactly following Kudela et al. [68]. The weak contact constraints were obtained using a full Gauss-Legendre quadrature of order $q = 50$. The results are depicted in Figure 6.7.

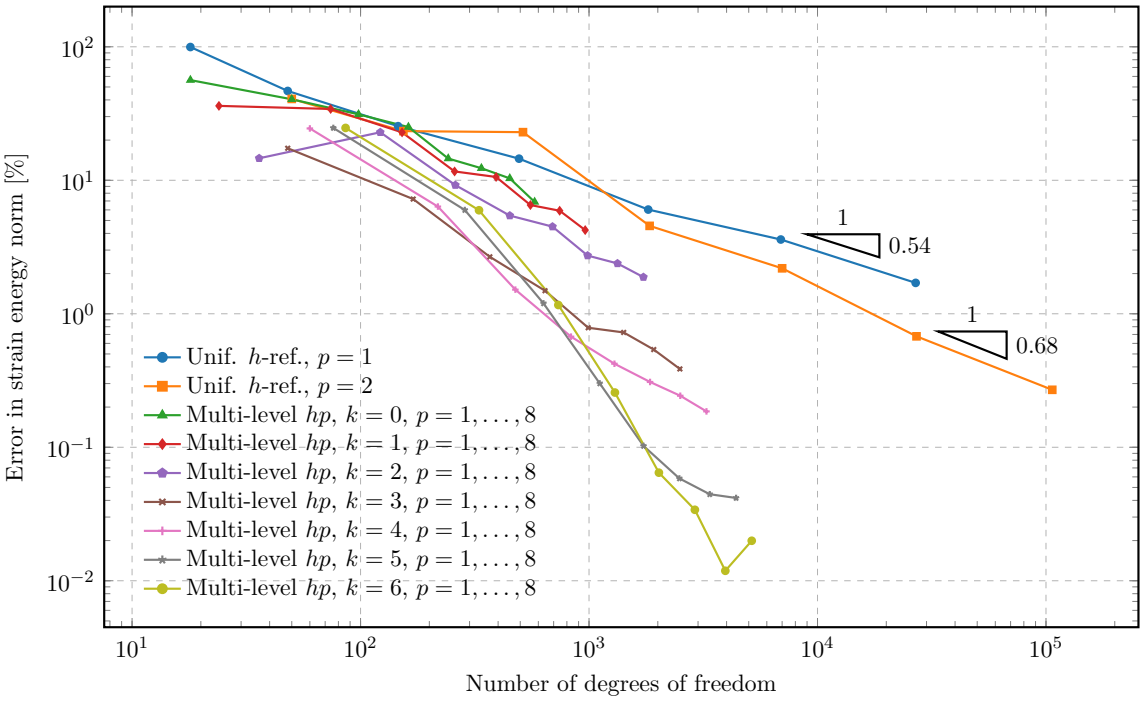


Figure 6.7: Convergence in strain energy error for uniform h-refinement for $p = 1$ and 2 and local refinement for ansatz orders $p = 1, \dots, 8$ and refinement depths $k = 0, \dots, 6$.

As expected, the strain energy error for the uniformly refined meshes converges algebraically. In contrast, the errors for the hierarchically refined meshes converge exponentially in the pre-asymptotic range with much higher rates.

6.4.1.4 Normal contact tractions

We first compare the normal contact tractions t_{N} computed with our finite element formulation with the solution (6.38) presented by Hertz [51]. The normal tractions are computed as

$$t_{\mathrm{N}} = (\boldsymbol{\sigma} \cdot \boldsymbol{n}) \cdot \boldsymbol{n} \ , \tag{6.41}$$

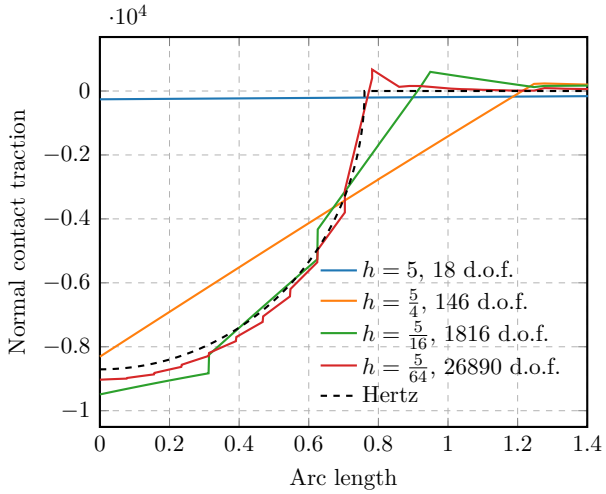
where $\boldsymbol{\sigma}$ and $\boldsymbol{n}$ denote the Cauchy stress tensor and the surface normal, respectively. Figure 6.8 depicts the contact tractions normalized with the maximum contact tractions $t_{\max}$ (see Table 6.1) computed for different discretizations (Section 6.4.1.3).
The results show that uniform h-refinements as well as localized multi-level hp refinements help to improve the contact tractions. However, the tractions obtained for the finest uniform refinement with $p = 1$ (Figure 6.8a) are slightly overestimated. The same holds for the contact stresses computed with the finest multi-level hp-mesh of order $p = 2$ and $k = 6$. The pure elevation of the polynomial order (Figure 6.8c) is not able to satisfactory approximate the analytical solution, since the polynomial solutions cannot capture the contact boundary.

6.4.1.5 Influence of geometric boundary resolution on contact tractions

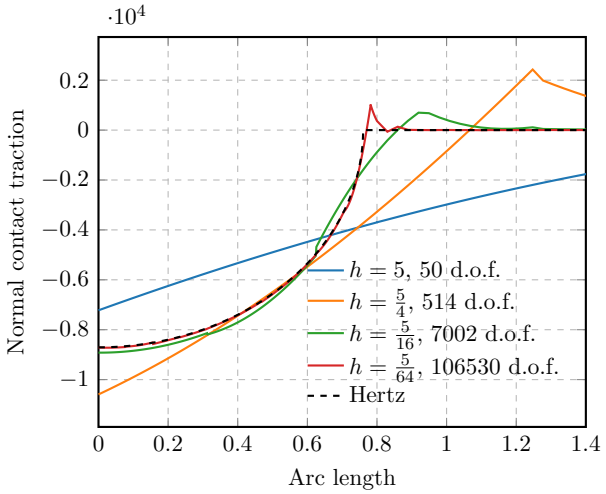
The geometric description of the contact surface plays a significant role within a contact simulation. This is especially true in the current formulation, where the contact contributions are obtained via numerical integration along the contact segments. Therefore, we will now look at the influence of surface description on the resulting contact stresses. To minimize all error contributions from the physical domain, we resort the numerical model used for the overkill solution (Figure 6.6b). The contact surfaces are obtained using the approaches introduced in Chapter 5 using linear as well as high-order curve segments. In the case of linear curve segments, the contact integrals are evaluated using a 2^{nd} order Gauss-Lobatto (trapezoidal) rule, and the number of segments is gradually increased. In the second case, one segment per sub-element will be recovered. The corresponding contact integrals are obtained using the standard Gauss-Legendre scheme with $p + 3$ integration points. The tractions resulting from the linear description (Figure 6.9a) show highly oscillatory behavior, which decreases with an increased number of segments. This suggests that a considerable number of integration points will be necessary to obtain an oscillation-free contact tractions for the current setup.
The high-order representations (Figure 6.9b-6.9d) also show oscillatory behavior but already align with the analytical solution for a boundary order of $p_{\mathrm{seg}} = 5$ (Figure 6.9d). While some oscillations remain, they are confined within the finest refinement level. This is the basic mechanism of hp-methods [121].

6.4.1.6 Stress solution

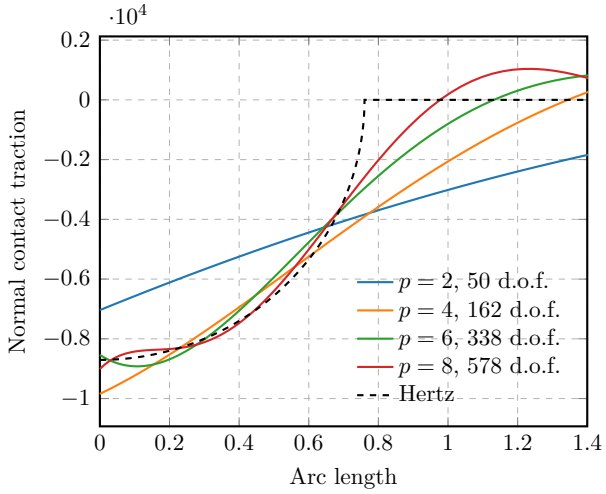
For completeness, we also present the global stress solution for some of the discretizations of Section 6.4.1.3 in Figures 6.10 and 6.11.
The results obtained with the uniformly refined meshes (Figures 6.10c and Figures 6.10f) can represent characteristic stress distribution for the Hertzian solution including the traction cone (e.g. [36]). A pure elevation of the polynomial order (Figures 6.10a, 6.10d, 6.11a) also improves the overall solution, although the contact singularity introduces some significant oscillations. As shown in [140], the hp-d refinement (Figure 6.11b) does not efficiently improve the global



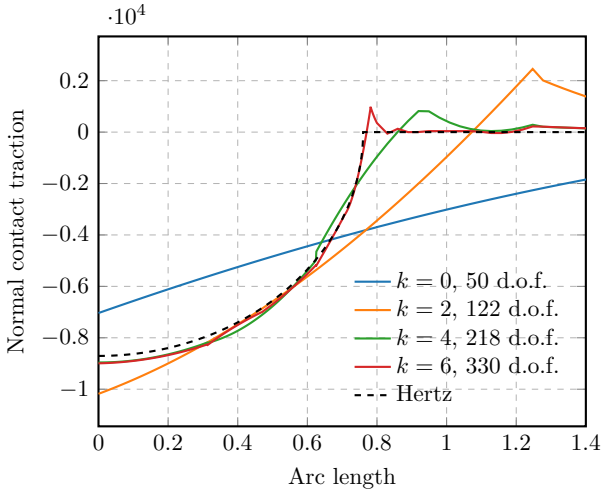
(a) Uniform h-refinement ($p = 1$)



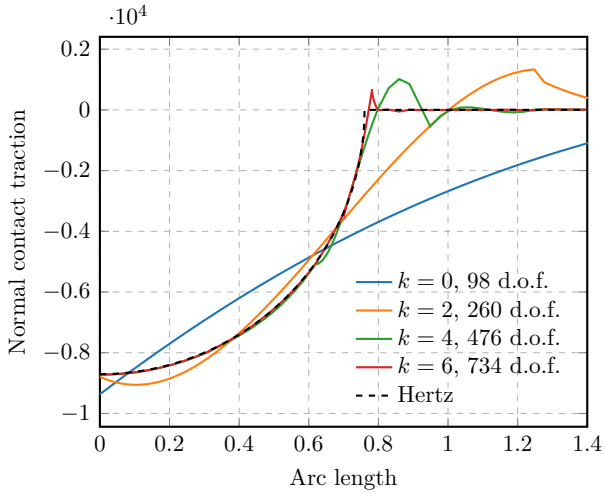
(b) Uniform h-refinement ($p = 2$)



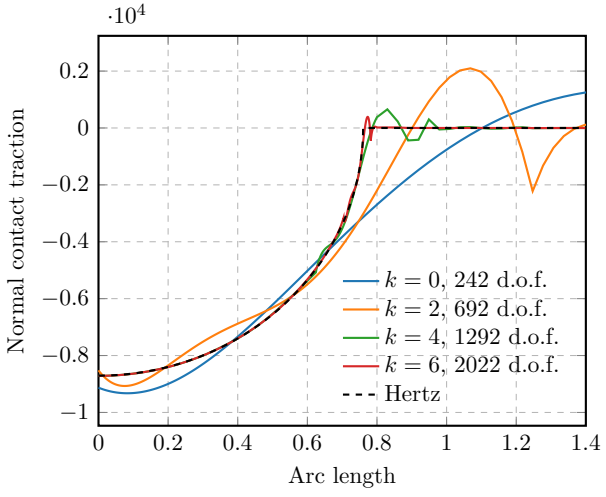
(c) Uniform p-refinement ($h = 5$)



(d) multi-level hp-refinement ($p = 2$).

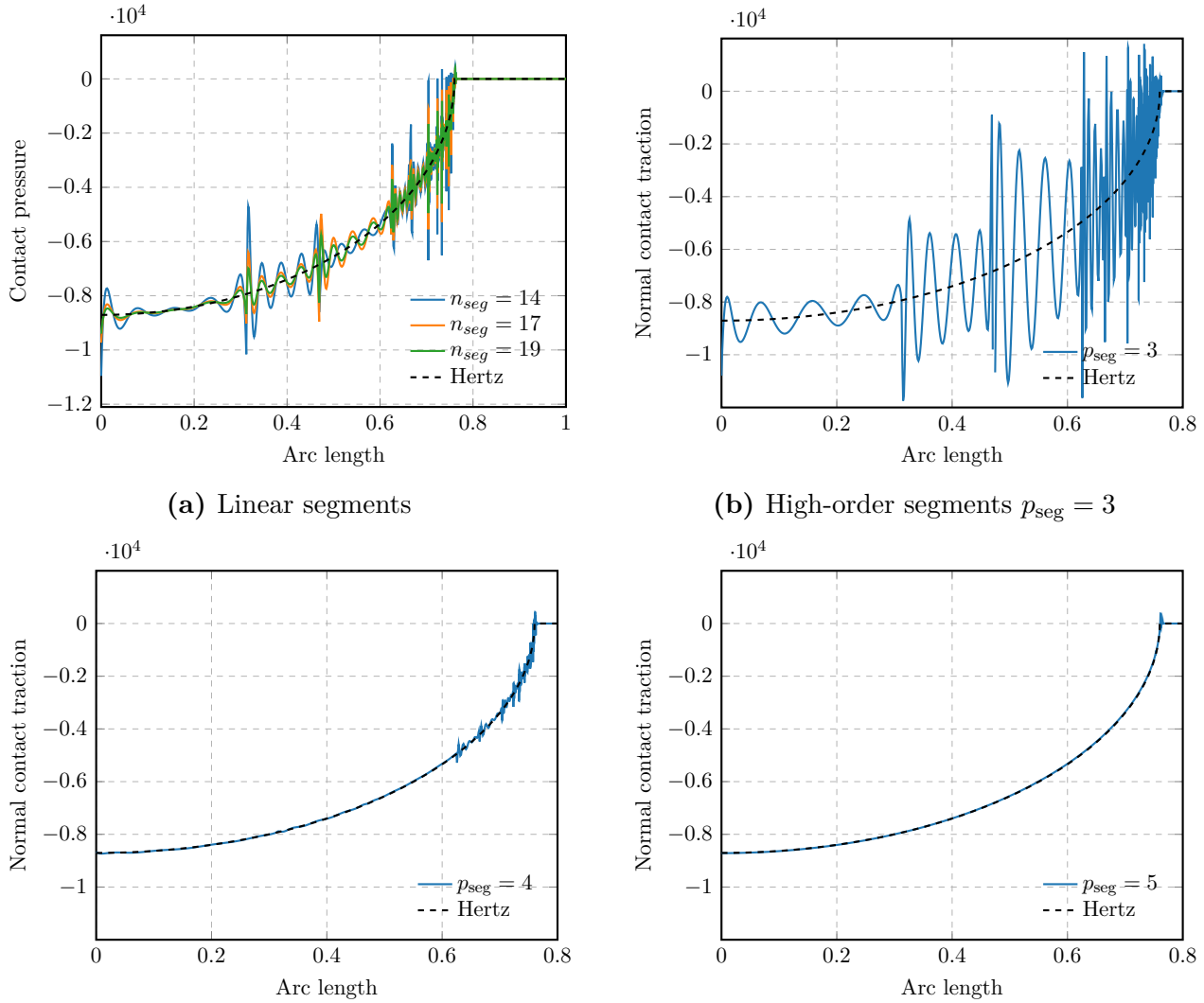


(e) multi-level hp-refinement ($p = 3$).



(f) multi-level hp-refinement ($p = 5$).

Figure 6.8: Contact stresses obtained with different contact discretizations. All studies start from an initial discretization of 2×2 finite cells (h: characteristic element size, p: polynomial order, k: refinement depth).



(a) Linear segments

(b) High-order segments $p_{\text{seg}} = 3$

(c) High-order segments $p_{\text{seg}} = 4$

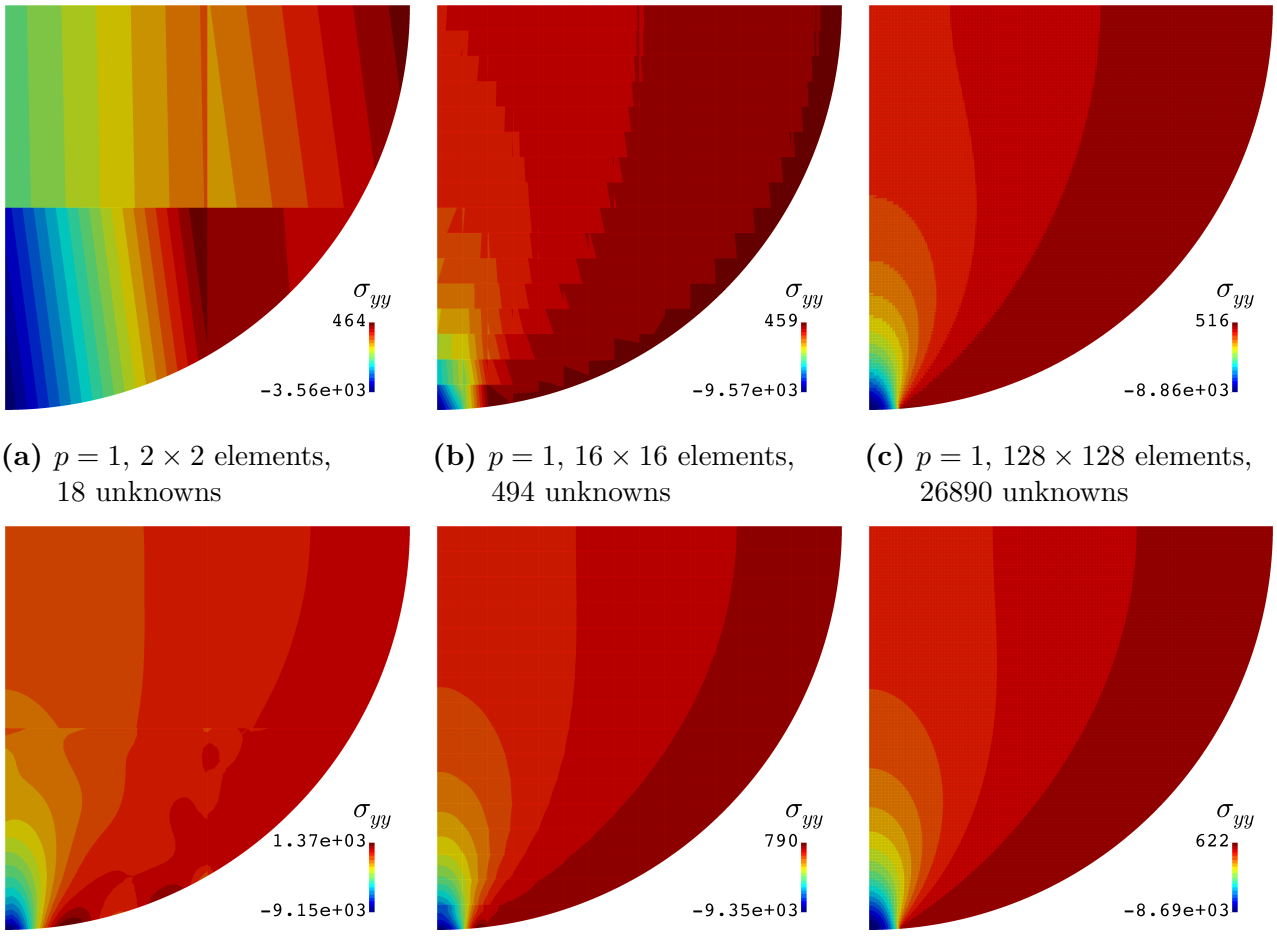
(d) High-order segments $p_{\text{seg}} = 5$

Figure 6.9: Normal contact tractions for different representations of the contact interface.

stress solution (compare Figure 6.11a). The reason is, that the *hp-d* approach only adds linear overlay modes, without the breaking the supports of the high-order shape functions. In contrast the multi-level *hp* approach (Figure 6.11c) is able to dramatically improve the solution locally, since it is able to localize high-order modes close to the singularity.

6.4.2 Ironing 2D

The following example considers bilateral contact in the large deformation regime. A curved elastic body is first pressed into another elastic block and then moved horizontally by prescribed displacements (Figure 6.12). A similar example has been presented by Temizer and Hesch [122] using a NURBS discretization with a fixed uniform hierarchical refinement.
Both bodies are modeled using the Neo-Hooke material given in Section 2.3.2.3. The Young's moduli for the upper and lower body are set to $E_{\text{U}} = 10^3$ and $E_{\text{L}} = 1.0$, respectively, while



(a) $p = 1$, 2×2 elements, 18 unknowns

(b) $p = 1$, 16×16 elements, 494 unknowns

(c) $p = 1$, 128×128 elements, 26890 unknowns

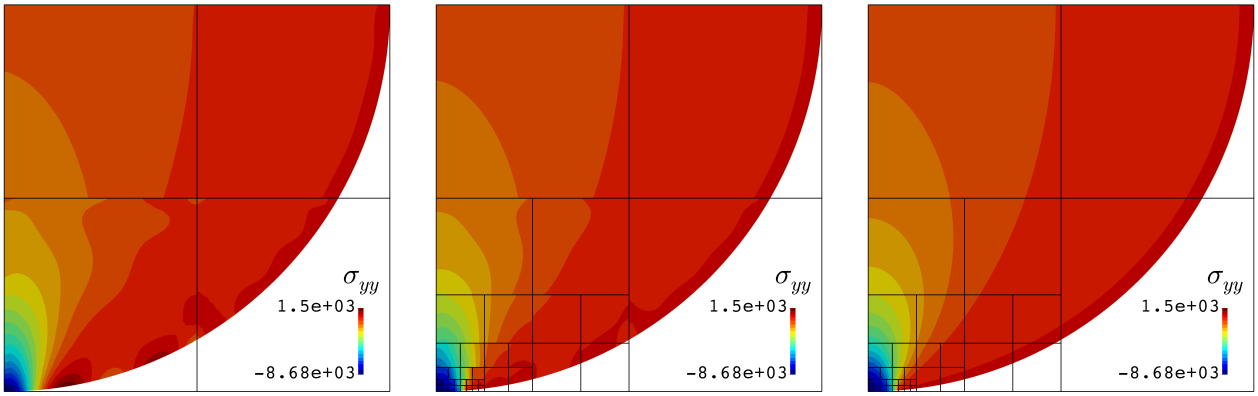
(d) $p = 2$, 2×2 elements, 50 unknowns

(e) $p = 2$, 16×16 elements, 1842 unknowns

(f) $p = 2$, 128×128 elements, 106530 unknowns

Figure 6.10: Global stress component σ_{yy} depicted for uniformly refined discretizations. Mesh lines have been ommited for better visualization.

the Poisson ratio is set uniformly to $\nu = 0.3$. Each body is embedded in its own separate grid (Figure 6.13), as described in Section 4.4. The polynomial order of all elements is set to $p = 4$ leading to a total of $1,300$ unknowns. All volume integrals are evaluated using a composed integration based on quadtree partitioning with depth three and depth one for the lower and upper body, respectively (Chapter 4). The fictitious domain parameter is set to $\alpha = 10^{-10}$. To increase the accuracy of the solution, we also employ local mesh refinement based on the multi-level *hp*-formulation [140] (Figure 6.14). Therefore, a local refinement of four levels is applied to both meshes, leading to at most 10,832 unknowns. The refinement zone moves along with the ends of the contact zone using a simple refinement indicator (Figure 6.14). This indicator is based on the gap function g (Figure 6.15a). Whenever the gap resulting from a closest point projection is within the range $[g_{\min}, g_{\max}]$, both the slave and the master mesh are refined at the material points $(\boldsymbol{X}^*_{\mathrm{S}}, \boldsymbol{X}^*_{\mathrm{M}})$ associated to the projection $(\boldsymbol{x}^*_{\mathrm{S}}, \boldsymbol{x}^*_{\mathrm{M}})$. For the current example, the gap range is set to $[10^{-8}, 10^{-3}]$. The penalty parameter for the contact constraint is set to $\epsilon_{\mathrm{N}} = 10^3$. The contact boundary is constructed by recovering one segment



(a) $p = 5$, 2×2 elements, 242 unknowns

(b) $p = 5$, 2×2 elements, *hp-d*, 308 unknowns

(c) $p = 5$, 2×2 elements, m.-l. *hp*, 1732 unknowns

Figure 6.11: Global stress component σ_{yy} depicted for different discretizations.

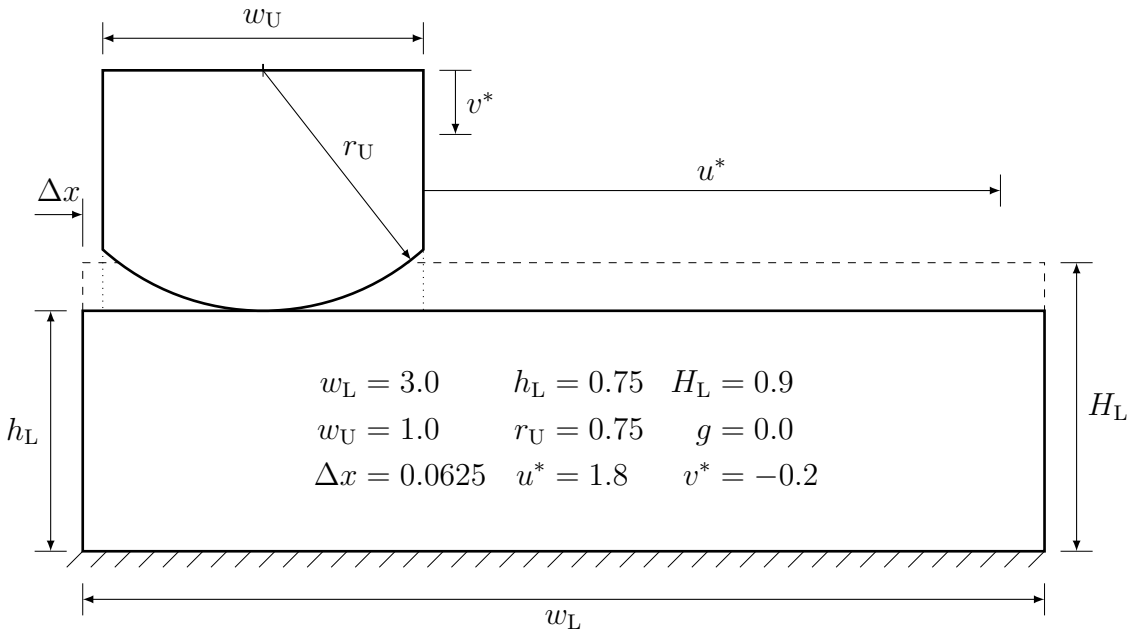
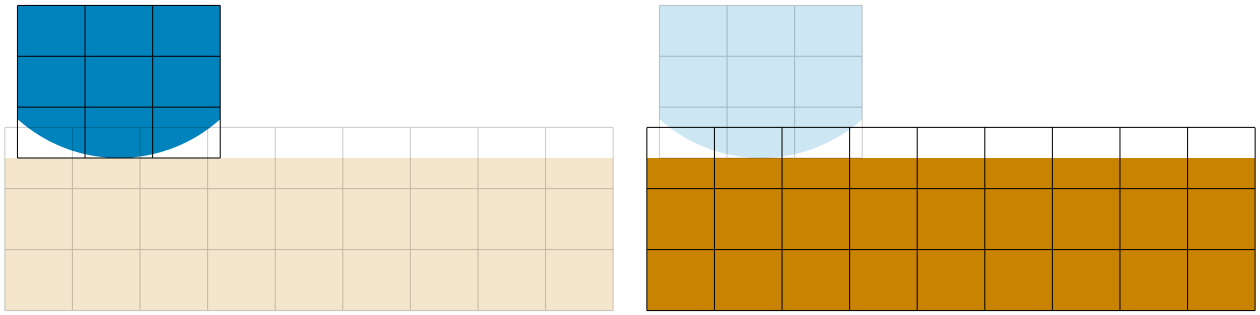


Figure 6.12: Ironing problem setup. The dashed and dotted lines show the extent considered for meshing.

of order $p_{\mathrm{seg}} = 4$ for every element using the approach presented in Chapter 5. To adequately resolve the discontinuous contact integral, the index space of every contact boundary segment is uniformly partitioned ten times and a Gauss-Legendre scheme of order $q = 10$ is employed. The upper body is first pressed downwards in 10 load steps by a prescribed displacement $v^* = -0.2$. Following the vertical loading, the upper body is dragged horizontally over a total distance of $u^* = 1.8$, which is linearly increased over 100 load steps. The convergence tolerance of the Newton solver is set to 10^{-6}.

In case of the locally refined discretization, every load step is first solved using the refinement



(a) Upper body (dark blue) with its finite cell mesh (black lines). The lower body (light orange) and its separate mesh (gray lines) are only shown for clarity.

(b) Lower body (dark orange) with its finite cell mesh (black lines). The upper body (light blue) and its separate mesh (gray lines) are only shown for clarity.

Figure 6.13: Discretized bodies.

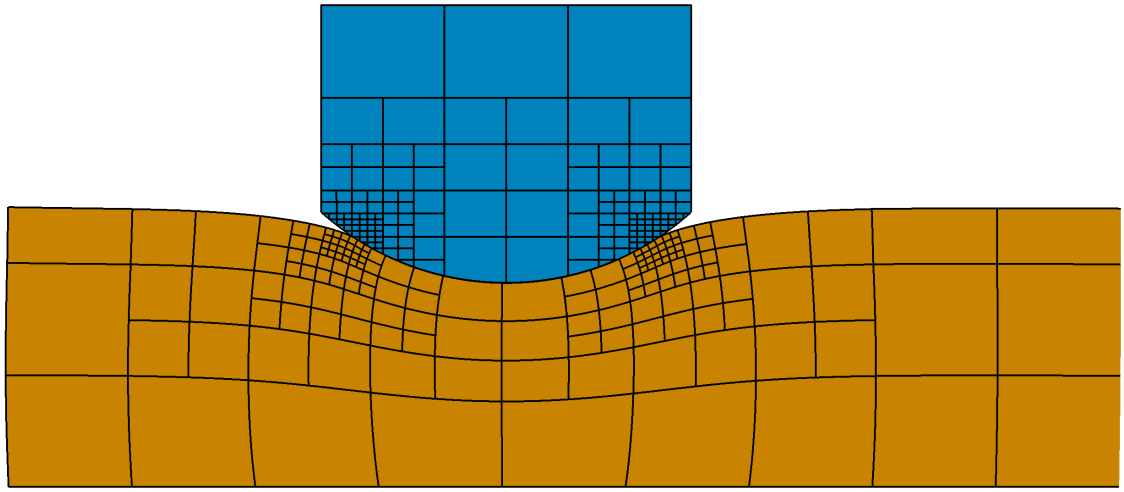
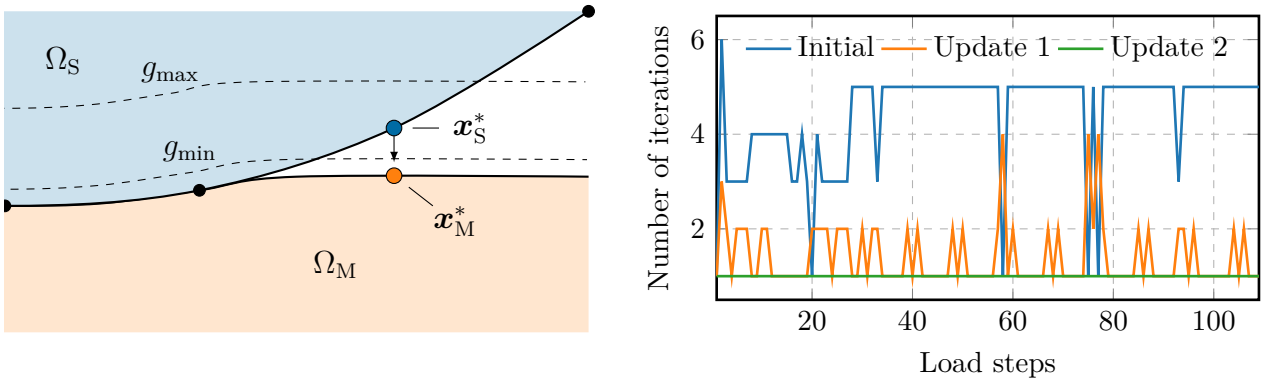


Figure 6.14: Local refinement following the contact boundary. Both bodies are shown in a deformed configuration, with their meshes imprinted.

from a previous load step. Afterward, the refinement is updated using the gap-based indicator (Figure 6.15a) and the load step is solved again. This correction step is repeated two times. Figure 6.15b shows the number of iterations for this refinement corrections scheme. In all load steps, the final correction step only uses on iterations. This implies that one correction step is sufficient for the current setup.

6.4.2.1 Normal contact traction

The normal contact tractions t_N obtained with the unrefined discretization (Figure 6.13) are shown in Figure 6.16 at different time steps. The tractions show pronounced irregularities inside the contact zone and oscillations at the contact boundaries. The local refinement improves the contact traction in the contact zone (Figure 6.17) and eradicates artifacts such as oscillations at the boundary of the contact zone.



(a) Gap based refinement indicator **(b)** Number of Newton iterations

Figure 6.15: Refinement indicator and resulting number of iterations per load step for a convergence tolerance of 10^{-6}.

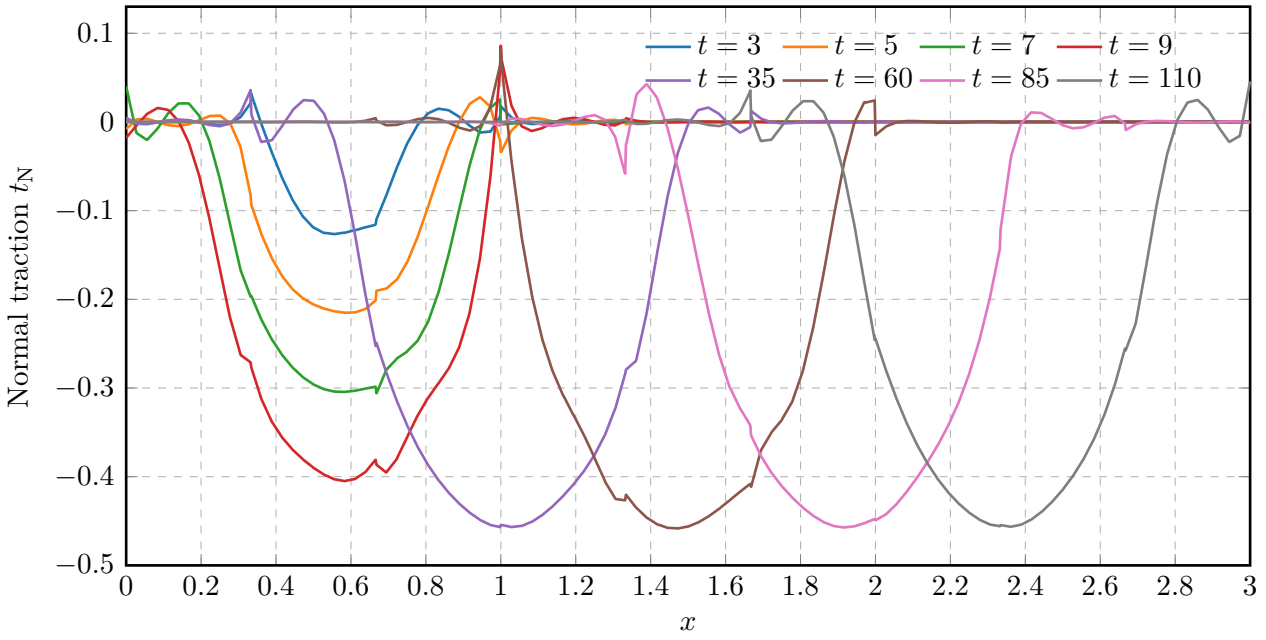


Figure 6.16: Normal tractions t_N for unrefined discretization (Figure 6.13).

6.4.2.2 Evolution of normal and tangential load

Figures 6.18a and 6.18b shows the normal and tangential reaction forces at the top edge of the upper body. The computed normal reaction forces are almost indistinguishable for all considered discretizations. Yet, the tangential reaction forces computed for the unrefined and the uniformly refined discretization strongly oscillate during the load history. However, the local refinement results in a much smoother progression of the tangential load.

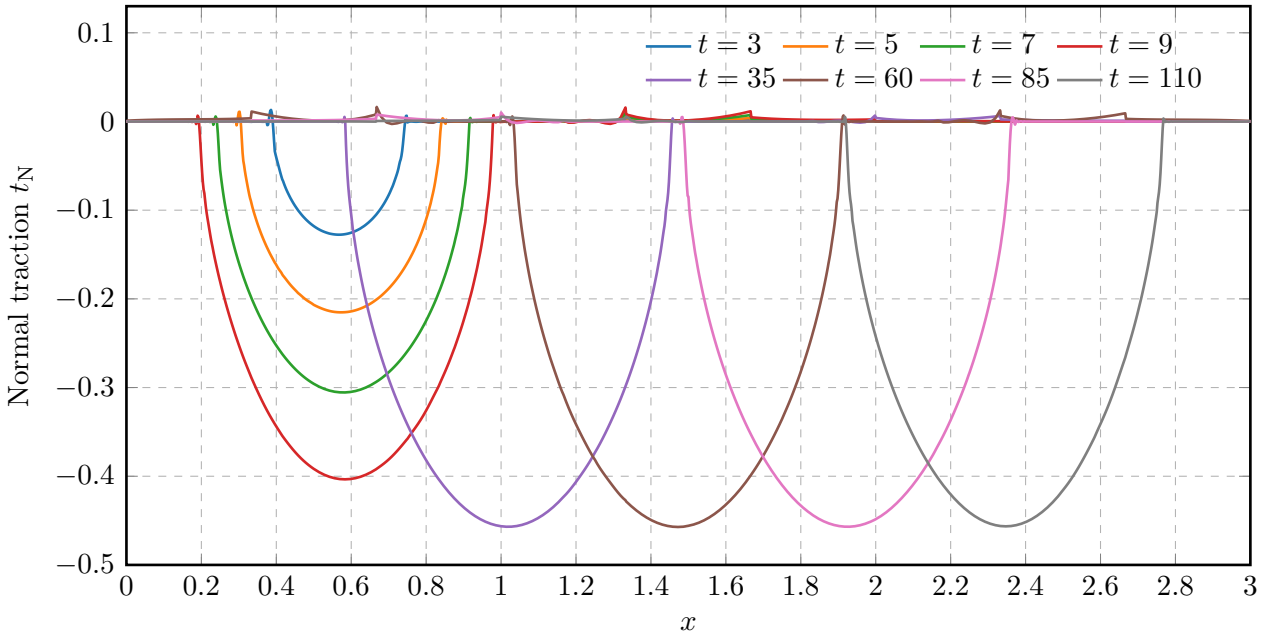
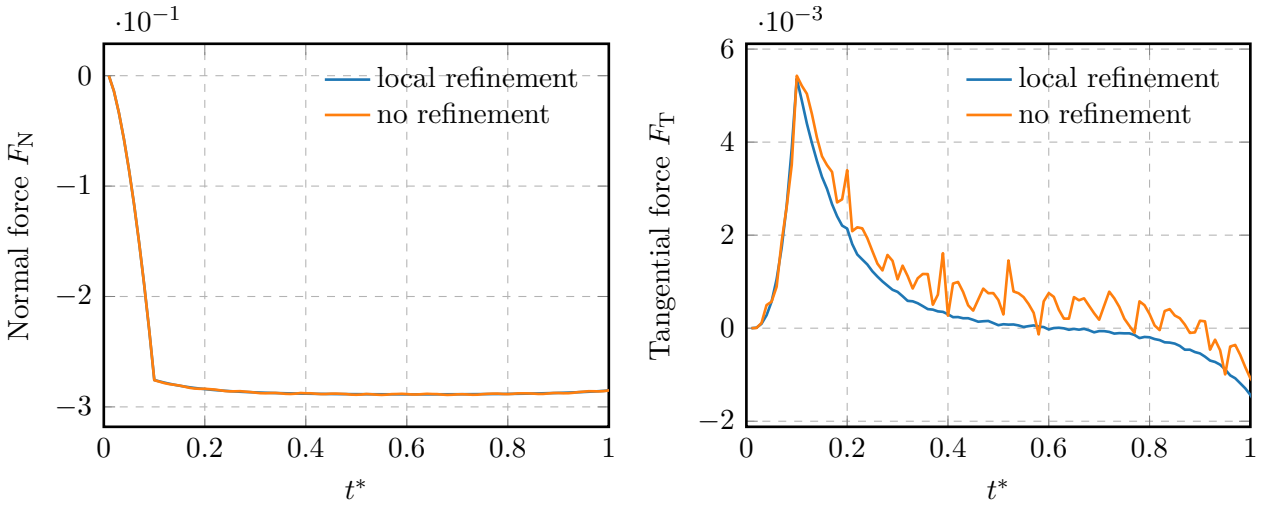


Figure 6.17: Normal tractions t_N for locally refined discretization (Figure 6.14).



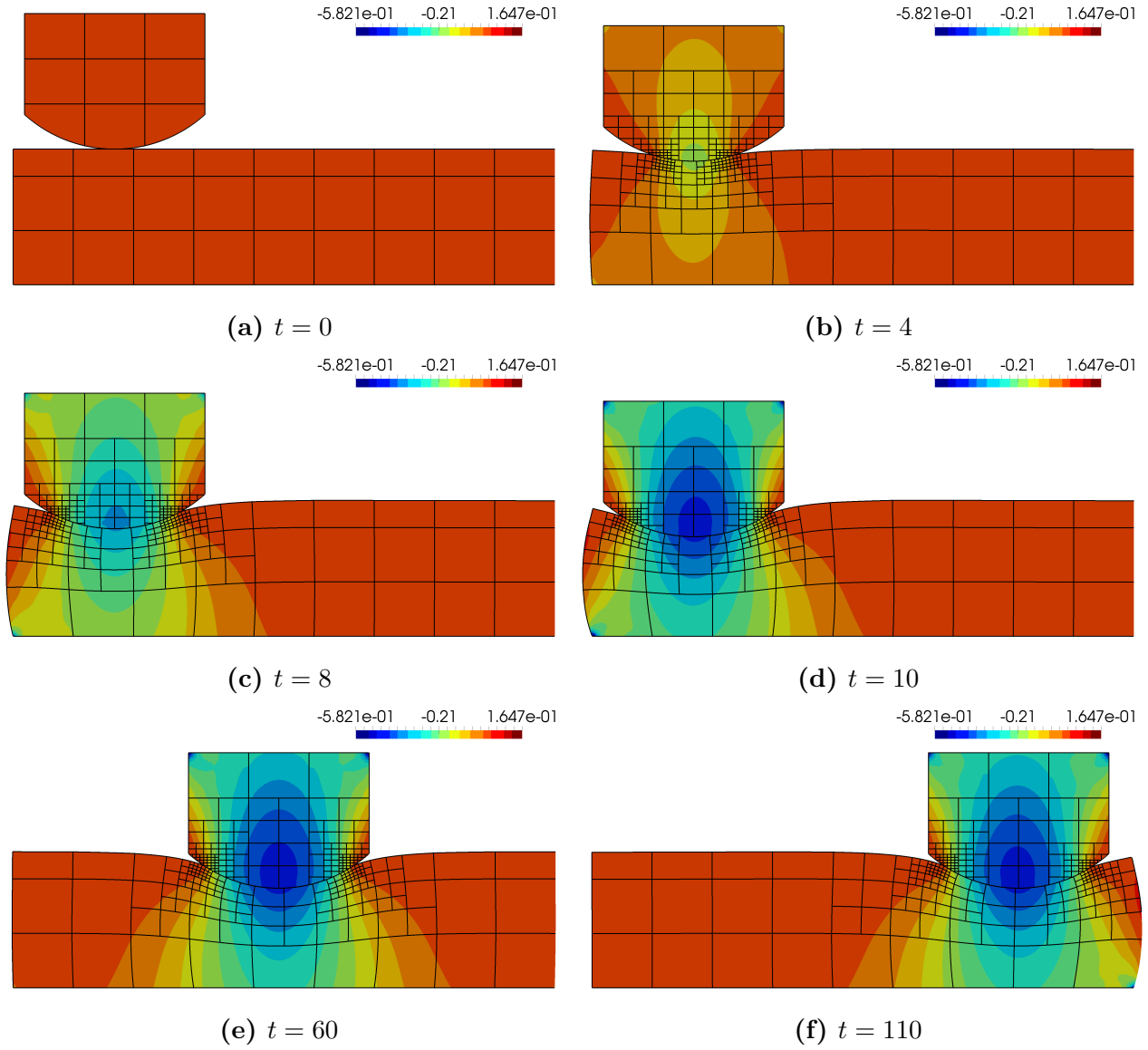
(a) Normal reaction forces: Both curves are virtually identical

(b) Tangential reaction forces

Figure 6.18: Progression of normal and tangential reactions forces for both discretizations over the whole (normalized) load history.

6.4.2.3 Global stress solution

Figure 6.19 shows that the employed coarse high-order base mesh can represent the global solution characteristic correctly, while it is not able to accurately resolve the stress concentration at the boundary of the contact zone. However, the results demonstrate that the dynamic *hp*-refinement obtained by the combination of the refinement indicator and the repeated load step corrections can effectively localize the refinement at the end of the contact zone. In this way, the progression of the stress solution is accurately resolved during the loading history.



(a) $t = 0$ **(b)** $t = 4$

(c) $t = 8$ **(d)** $t = 10$

(e) $t = 60$ **(f)** $t = 110$

Figure 6.19: Progression of stress component σ_{yy}. The upper and lower body are discretized with 3×3 and 9×3 elements, respectively. Based on a simple indicator (Figure 6.15a), both meshes are refined four times towards the ends of the contact boundary.

6.4.3 Three-dimensional example

This example considers a three-dimensional setup between a chain lock and a chain link. For the numerical model, the symmetries in the x-z-plane are exploited such that only one-half of the whole model is computed, as shown in Figure 6.20.

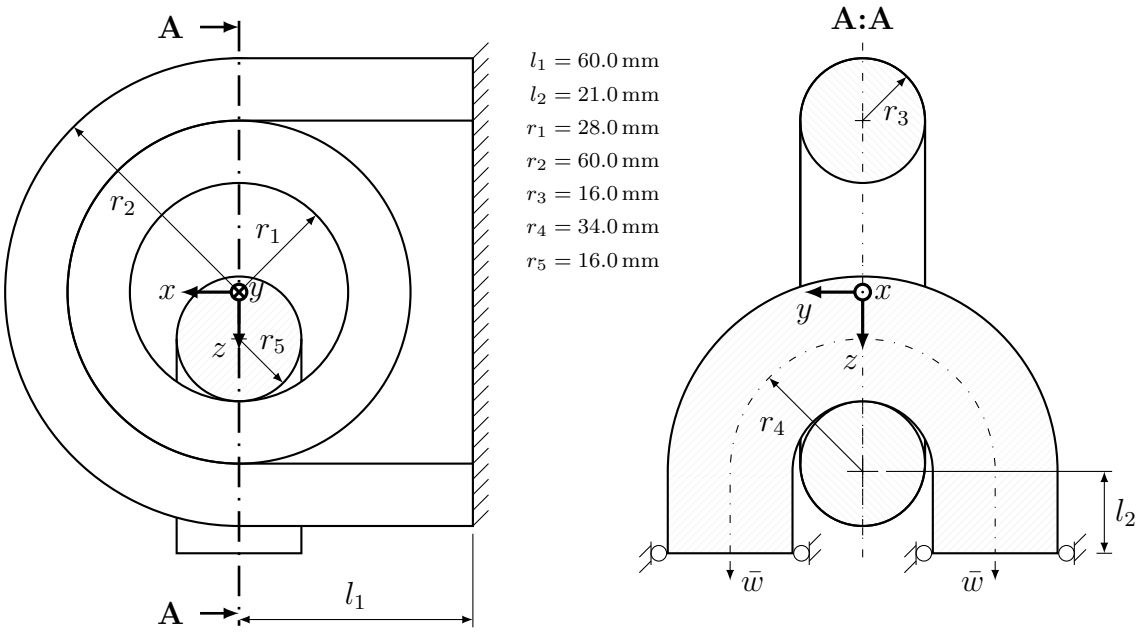
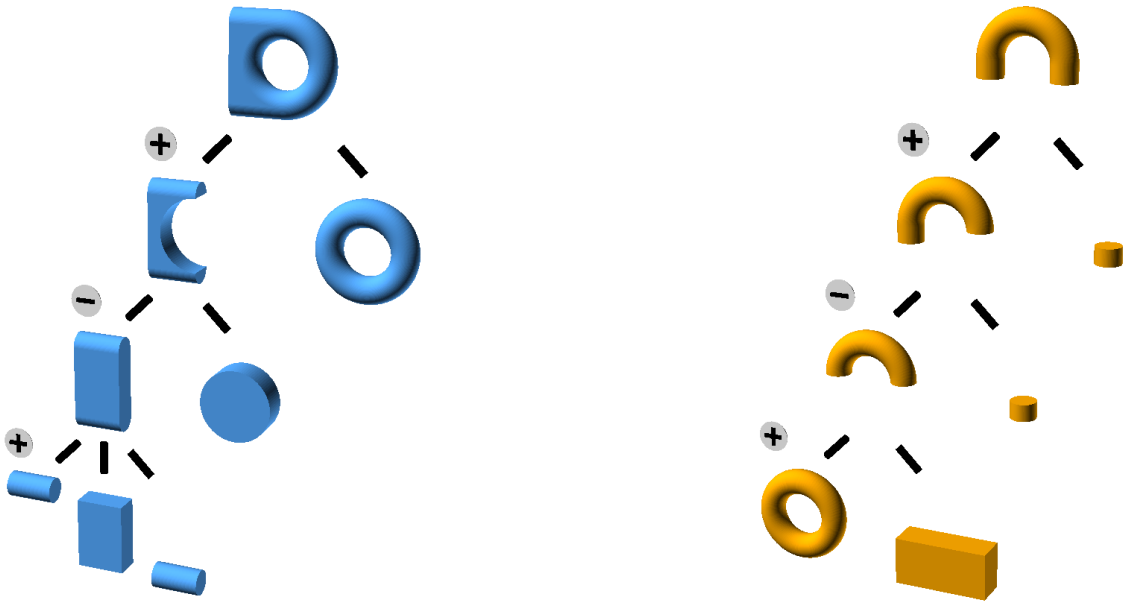


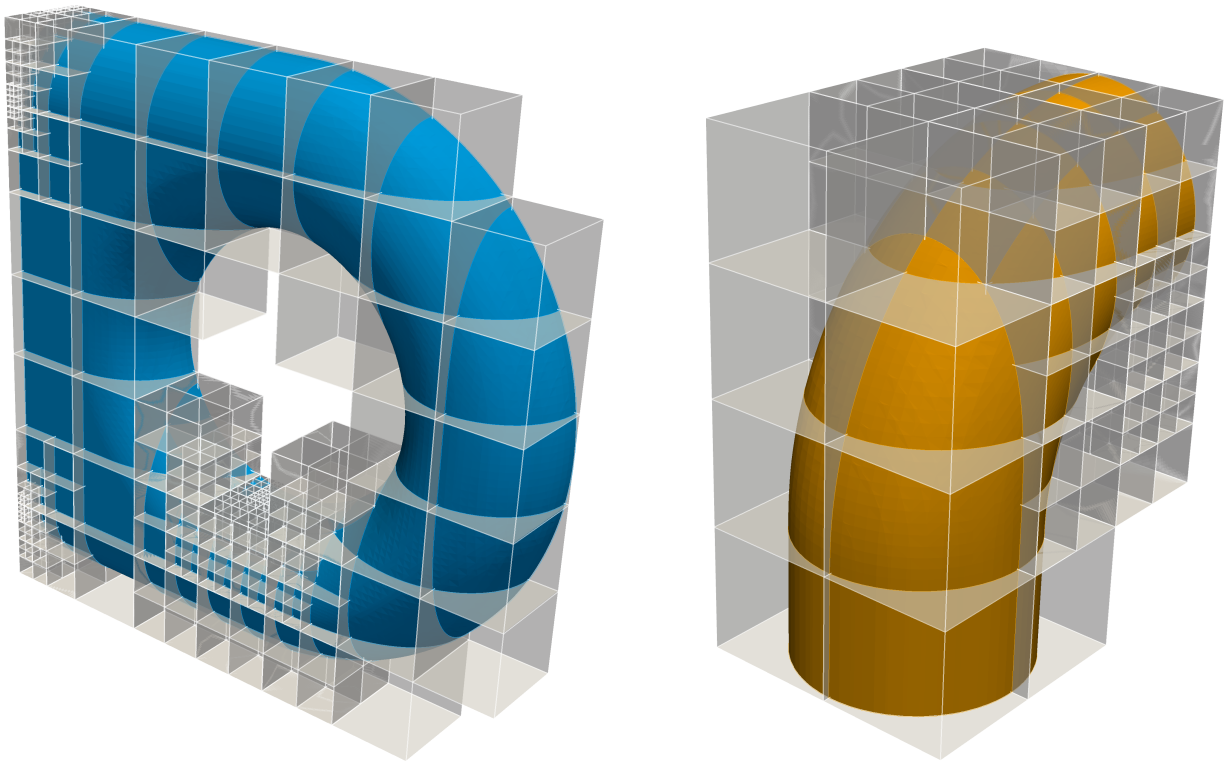
Figure 6.20: Contact between chain lock and chain link. The lock is fully supported on its back side. The link is loaded by a prescribed displacement of $\bar{w} = -0.02\,\text{mm}$ in the vertical direction. By exploiting symmetries, only one-half of the lock and link is modeled. ($E = 2.1 \times 10^5$ MPa, $\nu = 0.3$, $\epsilon_\text{N} = 2.1 \times 10^8$, $\alpha = 10^{-10}$).

The geometric bodies are modeled using only CSG primitives and Boolean operations (see Figures 6.21a and 6.21b). The resulting meshes for each body are shown in Figure 6.22b and 6.22a, respectively. The chain lock is fully clamped at its rear side, and the chain link is loaded by a vertical displacement $\bar{w}$ in its horizontal symmetry plane. To account for singularities introduced by the Dirichlet boundary condition of the chain lock and the contact zone, the mesh is refined four times at the respective positions (see Figure 6.24).
The stress solution of the refined meshes, as shown in Figure 6.22a, is compared to a stress solution obtained for the same setup without refinement. The contact boundary for both configurations is represented by Lagrange triangles of order $p_\text{seg} = 5$ as described in Chapter 5. Recovery grids of $1 \times 1 \times 1$ cubes and $20 \times 20 \times 20$ cubes have been used to obtain the contact boundaries in the refined and unrefined FCM grids, respectively. The contact contributions are integrated numerically using a standard Gauss-Legendre scheme of order $p + 3 = 8$. The integration points are distributed in a tensor product fashion and then mapped into the triangles. The volumetric integrals of both configurations are computed using octree-based composed integration with depth 4. Figure 6.25 shows the vertical stress component σ_{zz} for both configurations.
The unrefined configuration (Figure 6.25a) exhibits high oscillations around the contact zone and pronounced jumps along the element edges. In comparison, the refined configuration



(a) CSG tree of the chain lock **(b)** CSG tree of the chain link

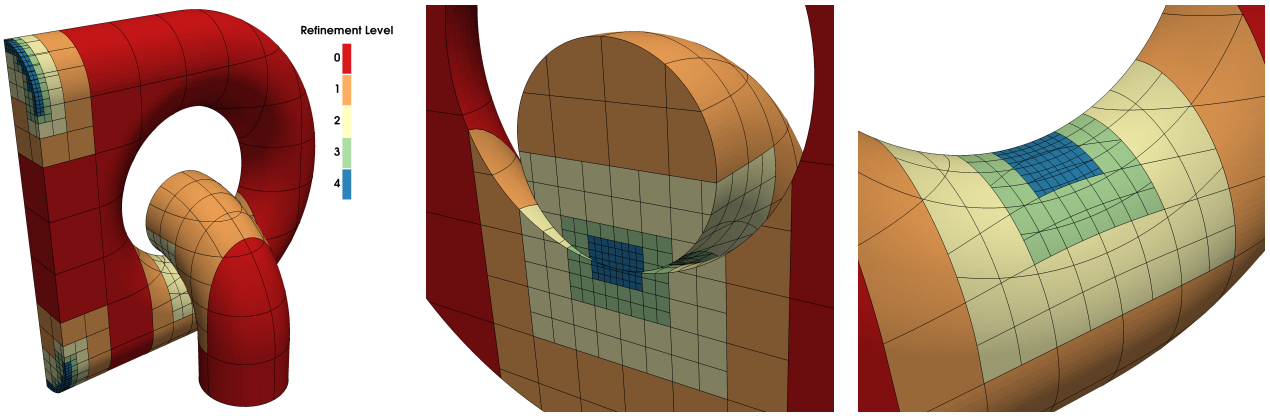
Figure 6.21: CSG construction trees.



(a) Chain lock **(b)** Chain link

Figure 6.22: Bodies embedded in their FCM grids.

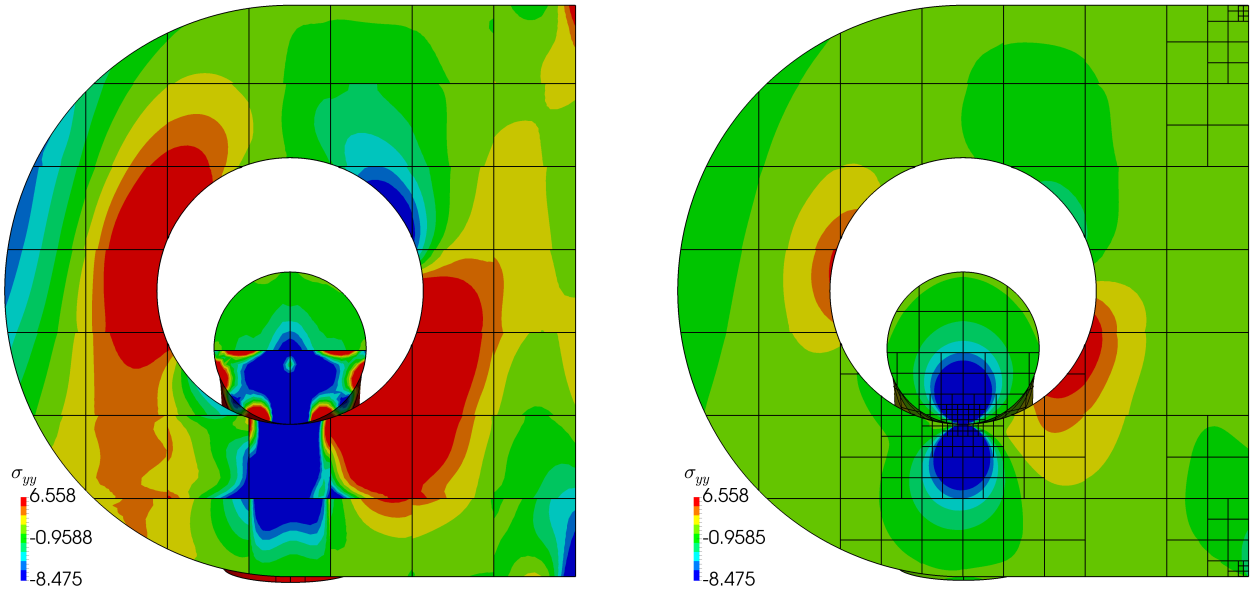
Figure 6.23: Both bodies and their FCM grids shown together.



(a) Whole model **(b)** Contact zone (both bodies) **(c)** Contact zone (chain lock)

Figure 6.24: FCM grids projected on their respective bodies. The grids are refined four times at the contact zone and at the curved parts of the clamped interface of the chain lock. The black lines represent the intersections of the bodies' surfaces with the element boundaries.

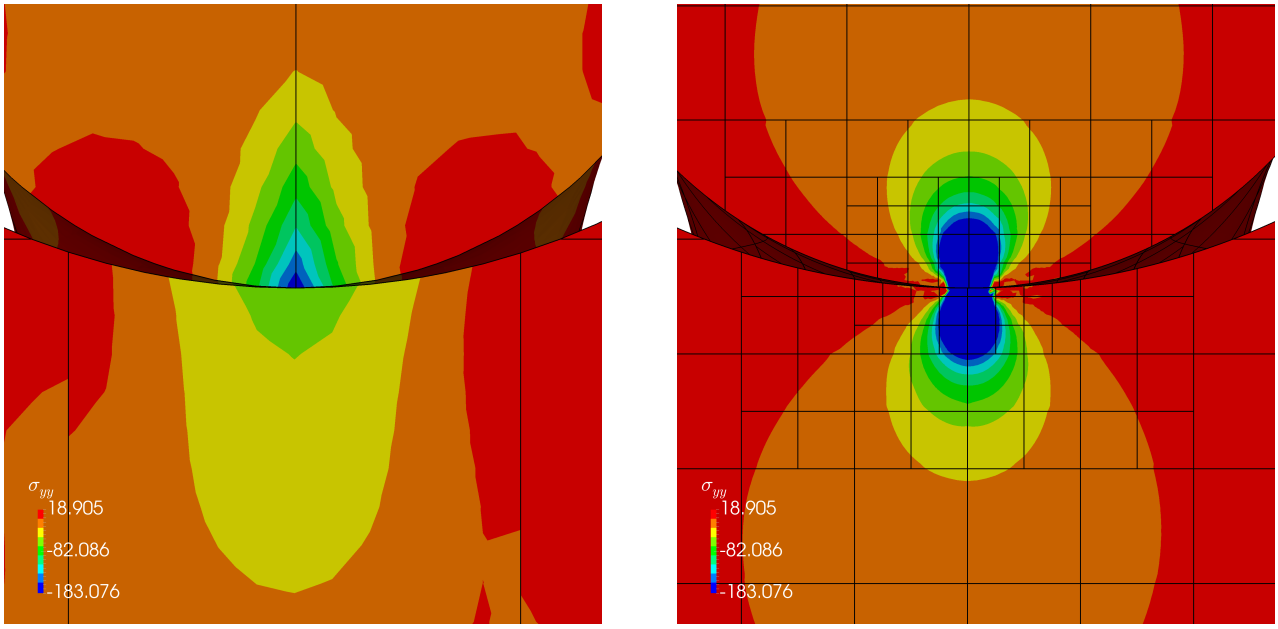
(Figure 6.25b) does not contain any visible jumps and localizes the compressive reaction in the contact zone. A more detailed view of the contact zone is shown in Figure 6.26.



(a) Unrefined mesh (7,962 unknowns)

(b) Multi-level *hp*-refined mesh (101,811 unknowns)

Figure 6.25: Stress component σ_{zz} in the central symmetry plane.



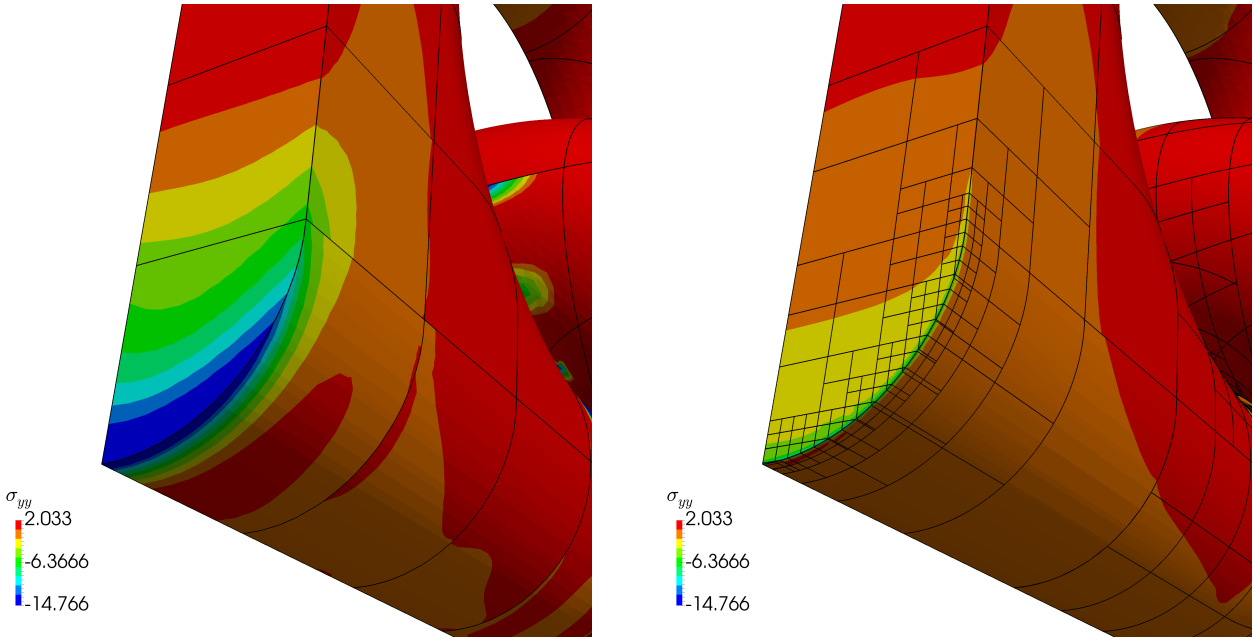
(a) Unrefined mesh

(b) Multi-level *hp*-refined mesh

Figure 6.26: Close-up of the contact zone showing the stress component σ_{zz}.

While some small oscillations remain around the end of the contact zone, they are restricted to the finest element level. Note that the refinement can also capture the singularities introduced

by the Dirichlet boundary conditions, as shown in Figure 6.27.



(a) Unrefined mesh **(b)** Multi-level *hp*-refined mesh

Figure 6.27: Close−up of the fully clamped lower part of the chain lock.

Chapter 7

Constraint enforcement using a contact material*

In the classical treatment of mechanical contact problems, constraints are formulated on domain boundaries [133, 71]. To this end, the discrete formulations of these constraints are assigned to potential contact interfaces *a priori* using special contact elements (Chapter 3). The approach introduced in this chapter follows a different idea. Physical bodies are embedded into a contact material, which replaces the usual contact constraints at the interface level. A similar model has been introduced lately by Wriggers et al. [134], where a different material model for the contact material was used. Furthermore, the current formulation is extended to higher order finite elements and combined with the fictitious domain concept of the FCM. The contact material used in our approach resembles the fictitious material used by the FCM (Chapter 4) until contact surfaces come into close proximity. It is only in this limit case that the contact material undergoes a substantial stiffening and generates the internal forces necessary to enforce the normal contact constraints. Note, however, that all possible contact interfaces (and only these!) are assumed to be discretized conformingly by element boundaries. This differs from the general FCM concept, where all domain boundaries may pass through the interior of elements. A non-conforming transition from physical to contact material within one element would cause significant oscillation in the approximate solution and a reduction of accuracy.

7.1 Governing equations

In the current approach, an additional (contact) domain Ω_{cont} is added to the formulation of the elastic bodies (Figure 7.1). The deformations $\boldsymbol{u}_{\text{phys}}$ and $\boldsymbol{u}_{\text{cont}}$ of the physical and the contact domain, respectively, are governed by the following partial differential equations:

$$\nabla \cdot \boldsymbol{\sigma}_{\text{phys}} + \boldsymbol{f} = \boldsymbol{0} \qquad \text{in } \Omega_{\text{phys}} \tag{7.1}$$

$$\boldsymbol{u}_{\text{phys}} = \bar{\boldsymbol{u}} \qquad \text{on } \Gamma_{\text{D}} \tag{7.2}$$

$$\boldsymbol{\sigma}_{\text{phys}} \cdot \boldsymbol{n} = \boldsymbol{t} \qquad \text{on } \Gamma_{\text{N,phys}} \tag{7.3}$$

$$\nabla \cdot \boldsymbol{\sigma}_{\text{cont}} = \boldsymbol{0} \qquad \text{in } \Omega_{\text{cont}} \tag{7.4}$$

* The major part of this chapter has been published in T. Bog, N. Zander, S. Kollmannsberger, and E. Rank, "Normal contact with high order finite elements and a fictitious contact material," *Computers & Mathematics with Applications*, vol. 70, no. 7, pp. 1370–1390, Oct. 2015.

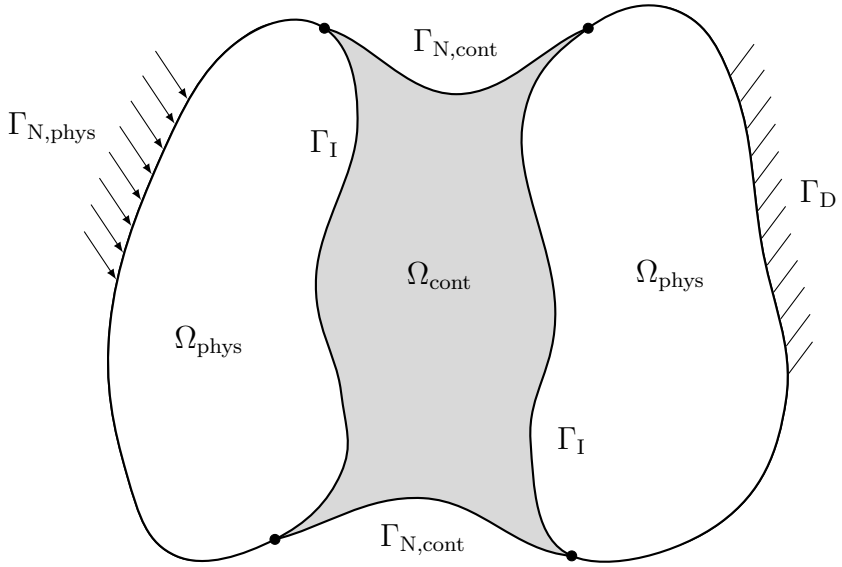


Figure 7.1: The physical and contact domains, Ω_{phys} and Ω_{cont}, respectively, are connected via interface boundaries Γ_{I}. Dirichlet boundaries Γ_{D} and the Neumann boundaries $\Gamma_{\mathrm{N,phys}}$ only touch the physical domain Ω_{phys}. The Neumann boundary $\Gamma_{\mathrm{N,cont}}$ is only part of the contact domain Ω_{cont}.

$$\boldsymbol{u}_{\mathrm{phys}} = \boldsymbol{u}_{\mathrm{cont}} \quad \text{on } \Gamma_{\mathrm{I}} \tag{7.5}$$
$$\boldsymbol{\sigma}_{\mathrm{cont}} \cdot \boldsymbol{n} = \boldsymbol{0} \quad \text{on } \Gamma_{\mathrm{N,cont}} \ , \tag{7.6}$$

where $\boldsymbol{\sigma}_{\mathrm{phys}}$ and $\boldsymbol{\sigma}_{\mathrm{cont}}$ are the Cauchy stress tensors in the physical and contact domain, respectively. $\boldsymbol{f}$ denotes a body load and $\boldsymbol{n}$ is the unit outward normal. Prescribed displacements and tractions are denoted by $\bar{\boldsymbol{u}}$ and $\boldsymbol{t}$, respectively. The total boundary is made up of non-overlapping Dirichlet (Γ_{D}), Neumann ($\Gamma_{\mathrm{N,phys}}, \Gamma_{\mathrm{N,cont}}$) and Interface ($\Gamma_{\mathrm{I}}$) parts (see Figure 7.1)

$$\partial\Omega_{\mathrm{phys}} \cap \partial\Omega_{\mathrm{cont}} = \Gamma_{\mathrm{I}} \tag{7.7}$$
$$\Gamma_{\mathrm{I}} \cup \Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N,phys}} = \partial\Omega_{\mathrm{phys}} \tag{7.8}$$
$$\Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{N,phys}} = \emptyset \tag{7.9}$$
$$\Gamma_{\mathrm{I}} \cap (\Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N,phys}}) = \emptyset \tag{7.10}$$
$$\Gamma_{\mathrm{I}} \cup \Gamma_{\mathrm{N,cont}} = \partial\Omega_{\mathrm{cont}} \tag{7.11}$$
$$\Gamma_{\mathrm{I}} \cap \Gamma_{\mathrm{N,cont}} = \emptyset \ . \tag{7.12}$$

7.2 Contact material formulation

The contact material used in this work is based on the isotropic, hyperelastic formulation by Hencky, introduced in Chapter 2.

7.2.1 Stress and constitutive tensor of contact material

The contact material is introduced as a means of approximately satisfying the Karush-Kuhn-Tucker (KKT) conditions, which relate the normal gap distance g to the normal traction t_{N}

in the contact domain (see Chapter 2). The material parameters μ and Λ are scaled by a parameter c, which will be denoted as contact stiffness in the remainder of this work. Thus, the principal stresses $\sigma_{ii,\text{cont}}$ obtained from the contact material regularize the KKT conditions, as shown in Figure 7.2 for the one-dimensional case.

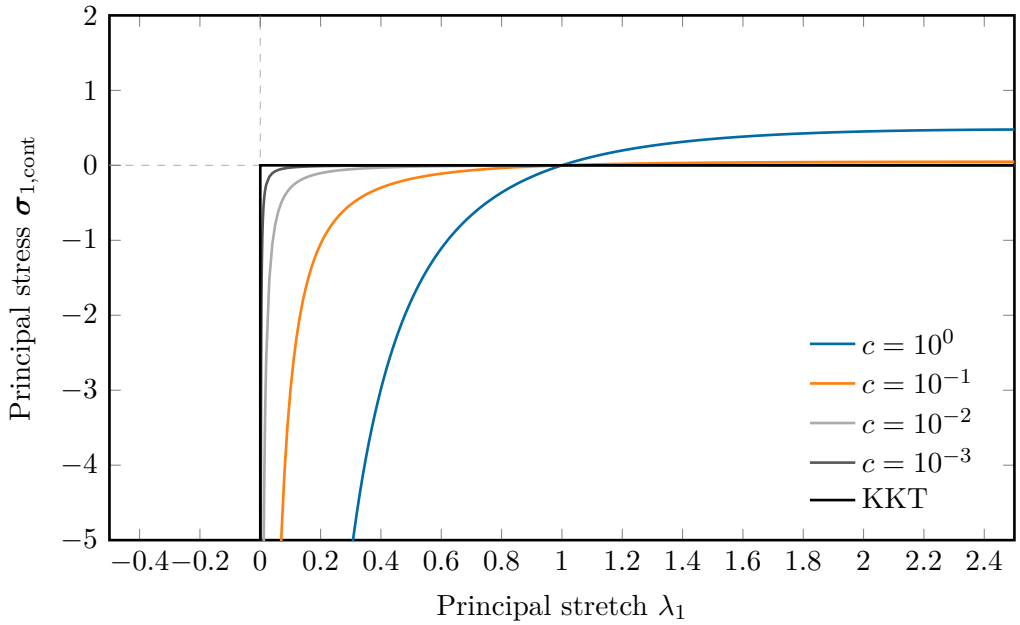


Figure 7.2: The minimum principal stretch and its corresponding stress are directly related to the gap function g and the normal traction t_N, respectively. Hence, the principal stretch can be considered as a normalized gap function. The contact material, thus, regularizes the KKT conditions via scaling of the material parameters using a contact stiffness c ($\mu = \Lambda = 1.0$).

Here, the stretches λ_i correspond to a normalized gap distance, while the principal stresses $\sigma_{ii,\text{cont}}$

$$\sigma_{ii,\text{cont}}\left(\lambda_1, \lambda_2, \lambda_3\right) = \frac{c}{J}\left(2\mu \ln \lambda_i + \Lambda \ln J\right) \tag{7.13}$$

represent the reaction force. Thus, the Cauchy stress tensor for the contact domain reads

$$\boldsymbol{\sigma}_\text{cont} = \sum_{i=1}^{3} \sigma_{ii,\text{cont}} \boldsymbol{n}_i \otimes \boldsymbol{n}_i \ . \tag{7.14}$$

The principal stretches λ_i and their associated directions $\boldsymbol{n}_i$ are obtained from the eigenvalue problem [17]

$$\left(\boldsymbol{b} - \boldsymbol{I}\lambda_i^2\right) \boldsymbol{n}_i = 0 \ , \tag{7.15}$$

where $\boldsymbol{b}$ is the left Cauchy-Green tensor((2.12b). Using Equations (2.40) and (2.41), the Cartesian components of the spatial elasticity tensor follow as

$$c_{\text{cont},ijkl} = \sum_{i,j=1}^{3} \frac{c}{J}\left(2\mu\delta_{ij} + \Lambda\right) \boldsymbol{n}_{iijj} - \sum_{i=1}^{3} 2\sigma_{ii,\text{cont}} \boldsymbol{n}_{iiii}$$

$$+\sum_{\substack{i,j=1\\ i\neq j}}^{3} \frac{\sigma_{ii,\mathrm{cont}}\lambda_j^2 - \sigma_{jj,\mathrm{cont}}\lambda_i^2}{\lambda_i^2 - \lambda_j^2} \left(\boldsymbol{n}_{ijij} + \boldsymbol{n}_{ijji}\right), \tag{7.16}$$

The choice of the parameter c largely depends on the initial gap between contacting bodies. The larger the initial gap, the earlier small stretch values—and thus significant reaction forces—will be encountered during the load history.

7.3 Discussion of frictionless behavior

The contact material formulation at hand is frictionless since no energy dissipation is included. Strictly speaking, the formulation is not free of sticking, but the following simplified investigation shows that sticking reactions are orders of magnitudes smaller than the normal reaction forces. Since a tangential gap leads to a shear deformation of the contact domain (Figure 7.3), shear stresses are responsible for enforcing sticking.

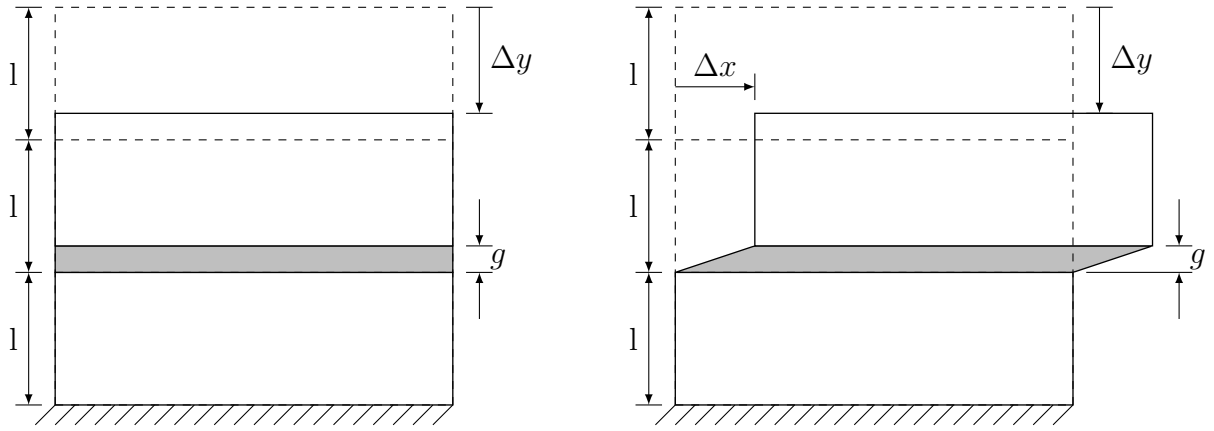


Figure 7.3: Two physical bodies (white) are connected by a contact domain (gray) containing the contact material formulation. a) Pure compression of the contact domain. b) Relative lateral displacement of the two bodies leads to a shear deformation of the contact domain.

These stresses, however, do not grow substantially during the compression and shear deformation of the contact material, which can be seen in Figure 7.4. Here, components of the stress tensor are plotted versus the shear parameter γ in a highly compressed contact zone ($\lambda_{\min} = 10^{-4}$). This deformation is described by the following deformation gradient:

$$\mathbf{F} = \begin{bmatrix} 1 & \gamma \\ 0 & 10^{-4} \end{bmatrix}. \tag{7.17}$$

The shear component σ_{xy} turns out to be orders of magnitude smaller than the corresponding compression stress σ_{yy}.

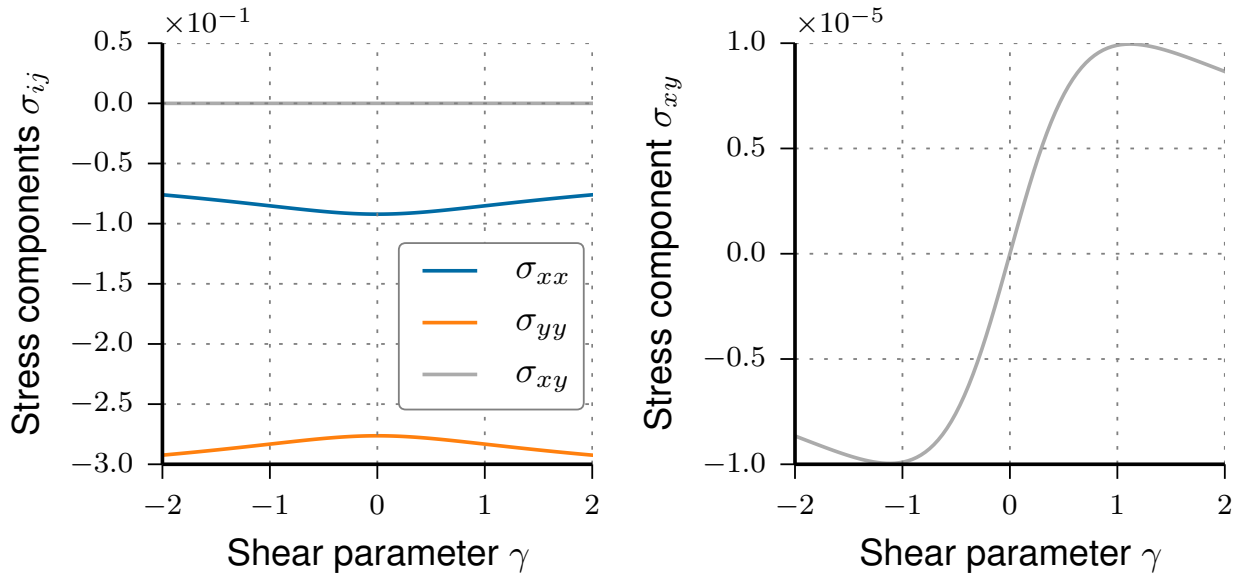


Figure 7.4: Stress components of the two-dimensional Cauchy stress tensor for an initial compression ($\lambda_2 = 10^{-4}$) of the contact material and a varying shear parameter γ. The shear stress σ_{xy}, shown also in detail on the right, is several orders of magnitude smaller than the component σ_{yy}, which is related to the contact pressure ($c = 10^{-6}, \mu = 1.0, \lambda = 1.0$).

7.4 Line search to prevent element collapse

The inherent nonlinearity of contact problems reflects in the nonlinearity of the contact material formulation at hand. Like in most nonlinear problems, the solution is obtained using the full Newton-Raphson method. Since the contact material is highly compliant at the beginning of the load history, it can happen that displacement increments are computed, which lead to an overlap of adjacent contact boundaries. For classical formulations, like the penalty or Lagrange multiplier method, this poses no further difficulty, as penetrations are removed in the succeeding iterations. Yet, the approach at hand can be classified as a barrier-type approach, which is why the solution has to be admissible at all times. To this end, measures must be taken to ensure that adjacent contact interfaces do not interpenetrate. A straightforward approach would be to cut load increments, if such a case is detected (e.g. by evaluating the determinant of the deformation gradients at integration points). Since this could lead to very small load increments and, in consequence, to a significant increase in computation time, we utilize an inexact line search based on the determinant of the deformation gradient:

$$\Delta \boldsymbol{u}^i_{k,m+1} = \Delta \boldsymbol{u}^i_{k,m} - \frac{f_m}{f_m - f_{m-1}} \cdot \left(\Delta \boldsymbol{u}^i_{k,m} - \Delta \boldsymbol{u}^i_{k,m-1} \right) \ , \tag{7.18}$$

$$f_m = \min \left(\det \boldsymbol{F}^i_{k,m}(\boldsymbol{X}_q) \right) , \ \forall \ \boldsymbol{X}_q \in \Omega_{\text{cont}} \ . \tag{7.19}$$

Here, $\Delta \boldsymbol{u}^i_{k,m}$ denotes the modified displacement increment at load step i, Newton iteration k and line search sub-iteration m. The quantity f_m is the minimum of all determinants of the gradient of deformation $\boldsymbol{F}$ at the integration points $\boldsymbol{X}_q$ in the contact domain Ω_{cont} at sub-iteration m. A combined stopping criterion is employed, which takes into account the change

of the modified displacement increment and the resulting minimum of the all determinants:

$$\Big(\big(\|\Delta \boldsymbol{u}^i_{k,m+1} - \Delta \boldsymbol{u}^i_{k,m}\| < \epsilon_\Delta\big) \vee \big(\|f_{m+1}\| < \epsilon_f\big)\Big) \wedge \big(f_{m+1} > 0\big) \ . \tag{7.20}$$

Here, ϵ_Δ and ϵ_f are stopping tolerances for the change of solution increment and the absolute value of f, respectively. In all numerical examples using the contact material approach, these tolerances have been set to 10^{-6}.

7.5 Numerical examples

In this section the capabilities of the material contact formulation are investigated using different numerical examples.

7.5.1 A large deformation model problem including self-contact

First, the influence of different parameters are investigated using a two-dimensional model problem in the large deformation regime. These parameters include the ansatz order p and the order of integration q inside the contact domain as well as the contact stiffness c, as defined in Section 7.2. Furthermore, the contact material formulation is compared to a classical h-FEM formulation using the commercial FEM-package ANSYS[†].

7.5.1.1 Definition of the model problem

The model problem under consideration is a rectangular block with a slotted hole subjected to a constant, vertical load as depicted in Figure 7.5. The physical part, shown in white, contains a plane-strain, neo-Hookean material (Section 2.3.2.3), whereas the slot (gray) is filled with the contact material model given in equation (7.16). Fillets at the corners of the slot are treated according to the finite cell method as described in Chapter 4. All boundary parts that may come into contact are discretized conformingly, i.e. element edges match the boundaries exactly. The total load is applied in 10 uniform, incremental steps.

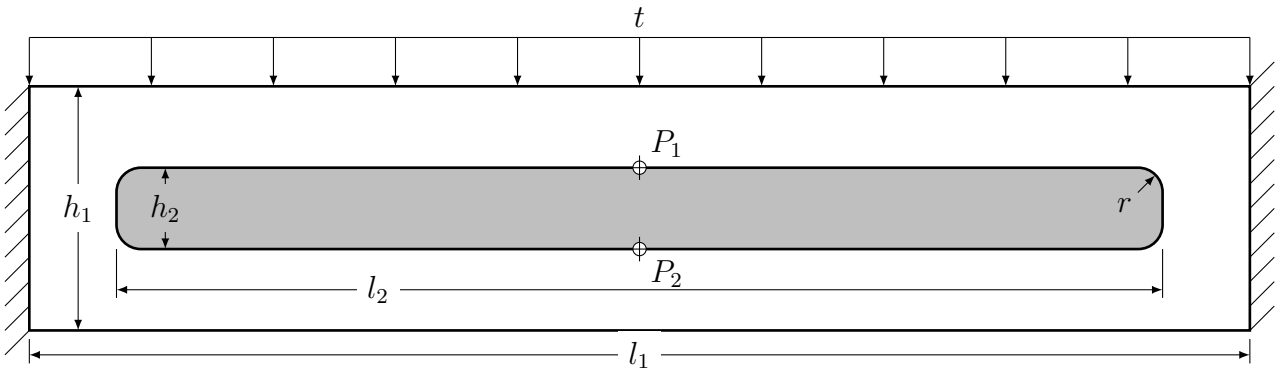
7.5.1.2 Choice of the polynomial degree in the contact domain

The aim of this first analysis is to examine the influence of the high-order shape functions inside the contact domain. To this end, the model problem introduced in the previous section is solved using different ansatz orders. Figure 7.6 shows the evolution of the residual measure obtained in the Newton-Raphson method versus the cumulative number of iterations for different polynomial degrees and a contact stiffness of $c = 10^{-6}$. The residual measure used here is defined as

$$r(\mathbf{u})^i_k = \sqrt{\mathbf{G}^i_k \cdot \Delta \mathbf{u}^i_k}, \tag{7.21}$$

where $\mathbf{G}^i_k$ and $\Delta \mathbf{u}^i_k$ are the residual (4.17) and the solution increment at load step i and iteration k, respectively. The convergence tolerance for the Newton solver is set to 10^{-6}. The total load is applied in 10 equally sized steps. The maximum number of iterations for all

[†]ANSYS®, *Academic Research, Release 14.5*



Physical Domain : $l_1 = 5.0$ $h_1 = 1.0$ $E_{\text{phys}} = 1.0$ $\nu_{\text{phys}} = 0.3$ $t = 0.005$

Contact Domain : $l_2 = \frac{7}{2} l_1$ $h_2 = \frac{1}{3} h_1$ $E_{\text{fict}} = c \cdot E_{\text{phys}}$ $\nu_{\text{fict}} = \nu_1$ $r = 0.1$

Figure 7.5: Model problem setup: Rectangular block with a slotted hole fully supported at both sides and subjected to a vertical, distributed load. The physical domain (white) contains the neo-Hookean material introduced in Section 2.3.2.3. The slotted hole (gray) is filled with the contact material described in Section 7.2.

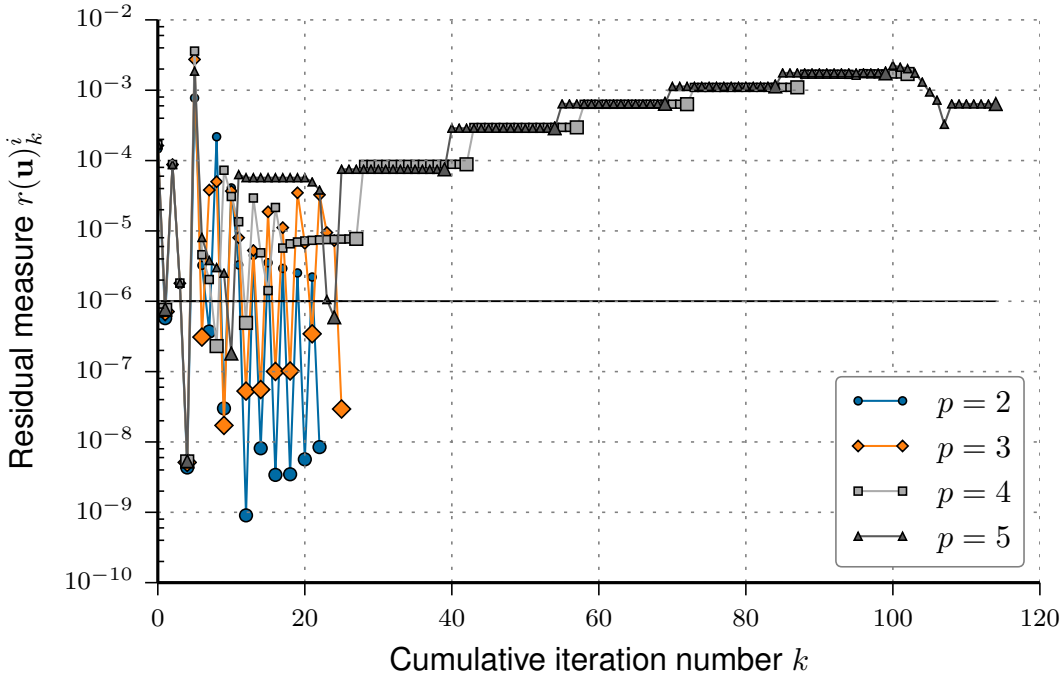


Figure 7.6: Convergence progress of Newton-Raphson method for uniform ansatz orders throughout the computational domain. Intermediate and final iterations for each load increment, including possible line searches, are depicted by dedicated small and large markers, respectively.

computations is restricted to 15 iterations. When applying the same ansatz order in the physical domain and the contact domain, the final solution only converges up to the desired

tolerance for $p = 2$ and $p = 3$. The contact stiffness in this analysis is rather low ($c = 10^{-6}$), and the elements in the middle are extremely compressed (Figure 7.7). Minor perturbations in

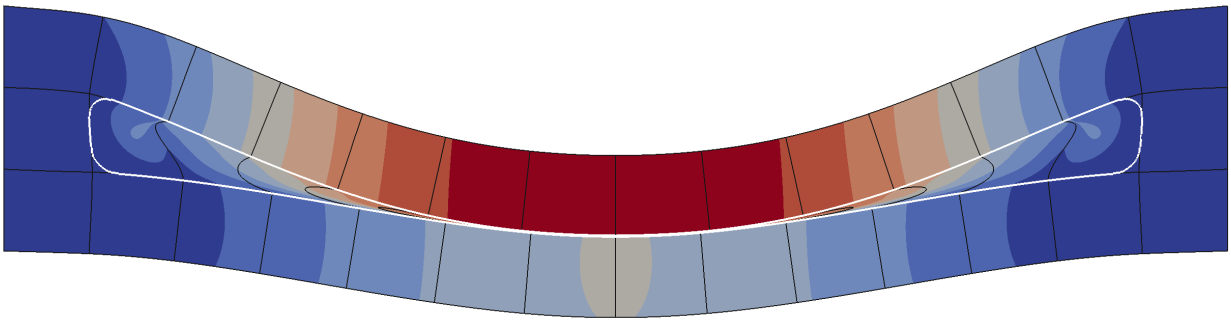


Figure 7.7: Displacement solution of the model problem for the highest ansatz order that converged ($p = 3$). The highly deformed edges inside the contact domain are clearly visible

the numerical solutions might lead to an invalid state, i.e. the elements collapse locally, and the respective Jacobians of the deformation gradient become negative. This is, of course, likely to happen during the discrete steps made during the Newton iterations. The line search proposed in section 7.4 tries to avoid this by reducing the current increment. However, the resulting configuration might not be close enough to the equilibrium state. Obvious possibilities to overcome this problem are an increase of the contact stiffness as can be seen from Figure 7.8 or the reduction of the load increment. The latter approach does not immediately lead to stable results for the given contact stiffness of $c = 10^{-6}$, as can be seen in Figure 7.9. However, if the contact stiffness is reduced to $c = 10^{-5}$, smaller load increments lead to converging results at the cost of a much higher number of computations (see Figure 7.10).
A third alternative is the deactivation of the high-order modes inside the contact domain, which is illustrated for the two-dimensional case in Figure 7.11. The goal of the simulation is not a good approximation of the deformation state in the fictitious contact domain, but the reaction in normal direction to the contact boundaries. By deactivating high-order modes inside the contact domain, physical interfaces can still deform according to their high polynomial degree and the robustness of the solution is drastically improved (see Figure 7.12). Also, the deactivation does not impose an overhead. On the contrary, it lies in line with the usual implementation of hierarchic shape functions in which different polynomial orders may be assigned to edges, faces and volumes. To ensure continuity, adjacent edges (or faces in 3D) are set to the highest degree of all involved entities. Therefore, it is only necessary to set all modes of the elements within the contact domain to $p = 1$, and the rest can be handled by the standard routines to ensure compatibility. Additionally, the total number of degrees of freedom is reduced, which has a positive effect on the computational efficiency, see Figure 7.13.
Since this modification allows us to use larger and thus fewer load increments and also reduces the total number of degrees of freedom(see Figure 7.13), we apply this approach in all of the following examples.

7.5.1.3 Influence of integration order q on minimum contact gap $g_{\min}$

Classical approaches for the simulation of contact problems usually track the status of contact at discrete points [133]. The current approach evaluates the principal stretches at integration

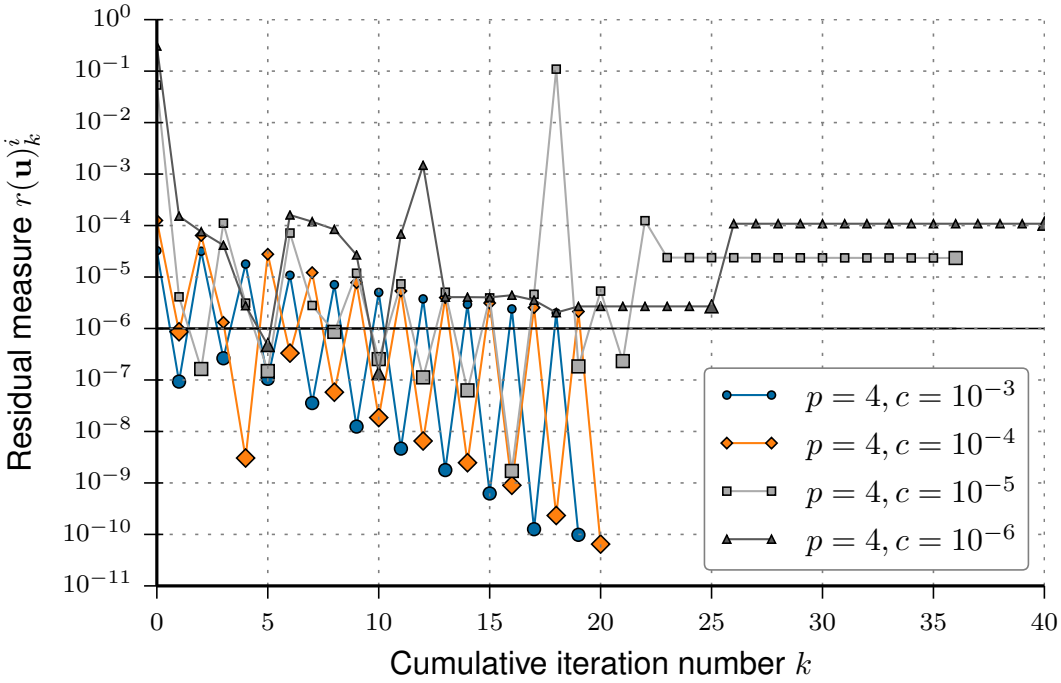


Figure 7.8: Convergence progress for different contact stiffnesses c and a uniform ansatz order of $p = 4$ in all elements.

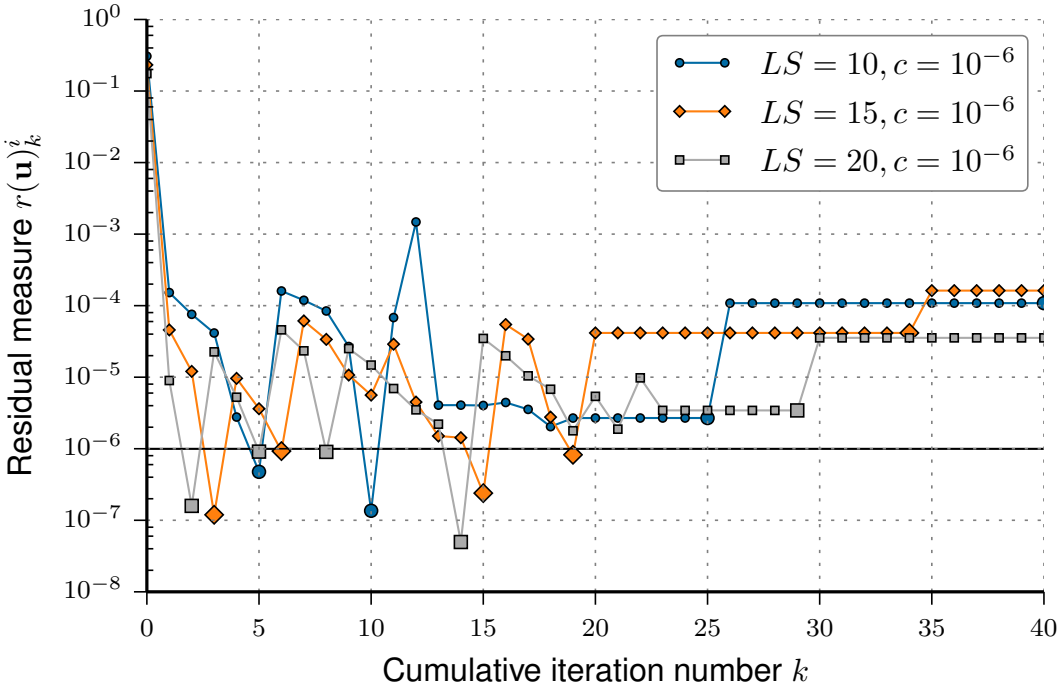


Figure 7.9: Convergence progress for a different number of load steps LS, a uniform ansatz order of $p = 4$ in all elements and a contact stiffness of $c = 10^{-6}$.

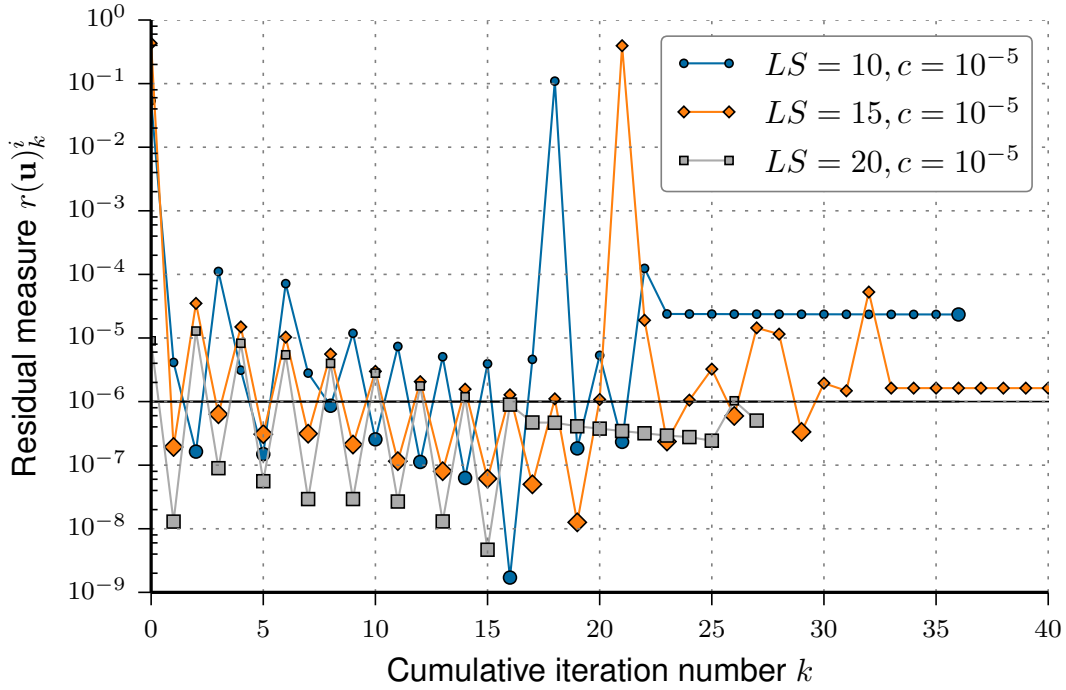


Figure 7.10: Convergence progress for a different number of load steps LS, a uniform ansatz order of $p = 4$ in all elements and a contact stiffness of $c = 10^{-5}$.

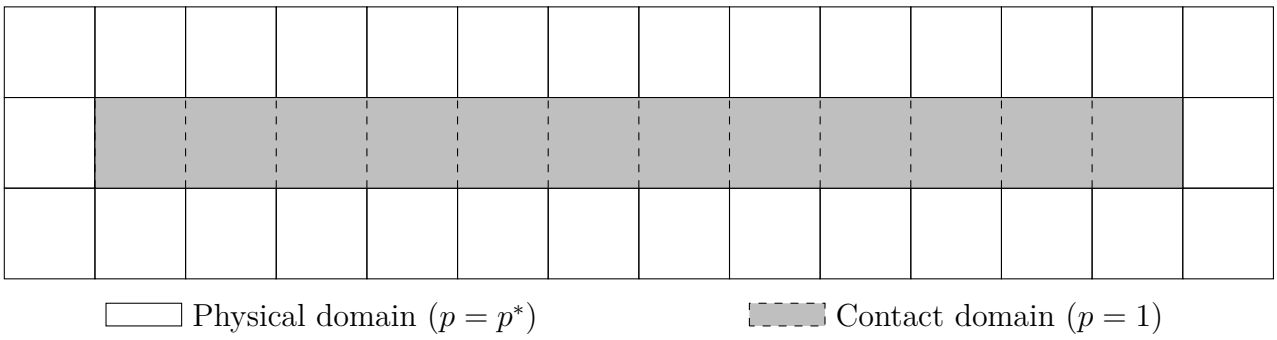


Figure 7.11: The domain of the model problem given in Section 7.5.1.1 is discretized by 14×3 high-order finite elements. All higher modes (i.e. edge and face modes) of elements, which are fully or partially in the contact domain (gray), are deactivated. Edge modes touching the physical and the contact domain remain active

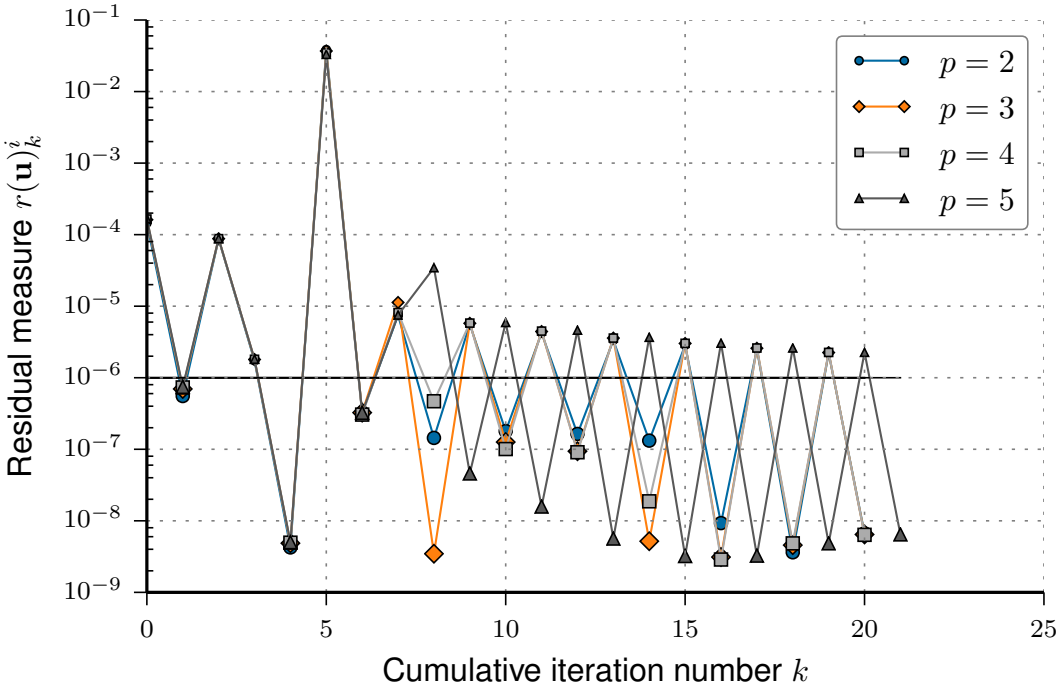


Figure 7.12: Convergence progress of Newton-Raphson method for modified ansatz orders inside the contact domain.

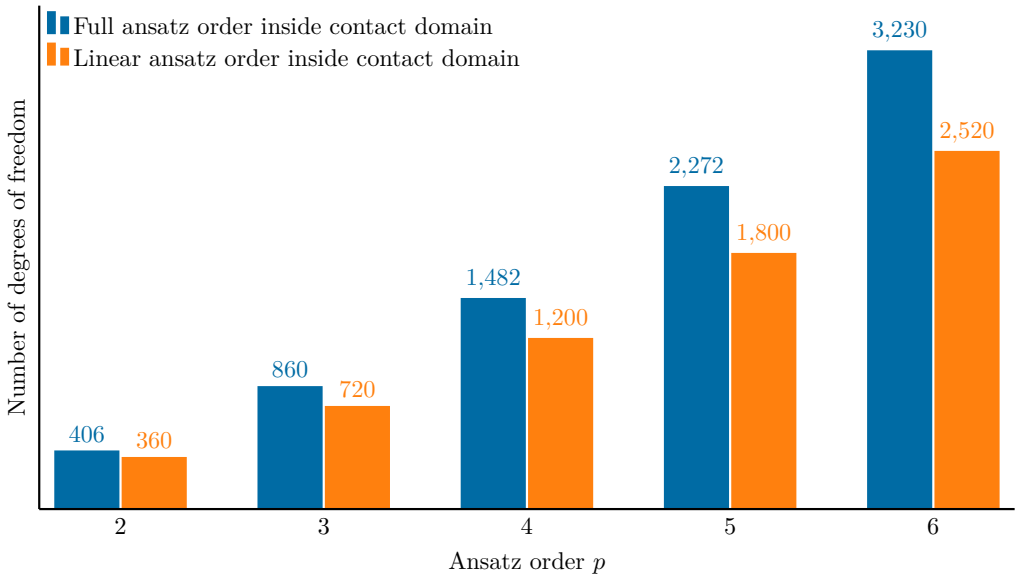


Figure 7.13: Total number of degrees of freedom for full (blue bars) and modified (orange bars) ansatz orders inside the contact domain.

points during the numerical integration of the discretized weak form. Contact is established if one of the principal stretches approaches zero. Thus, the integration points act as sampling points for contact detection. Therefore, the influence of the number of integration points on the resulting contact gap is investigated. To this end, the gap is computed for different orders of numerical integration q and for different ansatz orders p. Note that the ansatz orders are only varied inside the physical domain (Section 7.5.1.2). All computations draw on the standard Gauss-Legendre scheme. The contact gap is measured as the distance between the points P_1 and P_2 (Figure 7.5). The results are computed using a contact stiffness of $c = 10^{-6}$. The obtained minimum contact gaps (g_{min}) normalized by the initial gap ($g_0 = 1/3$) are summarized in Table 7.1. The results show that the normalized contact gaps remain close to 4% for the

Table 7.1: Influence of the integration order q on the resulting contact gap g ($c = 10^{-6}$)

$p = 2$		$p = 3$		$p = 4$	
q	$g_{\text{final}}/g_0[\%]$	q	$g_{\text{final}}/g_0[\%]$	q	$g_{\text{final}}/g_0[\%]$
3	3.97008949146	4	4.00308944753	5	4.01568941207
4	3.96618949065	5	4.00578943907	6	4.01688941207
5	3.97008949146	6	4.00698943907	7	4.01568940280
6	3.97128949146	7	4.00578943907	8	4.01598941207
7	3.97008949146	8	4.00608943907	9	3.96138927748

investigated configurations.

For this setup, the order of integration does not have a significant influence on the resulting minimum contact gap, and should thus only be governed by the polynomial degree p to precisely integrate the element matrices. We will further investigate the influence of the integration order on contact detection for a setup involving curved boundaries in a later section.

7.5.1.4 Influence of contact stiffness c on minimum contact gap g_{min}

Considering again the example setup depicted in Figure 7.5, the influence of the contact stiffness c on the minimum contact gap g_{min} is investigated. This is a study on the *model error*, as obviously even the exact solution using a contact domain with a finite c corresponds to a slightly different physical model. Figure 7.14 depicts the ratios of the resulting minimal gap g_{min} and the initial gap g_0 for ansatz orders $p = 2, 3, 4$. As can be seen, the ratios approach zero as the contact stiffness c is reduced. Also, the difference between the ratios computed with different ansatz orders is negligible. The contact material, therefore, converges to the limit state defined by the Karush-Kuhn-Tucker conditions for normal contact, as outlined in Chapter 3. Furthermore, the gap ratios for a contact stiffness of $c = 10^{-5}$ are already in the range of 10%, which is sufficient for many engineering applications. The gap ratios can also be reduced to a range of 1% by choosing $c = 10^{-6}$.

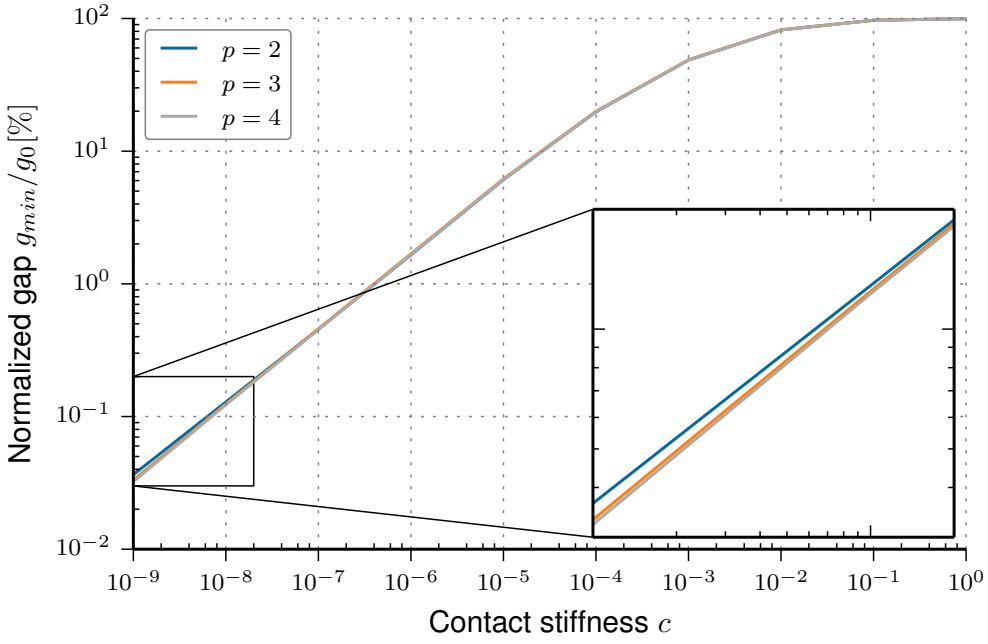


Figure 7.14: Contact stiffness c vs. ratio of minimum contact gap g_{min} and initial gap g_0. Results are obtained for ansatz orders $p = 2, 3, 4$.

7.5.1.5 Investigation of gap function g for different ansatz orders p compared to a classical h-FEM

In the following, the resulting contact gap g is investigated for different ansatz orders and contact stiffnesses. Figure 7.15 shows the gap values along the lower contact boundary for the ansatz orders $p = \{2, 4\}$ and contact stiffness $c = \{10^{-6}, 10^{-9}\}$.
The results are compared to values obtained using the commercial FE-package ANSYS (Figure 7.17). This ANSYS simulation uses 1600 quadratic elements of type *PLANE183* [3]. The curves of the contact material and the ANSYS results have the same characteristics. However, due to the different physical models represented by the approaches under comparison, the curves are offset by a constant value corresponding to the contact gap g_{min} for the respective stiffness, as discussed in Section 7.5.1.4. Drawing on the Augmented-Lagrange method [133], the ANSYS simulation is able to reduce the final gap to zero. The contact material approach, which can be interpreted as a barrier method [10], would only be able to realize a zero gap for the theoretical limit of $c \to 0$. However, the resulting gap distributions for the contact stiffness of $c = 10^{-9}$ closely follow the distribution of the ANSYS result.

7.5.1.6 Equivalent von Mises stresses σ_m compared to a classical h-FEM

The following example serves to compare the stress solution of the setup depicted in Figure 7.5 with a simulation conducted with ANSYS. The latter uses 1600 quadratic elements of type *PLANE183* [3]. The results for the contact material ($p = 5$) and ANSYS are shown in Figures 7.16 and 7.17, respectively. Both simulations result in similar stress distributions. However, the analysis using the contact material (1800 degrees of freedom) uses far fewer degrees of freedom, than the ANSYS simulation (10.480 degrees of freedom).

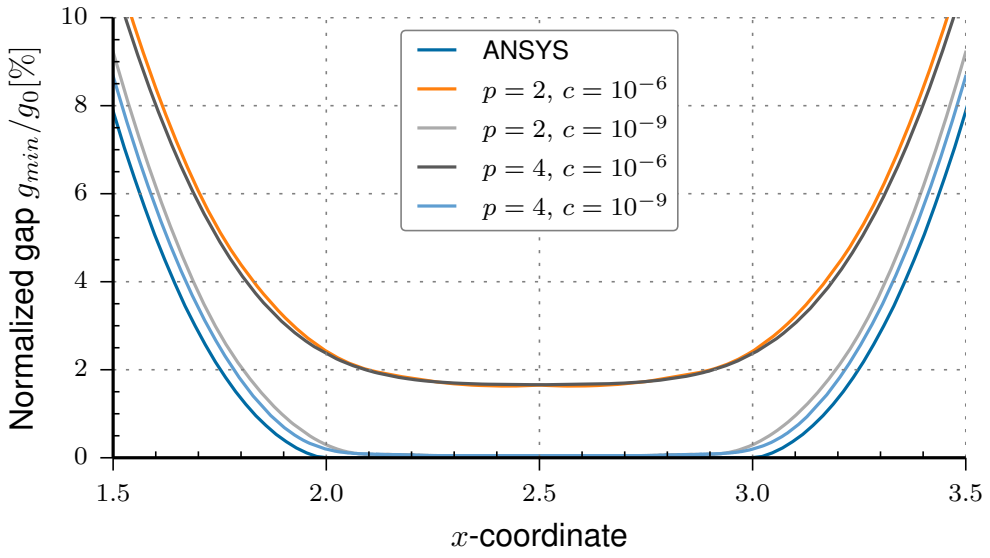


Figure 7.15: Comparison of resulting contact gaps g for different ansatz orders p with results obtained using ANSYS.

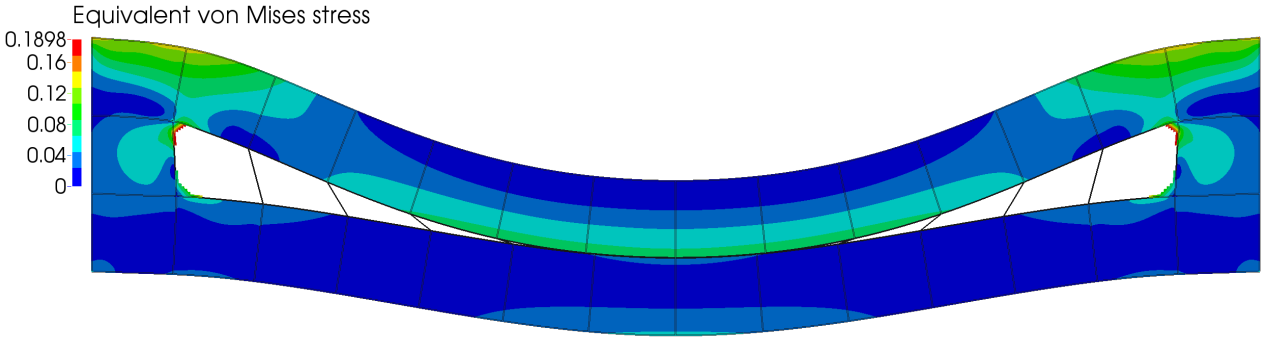


Figure 7.16: Equivalent von Mises stresses obtained using the contact material ($p = 5$)

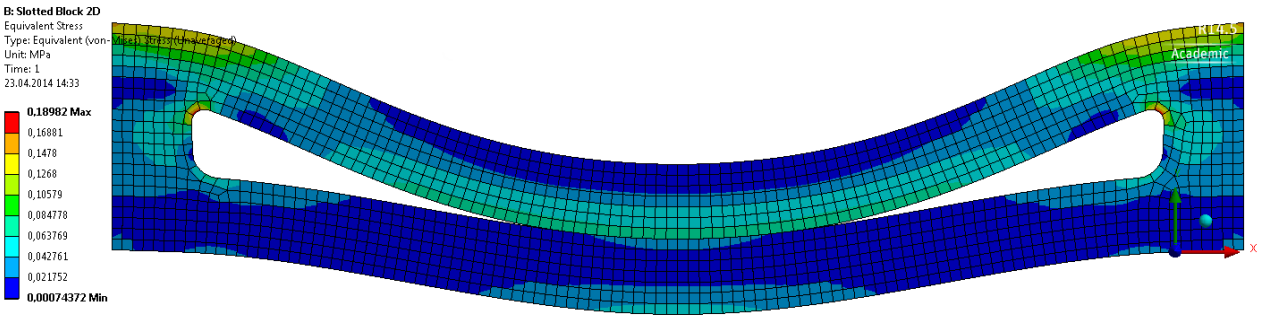


Figure 7.17: Equivalent von Mises stresses obtained by ANSYS

7.5.1.7 Comparison of normal pressures p

In the following, the normal contact pressures p computed with the contact material are compared with the values obtained by the ANSYS simulation (Figure 7.17). The pressures for the contact material are obtained by evaluating the stress tensor $\boldsymbol{\sigma}$ and the normals $\mathbf{n}$ along the points of a segmented line (10000 segments) on the lower contact interface (Figure 7.5). The normals are not continuous along the discrete line segments. Hence, the average of the two segment normals is taken for every evaluation point connecting two line segments. The normal pressure is then computed as:

$$p = \sigma_{ij} n_i n_j. \tag{7.22}$$

The graphs of both the h-FEM and the contact material results show the same general behavior, with similar values of normal contact pressure (Figure 7.18).

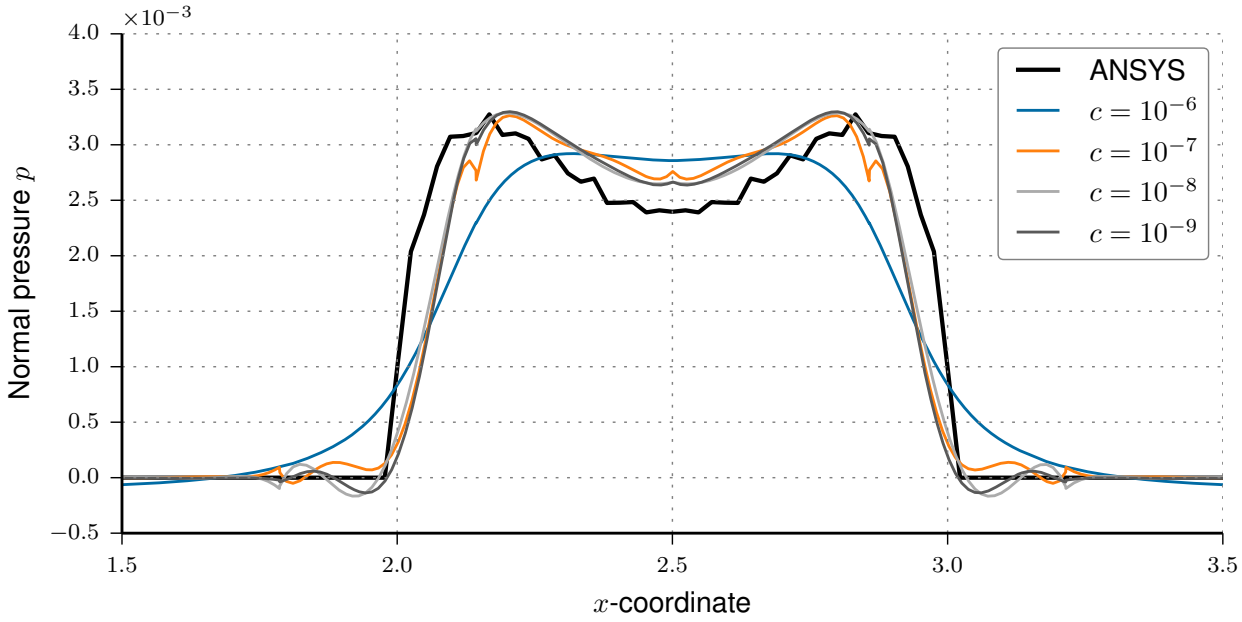


Figure 7.18: Comparison of resulting normal pressures p for different contact stiffnesses c with results obtained using ANSYS. The ansatz order is set to $p = 5$.

The contact material pressures converge with decreasing contact stiffness c, leading to a much smoother distribution than the ANSYS results. The h-FEM simulation is able to represent the kink at the contact boundary, while the p-FEM solutions show slight oscillations at the ends of the contact zone. This is a characteristic inherent to the p-version, but can be treated by different means including r-extension [85, 38] or hp-adaptivity [108, 140]. However, this is beyond the scope of this work.

7.5.2 Two-dimensional Hertzian contact

In this second example, we again consider the Hertzian frictionless contact problem between an infinitely long elastic cylinder and a rigid surface [51], which has also been investigated

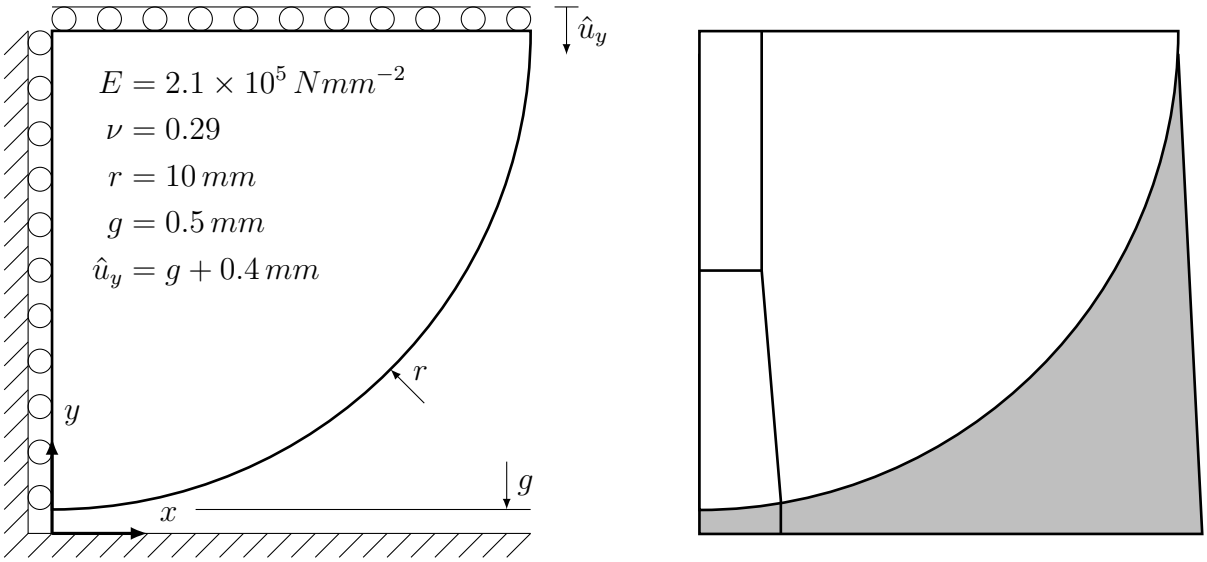


Figure 7.19: a) Setup for the Hertzian contact problem between an elastic cylinder and a rigid surface. b) Discretization for the contact material approach: elements containing the physical and contact material domain are depicted in white and gray, respectively. The circular boundaries are represented exactly using the blending function method [44]

in Section 6.4.1.2. Here, however, the model is discretized using conforming elements and is subjected to a higher load (see Figure 7.19).
Utilizing symmetries only a quarter of the cylinder is represented. Symmetry boundary conditions are applied in the y–z-plane. Following Franke et al. [37], the point load of the Hertzian model is represented by a constant prescribed displacement $\hat{u}_y$ at the horizontal center line of the cylinder. Franke et al. showed that this assumption is valid and that the resulting modeling error is negligible. The circular boundary is represented exactly using the blending function method [44]. Following this approach allows discretizing the geometry with a minimal number of five coarse high-order elements. Note that the mesh is constructed in such a way that the smallest element is 25% wider than the analytically computed contact width b (Equation (6.37)). This corresponds to the case where only a rough estimate is available of the final contact zone.

7.5.2.1 Contact normal pressure and stress distribution

The normal pressure distribution along the circular boundary is depicted in Figure 7.20 for different contact stiffnesses and different polynomial degrees.
The results are compared with the analytical solution of the Hertzian theory [51], for which the point load was obtained by integrating the stress component σ_{yy}, computed with $p = 20$ and $c = 10^{-6}$, along the upper edge of the model. The contact material pressures correspond well with the Hertzian distribution in the interior of the contact zone. For higher contact stiffnesses, the numerical solution regularizes the kink of the analytical solution. For lower contact stiffnesses, the solution follows the analytical solution much better—although it does show oscillations around the contact boundary. Similar results have been reported by Franke et al. [38]. The results can be improved drastically by moving the closest node to the analytical

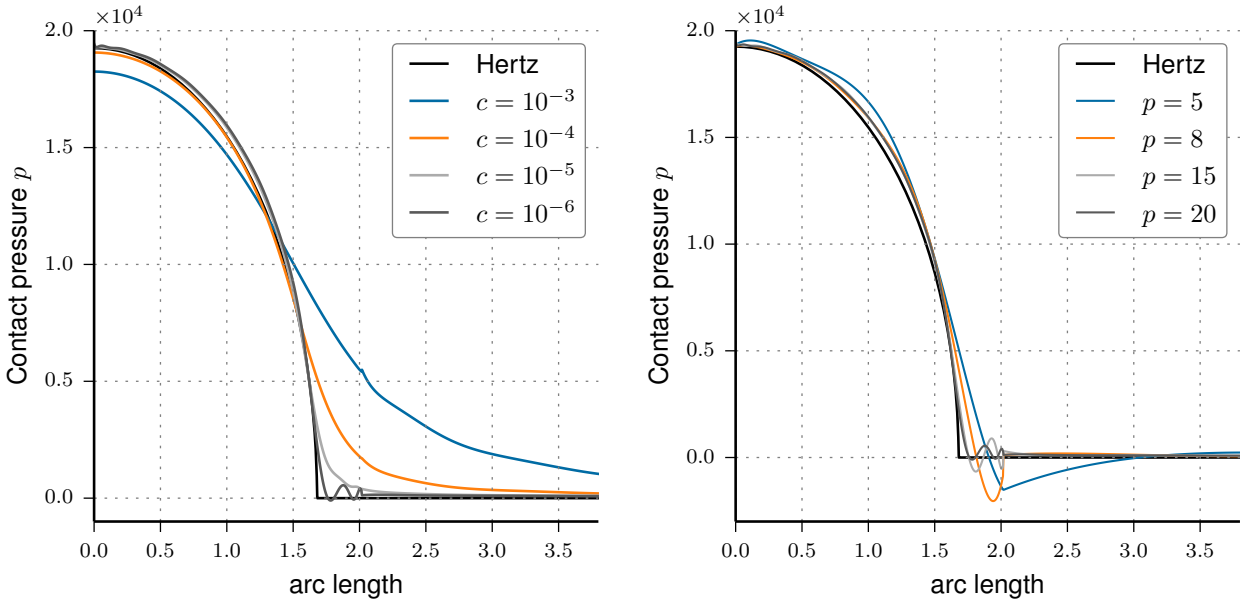


Figure 7.20: Normal contact pressure computed for a) different contact stiffnesses and a polynomial degree of $p = 20$ and b) different polynomial degrees and a contact stiffness of $c = 10^{-6}$. For reference, the Hertz solution is given for a point load of $50810N$

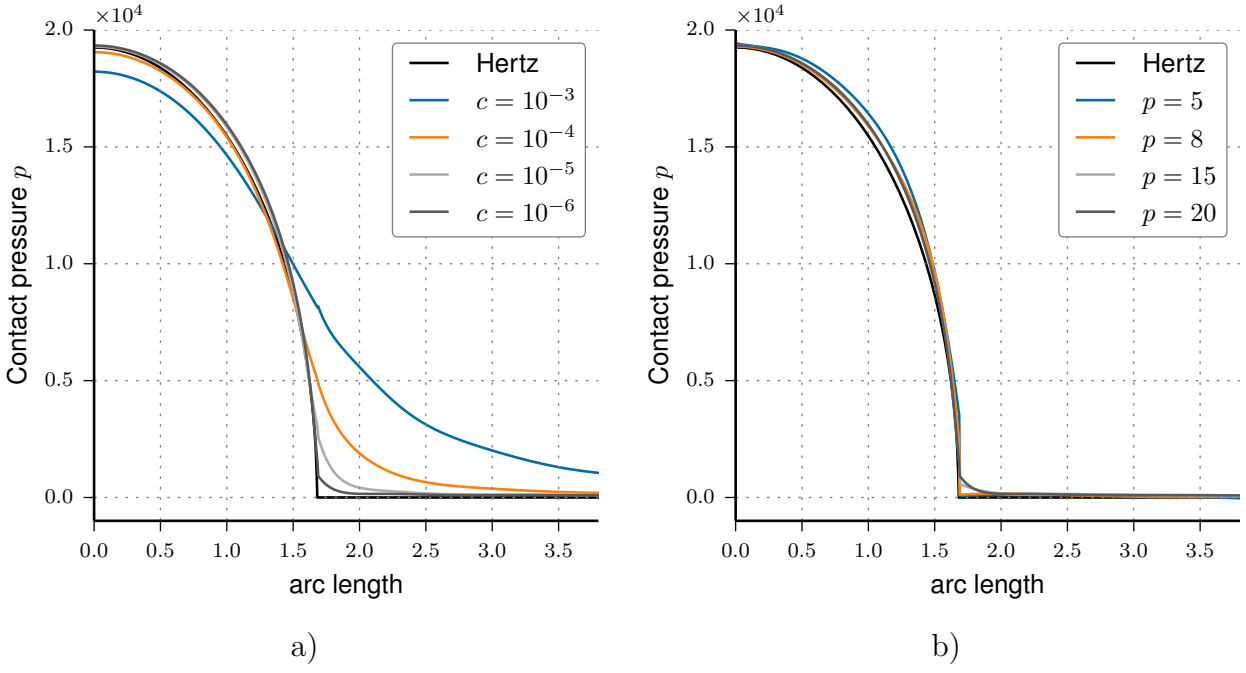
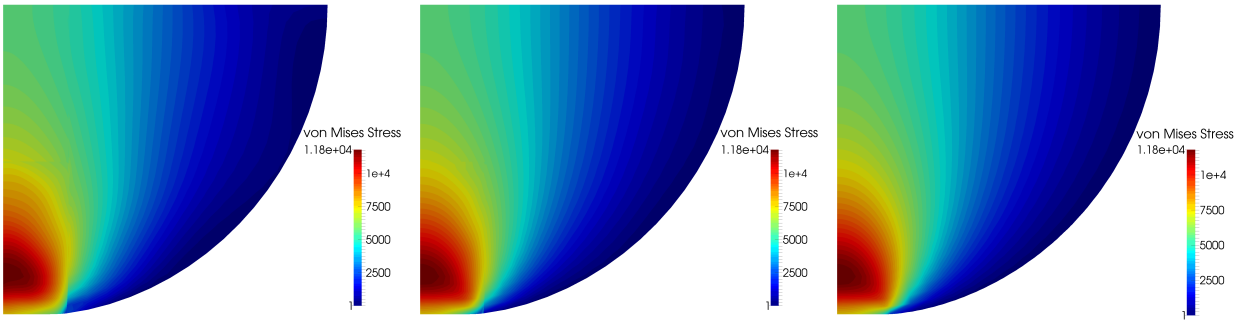


Figure 7.21: Relocating the nearest node onto the analytically obtained contact boundary drastically reduces any oscillations

contact boundary (Figure 7.21). This, of course is not generally suitable for all practical applications— especially in 3D—but it does emphasize the capabilities of the contact material formulation. Figure 7.22 depicts the global stress solution obtained with the contact material for ansatz orders $p = 5$ and 8.



a) b) c)

Figure 7.22: Stresses computed using the contact material. Results are shown for ansatz orders a) $p = 5$ and b) $p = 8$. As reference, results are provided for c) $p = 20$ using a mesh where the closest node has been moved to the analytical contact boundary

As reference, results are provided using a mesh of $p = 20$ where the closest node has been moved to the analytical contact boundary. The resulting distributions are typical for the current setup, see e.g. Franke [36]. They show that good global stress solutions can be obtained already for moderate ansatz orders.

7.5.2.2 Integration order for strongly curved contact zones

The influence of the number of integration points was already investigated for the large deformation example in Section 7.5.1.3, where no significant influence on the final gap could be observed. Now, the case of large elements with a high curvature is investigated. Therefore, we consider the previous setup in its full configuration—without utilizing symmetry conditions—and a much larger central element. This is depicted in Figure 7.23 together with a displacement solution using an ansatz order of $p = 3$ and an integration order of $q = 4$. Since there are no integration points in the center, the contact boundary penetrates the rigid target. Hence, the violation of the penetration condition is not fully resolved by the given integration order. For this special setup, this could easily be circumvented by an odd integration order. Figure 7.24 shows this nicely. Here, the gap value g at the center line of the model is plotted for a fixed ansatz order of $p = 3$ but with different orders of integration q. All the gap values for the even integration orders are negative but converge to zero. Odd integration orders avoid a penetration altogether and lead to much smaller gap values with little variance. It yet needs to be noted that this 'even-odd-peculiarity' is due to the special construction of this example, where the contact is exactly symmetric with respect to the midpoint of the corresponding element edge. A general remedy is to use a higher integration order in the elements containing contact material.

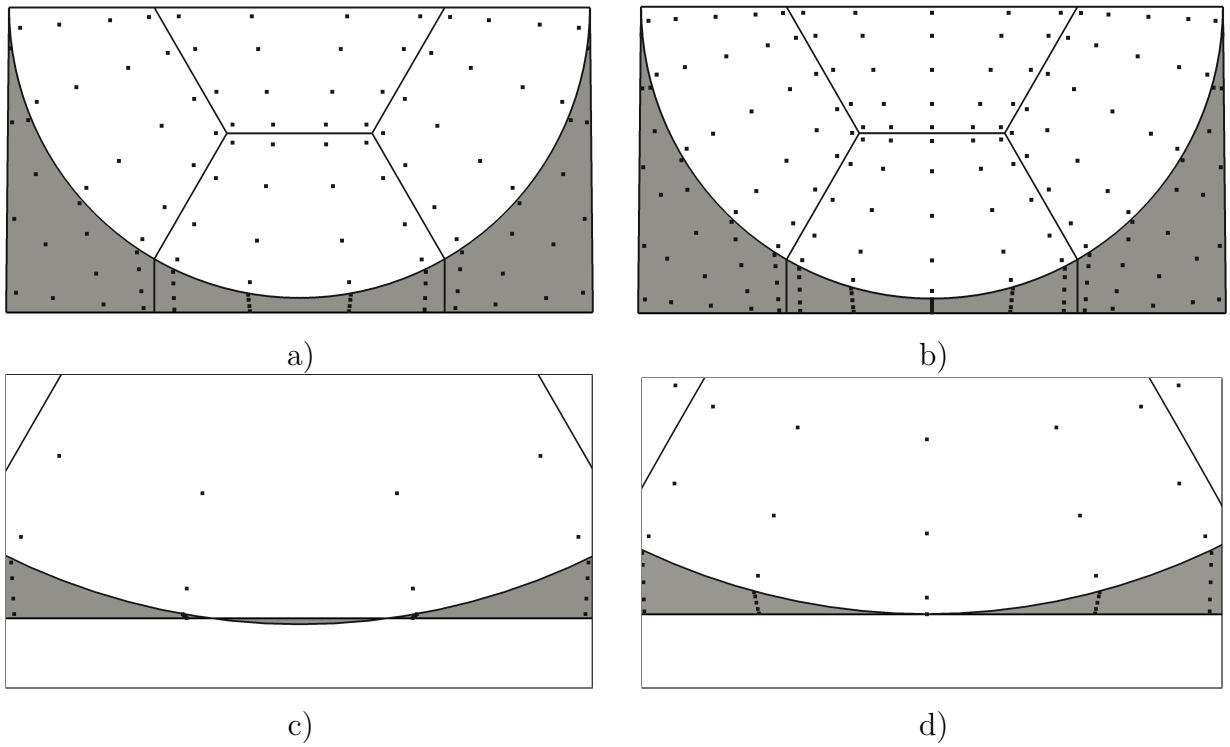


Figure 7.23: Mesh and integration points for integration order a) $q = 4$ and b) $q = 5$. Physical and contact domain are depicted in white and gray, respectively. c) Detail of the final deformed configuration using an ansatz order of $p = 3$ and $q = 4$. The penetration is clearly visible. d) For an odd integration order (here $q = 5$) no penetrations occur

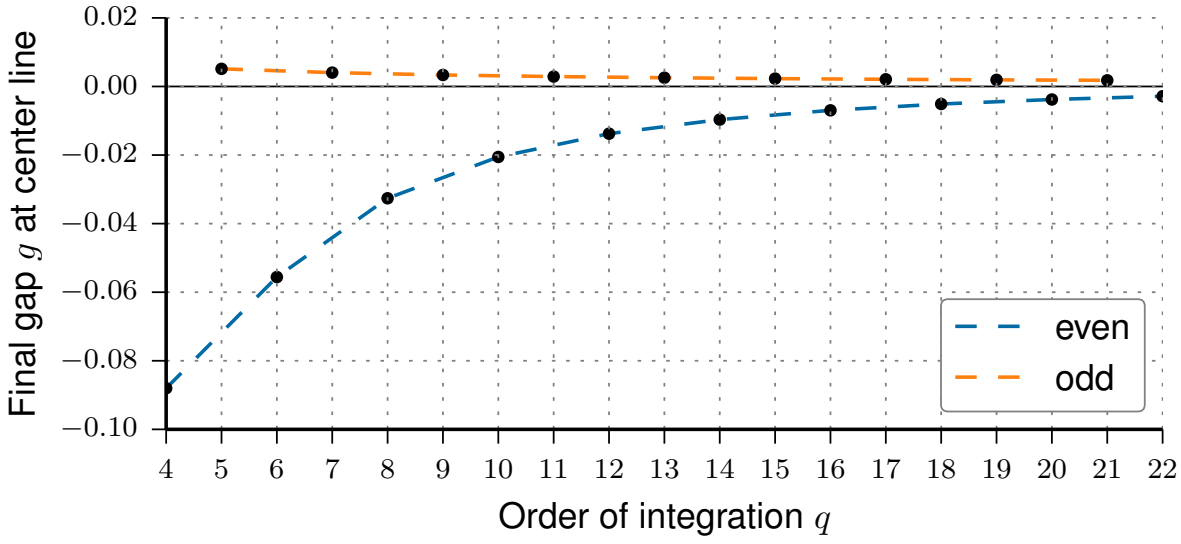


Figure 7.24: Final gap g at the center line of the model shown in Figure 7.23 versus integration order q. The initial gap was set to $g = 0.5$ and the radius to $r = 10$ ($E = 1.0$, $\nu = 0.3$, $\hat{u}_y = -0.6$, $c = 10^{-4}$)

7.5.3 A 3D example: elastic buffer element with multiple self-contact

The following example covers multiple self-contact in a three-dimensional setting. An elastic buffer element consisting of several thin-walled layers is subjected to a distributed, vertical surface load as shown in Figure 7.25.

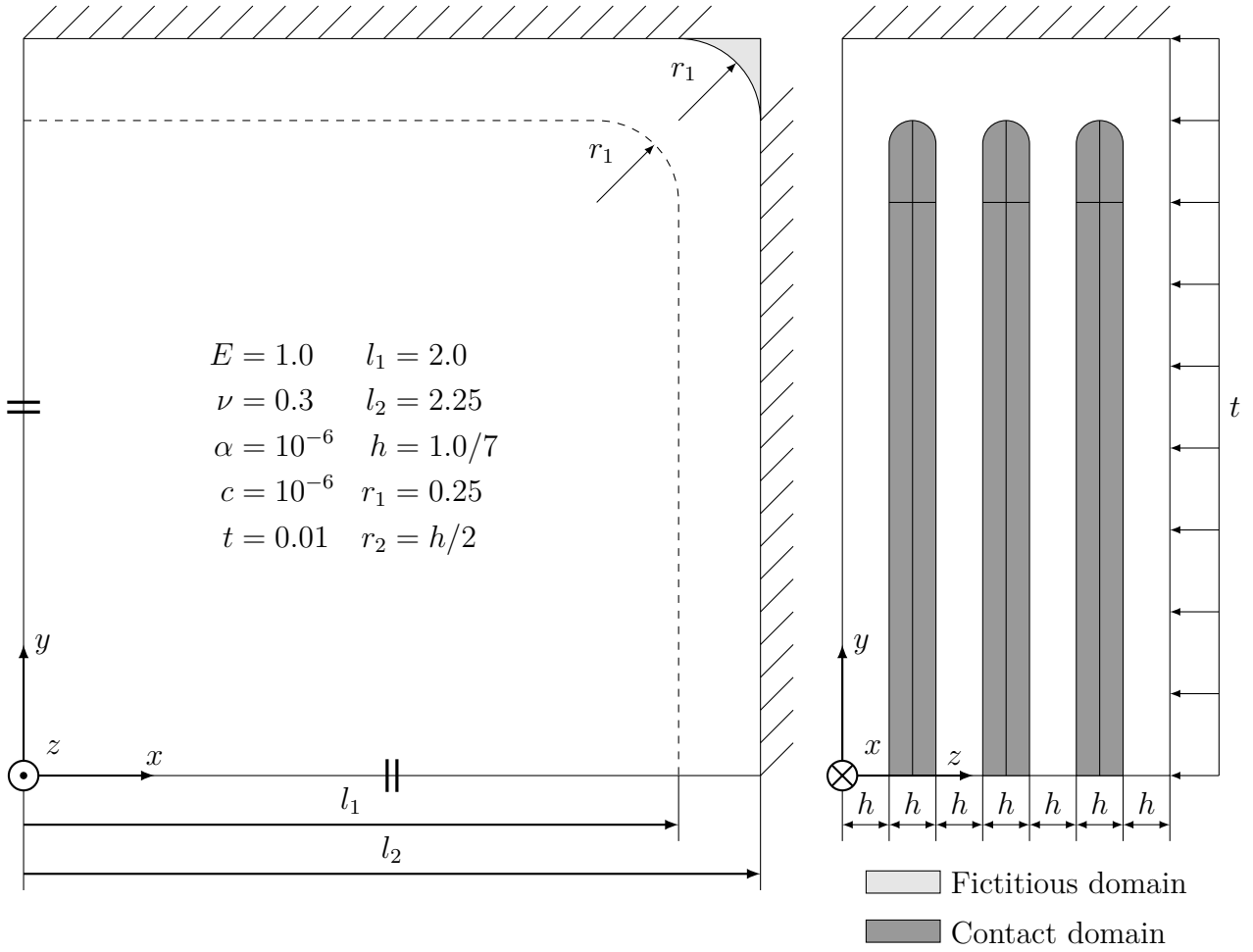
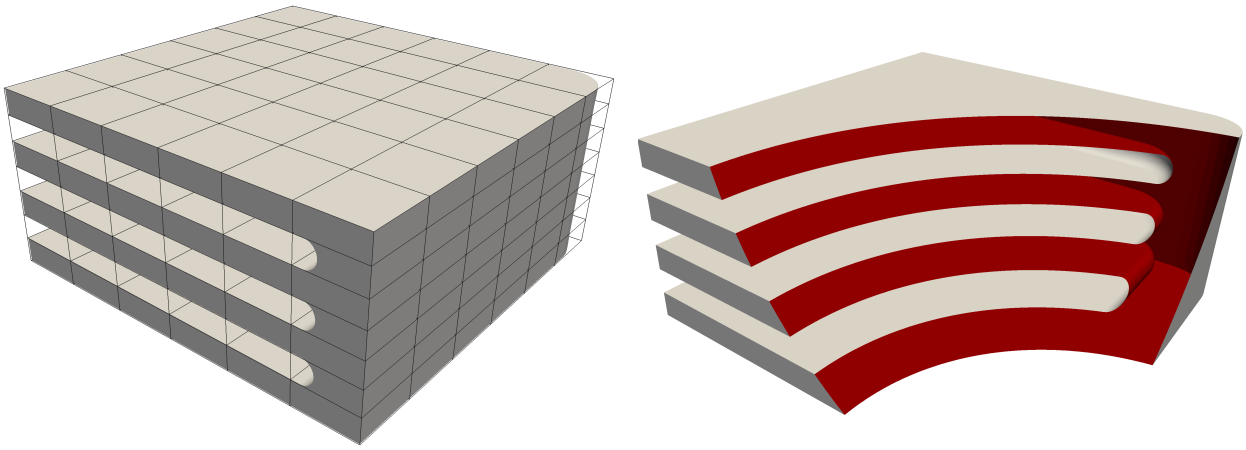


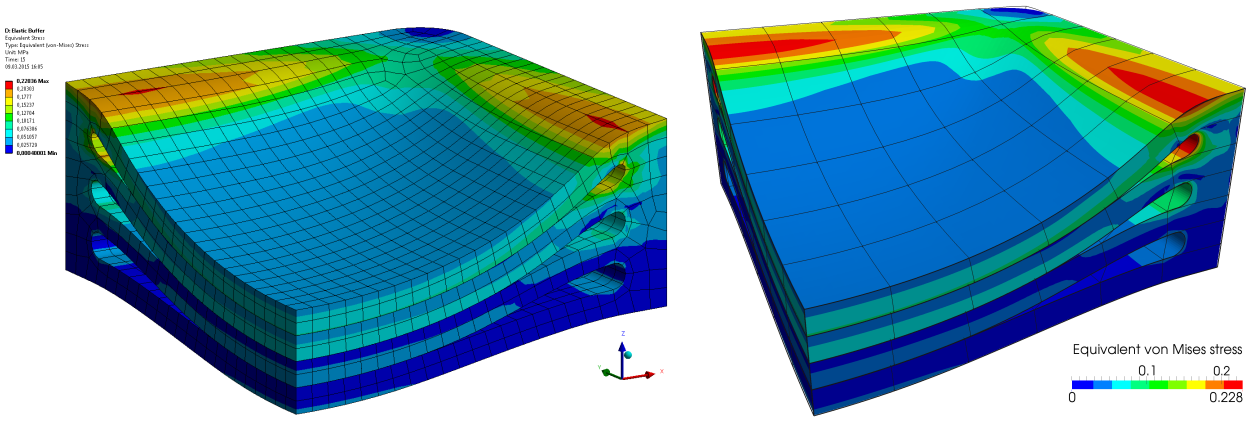
Figure 7.25: Schematic setup of complex 3D example: Elastic buffer element with multiple self-contact. The material parameters of the fictitious domain (light gray) as well as the contact domain (dark gray) are scaled by the parameters α and c, respectively. Exploiting symmetries along the x- and y-directions, the numerical model only considers a quarter of the damper, as shown

The geometry of this buffer element is embedded in a mesh of finite cells of ansatz order $p = 5$, as depicted in Figure 7.26. All cavities and fillets are considered at the integration level as explained in Section 4.5. Thus, it is possible to represent the small geometric features of the structure—such as the fillets—without explicitly meshing them. The equivalent von Mises stresses obtained from the contact material are compared to a simulation performed with ANSYS, see Figures 7.27a and 7.27b, respectively. The simulation using the contact material approach is performed with 18,990 degrees of freedom, whereas the ANSYS simulation employed quadratic hexahedra of type *SOLID186* and *SOLID187* [3], leading to 84,195 degrees of freedom. Although the contact material approach only uses one-fourth of the number of DOFs



(a) **(b)**

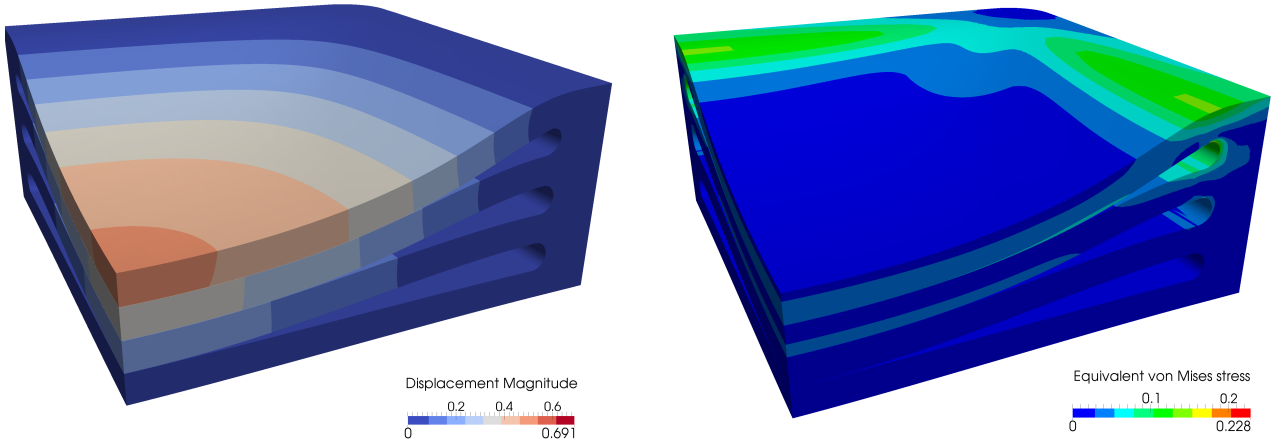
Figure 7.26: a) The buffer is embedded in a structured Cartesian grid of $6 \times 6 \times 7$ high-order finite elements ($p = 5$). b) A cut view of the boundary representation of the buffer. Inner fillets and cavities are recovered at the integration level as outlined in Section 7.26.



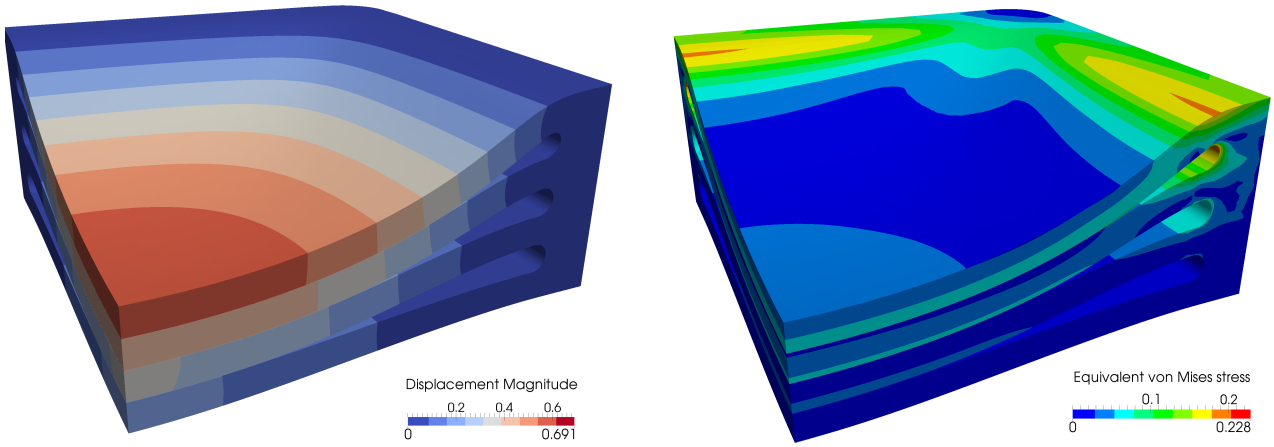
(a) ANSYS model **(b)** Contact material ($p = 5$)

Figure 7.27: Comparison of equivalent von Mises stresses.

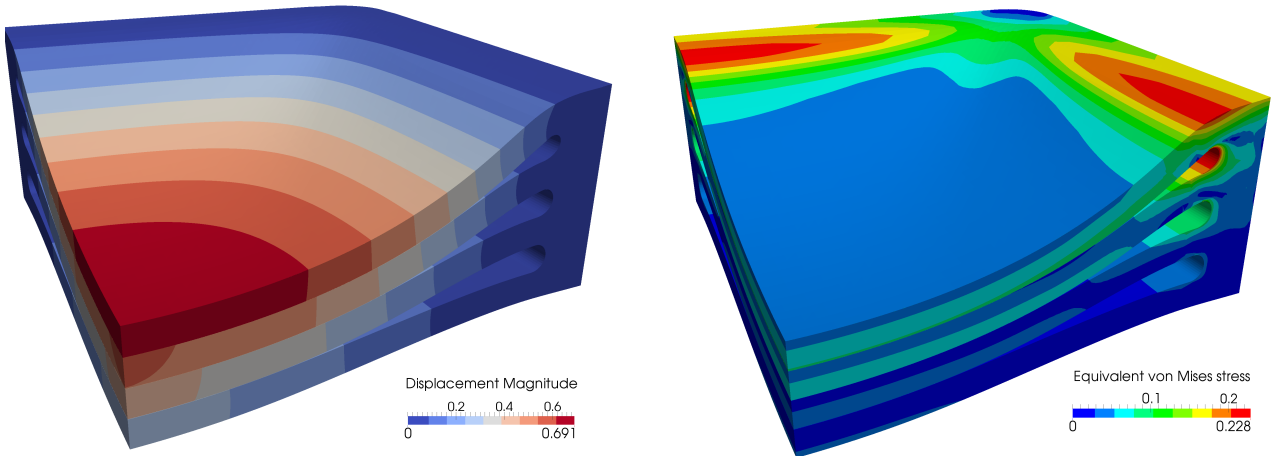
of the ANSYS solution, both results show the same characteristic stress distribution. Even if the contact formulation in our implementation is far from being optimized, the computational time is in the same order as the one for the ANSYS simulation. The evolution of the solution is depicted in Figure 7.28. The results show that the current method is able to reproduce the self-contact as well as the complex state of deformation in the elastic buffer.



(a) $t = 1/3$



(b) $t = 2/3$



(c) $t = 1.0$

Figure 7.28: Load history depicting the displacement magnitude (left) and equivalent von Mises stress (right) for different load steps.

Chapter 8

Summary, Conclusion and Outlook

The present work investigated possibilities to numerical model frictionless contact problems within the framework of the FCM, a fictitious domain method based on high-order finite element methods. To this end, two rather different contact formulations have been developed.

Weakly enforced contact constraints

The first contact formulation can be considered as an extension of the concept of weakly enforced Dirichlet conditions. Therefore, normal constraints of inequality type, based on a penalty regularization, were applied on embedded, high-order interfaces. Multi-level *hp*-refinements have been employed to improve the numerical solution locally, and to reduce oscillations stemming from singularities at the end of the contact zone.
A prerequisite to facilitate simulation in an automated manner is to algorithmically recover embedded surfaces. Therefore, an approach was presented to recover surfaces of geometric models with high-order accuracy. For this approach to work, these models only need to provide simple inside-outside tests. The basic ingredient is an extended Marching Cubes algorithm, which performs ray intersections using an inexact line search. These ray intersections are also used to obtain additional control points to create polynomial Lagrange segments forming the final surface. The high-order representation not only allows for more accuracy but also reduces the number of segments necessary to describe the geometry. A desirable extension would be to remove triangles with very small angles, which get created whenever the intersection between the geometry and the recovery grid is small. The feature-preserving reduction algorithm by Garland et al. [42] would be a suitable candidate. Such a reduction approach could also be used to account for kinks in the geometry. In this case, the surface could be recovered first with a high-resolution followed by a feature-preserving reduction. Alternatively, the surfaces could also be recovered using more sophisticated algorithms which directly reproduce kinks. Possible algorithms include Dual Contouring [59] and Dual Marching Cubes [107].
The proposed contact formulation was investigated by numerical experiments including a two-dimensional, unilateral Hertzian contact problem [51], a two-dimensional “ironing” problem and a three-dimensional bilateral setup. The investigations showed that the proposed high-order formulation provides highly-accurate results for the Hertz problem in an embedded domain setting. Furthermore, the “ironing” example helped to show that the formulation can be applied to problems involving large sliding and large deformations. A simple penetration based refinement indicator allowed for dynamic, local refinements during the load history, which led

to significant improvements in the contact tractions. Finally, the three-dimensional example showed that the new formulation can be applied directly to practical engineering problems.

The current formulation is based on a penalty regularization of the Karush-Kuhn-Tucker conditions for frictionless contact. Hence, future work should focus to incorporate more sophisticated constraint formulations such as parameter-free Nitsche formulations (e.g. [19, 22, 110, 46]) or dual mortar formulations [93, 94] and the incorporation of frictional effects.

Contact material formulation

The second contact formulation is based on a contact material formulation [134, 16]. Contact constraints are enforced on the domain level by embedding the physical structure in a contact material, which prevents adjacent contact boundaries from interpenetrating. The proposed material is based on the hyperelastic formulation by Hencky, which uses a logarithmic strain measure and regularizes the Karush-Kuhn-Tucker conditions by scaling the material parameters. To avoid penetration of contact interfaces, a line search method was utilized that evaluated the determinant of the gradient of deformation inside the contact domain. The applicability of the method was demonstrated by means of two- and three-dimensional examples, including a Hertzian contact example. The application of high ansatz orders led to results comparable to classical approaches in the physical domain for only a small number of coarse elements. However, applying the same ansatz order in the contact and the physical domain imposes an additional and unnecessary numerical burden for the Newton-Raphson procedure, inhibiting a reduction of the error level to lower thresholds. These difficulties can be overcome by either choosing a lower penalization of the contact material or by deactivating higher-order modes in the contact domain only. As long as the contact interface is still resolved with high-order shape functions, no reduction in accuracy was observed. It also lies in line with the usual implementation of hierarchical ansatz spaces and has the added advantage that it leads to fewer degrees of freedom. Our formulation tracks the contact status at the integration points by evaluating the principal stretches. Hence, the influence of the number of integration points on the resulting minimum gap was investigated. The results showed that for discretizations involving large elements with strongly curved boundaries, penetrations can occur if not enough integration points are present in the contact domain. This is an effect, however, common for most contact formulations.

The utilization of a contact material can be classified as a kind of barrier method. These methods have the property that penetrations are avoided at all times. Unfortunately, they also leave small gaps where, ideally, the contact interfaces should touch. A parameter study served to assess the influence of the contact stiffness on the final gap. The results showed that final gaps can be reduced to very small magnitudes by choosing small values of contact stiffness. Moreover, the contact stiffness turned out to be the most influential parameter in comparison to the ansatz order and the number of integration points. As shown by the numerical examples, the contact material formulation has the advantage that explicit geometric searches on the global and local level can be avoided. Applying high-order finite elements allows structures to be discretized for a desired approximation accuracy with a much smaller number of degrees of freedom compared to the h-version of the FEM. A possible extension to the proposed formulation is the incorporation of friction. This will call for a more sophisticated and most likely anisotropic material formulation.

Comparison of both contact formulations

The presented contact formulations were both able to model frictionless contact phenomena in the small and large strain regimes.
The weak constraint formulation is a natural extension of weak Dirichlet and coupling constraints. Therefore, it is strongly related to segment-to-segment or mortar-type contact formulations, which are well understood and state of the art. The contact material formulation is a more uncommon and novel approach to enforce contact constraints. It is motivated by the fictitious material employed by the FCM. The main benefit of the contact material is that no explicit contact searches are necessary. The search effort is shifted to the evaluation of a volumetric integral in the contact domain and the solution of an Eigenvalue problem on each contact integration point. The latter can be avoided by using a material formulation, which is not based on principal stretches.
The material contact formulation handles self-contact naturally. While this is not the case for the weak constraint formulation, it is also not a problem. When the contact interfaces are assigned correctly or a two-pass approach is employed [80], the weak constraint formulation is also able to handle self-contact problems.
In theory, the non-linear problem stemming from the contact material is smoother than that of the weak constraint formulation since no inequality constraints are present. In practice, however, care must be taken that the solution is feasible at all times. Therefore, the incremental steps of the Newton-Raphson method have to be limited to avoid any interpenetration. This either leads to very small load steps or introduces *numerical* nonlinearities stemming from safeguard algorithms.
Extending both formulations to friction or other constitutive interface models poses different challenges. There are many formulations available, which can be directly applied to the weak constraint formulation. However, their efficient implementation in conjunction with the local refinement and the independent description of the contact interface is far from trivial. Incorporating constitutive *interface* models into the (volumetric) contact material formulations calls for new developments altogether.
Finally, unlike the contact material, the weak constraint formulation allows for the simulation of problems involving large sliding.
Considering these points, the weak constraint formulation seems to be applicable for a wider range of problems. Since it is based on state of the art technology, it provides easier extension points and would probably meet a higher acceptance in the engineering community. The contact material formulation, on the other hand, is more innovative and has several open academic questions, which still makes it interesting for further research.

Appendix A

High-order geometric segments based on Lagrange polynomials

Geometric segments based on Lagrange polynomials, in the following denoted as *Lagrange segments*, allow for simple interpolation of point data. In contrast, interpolations based on Hermite or Bernstein polynomials lead to a system of equations, which has to be solved. Therefore, Lagrange segments can also be updated easily, which makes them very attractive for boundary descriptions in contact simulations.
A common drawback of Lagrange interpolation is the increase of the Lebesgue constant for increasing polynomial order, which can result high interpolation errors (Runge's phenomenon) [29]. However, in practical applications the interpolation order is generally restricted to moderate degrees of $p = 2..5$, so that this problem is easily avoided. Also, if higher orders are used for interpolation, the error can be largely reduced by using points from Gauss-Lobatto or Chen-Babuška [21] abscissae instead of equidistant control points.

A.1 Lagrange curves

This section recalls the basic formulation of Lagrange curves [1]. They are especially useful for describing boundaries in a two-dimensional setting.

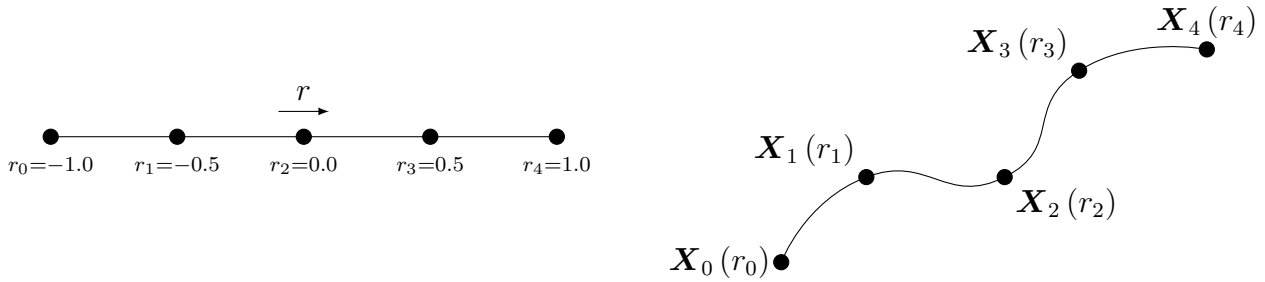
A.1.1 Parametrization

A curve interpolating $n+1$ control points $\boldsymbol{X}_i$ can be described parametrically as

$$\boldsymbol{X}(r) = \sum_{i=0}^{n} \boldsymbol{X}_i(r_i)\, L_i(r) \ . \tag{A.1}$$

Here, r_i denotes the index parameter at which the curve interpolates the control point $\boldsymbol{X_i}$ (Figure A.1) and $L_i(r)$ are the one-dimensional Lagrange polynomials defined by

$$L_i(r) = \prod_{\substack{m=0 \\ m \neq i}}^{n} \frac{r - r_m}{r_i - r_m} \ . \tag{A.2}$$



(a) Index space of Lagrange curve **(b)** Lagrange curve in real space

Figure A.1: Lagrange curve of order $n = 4$ with equidistant control points.

In this classical form the evaluation of a Lagrange interpolation has a complexity of $\mathcal{O}(n^2)$. A more efficient representation is given by the so-called barycentric form [14]

$$\boldsymbol{X}(r) = \frac{\sum_{i=0}^{n} \frac{w_i}{r - r_i} \boldsymbol{X}_i}{\sum_{i=0}^{n} \frac{w_i}{r - r_i}} , \tag{A.3}$$

which only has a complexity of $\mathcal{O}(n)$. Here, the barycentric weights w_i can be computed in advance and are defined as

$$w_i = \prod_{\substack{m=0 \\ m \neq i}}^{n} \frac{1}{r_i - r_m} . \tag{A.4}$$

A.1.2 Jacobian and Metric

The Jacobian is an essential quantity for the numerical integration as well as for the computation of the normal vector. For curves, the Jacobian represents a vector, which follows the parametrization tangentially

$$\boldsymbol{J}(r) = \begin{bmatrix} \frac{\partial X}{\partial r} & \frac{\partial Y}{\partial r} & \frac{\partial Z}{\partial r} \end{bmatrix} = \frac{\partial \boldsymbol{X}}{\partial r}(r) . \tag{A.5}$$

The derivatives can be computed as

$$\frac{\partial \boldsymbol{X}}{\partial r}(r) = \sum_{i=0}^{n} \boldsymbol{X}_i(r_i) \frac{\partial L_i(r)}{\partial r} , \tag{A.6}$$

with the derivative of the Lagrange polynomials given as

$$\frac{\partial L_i(r)}{\partial r} = L_i(r) \sum_{\substack{m=0 \\ m \neq i}}^{n} \frac{1}{r - r_m} . \tag{A.7}$$

The metric needed for numerical integration is defined as

$$\det \boldsymbol{J}^* (r) = \sqrt{\boldsymbol{J} (r) \cdot \boldsymbol{J} (r)} \ . \tag{A.8}$$

It defines the ratio between the length of an infinitesimal curve segment in the parametric space and its representation in global space.

A.1.3 Normal Vector

In the three-dimensional space the normal vector of a curve is not uniquely defined. For two-dimensional problems however, it can be computed by evaluating the normalized cross product of the Jacobian (tangent) vector and the out-of-plane base vector $\boldsymbol{E}_3$

$$\boldsymbol{n} = \frac{\boldsymbol{J} (r) \times \boldsymbol{E}_3}{\| \boldsymbol{J} (r) \times \boldsymbol{E}_3 \|} \ . \tag{A.9}$$

A.2 Barycentric Lagrange triangles

This section recalls the formulation of equidistant Lagrange triangles based on barycentric (*area*) coordinates. High-order curve segments with triangular topology can be directly applied to a variety of surface recovery algorithms such as Marching Cubes [73].

A.2.1 Barycentric coordinates

Any point $\boldsymbol{P}$ within a planar, non-degenerate triangle spanned by the points $\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3$ can be expressed using barycentric coordinates $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)$ (Figure A.2)

$$\boldsymbol{P} = \sum_{i=1}^{3} \tau_i \boldsymbol{T}_i \ . \tag{A.10}$$

Barycentric coordinates have the property that they sum up to one

$$|\boldsymbol{\tau}| = \sum_{i=1}^{3} \tau_i = 1 \ . \tag{A.11}$$

The interpolation can also be reverted to compute barycentric coordinates from Cartesian coordinates. In the two-dimensional case, where the triangle is spanned by points $\boldsymbol{T}_i = (x_i, y_i)$, the transformation reads

$$\tau_1 = \frac{(y_2 - y_3)(x - x_3) + (x_3 - x_2)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)} \tag{A.12a}$$

$$\tau_2 = \frac{(y_3 - y_1)(x - x_3) + (x_1 - x_3)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)} \tag{A.12b}$$

$$\tau_3 = 1 - \tau_1 - \tau_2 \ . \tag{A.12c}$$

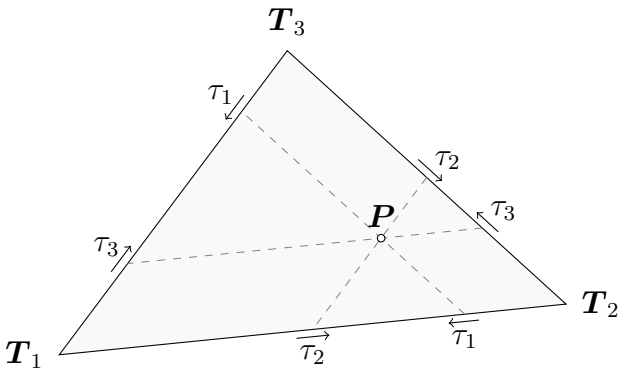


Figure A.2: Barycentric interpolation on a flat triangle. The dashed lines indicate constant values of the components τ_i.

In the three-dimensional case the barycentric coordinates can be obtained using the projection method presented by Heidrich [49]. Therefore, the following vectors are defined

$$\boldsymbol{u} = \boldsymbol{T}_2 - \boldsymbol{T}_1 \tag{A.13a}$$
$$\boldsymbol{v} = \boldsymbol{T}_3 - \boldsymbol{T}_1 \tag{A.13b}$$
$$\boldsymbol{w} = \boldsymbol{r} - \boldsymbol{T}_1 \tag{A.13c}$$
$$\boldsymbol{n} = \boldsymbol{u} \times \boldsymbol{v} \ , \tag{A.13d}$$

where $\boldsymbol{r}$ is the Cartesian coordinate vector to be projected or mapped into the barycentric space of the triangle. The barycentric components then follow as

$$\tau_3 = \frac{(\boldsymbol{u} \times \boldsymbol{w}) \cdot \boldsymbol{n}}{\boldsymbol{n} \cdot \boldsymbol{n}} \tag{A.14a}$$
$$\tau_2 = \frac{(\boldsymbol{w} \times \boldsymbol{v}) \cdot \boldsymbol{n}}{\boldsymbol{n} \cdot \boldsymbol{n}} \tag{A.14b}$$
$$\tau_1 = 1 - \tau_2 - \tau_3 \ . \tag{A.14c}$$

A.2.2 Triangle parametrization

Following [32], the parametrization $\boldsymbol{X} = (X, Y, Z)$ for a Lagrange triangle of order n based on equidistant barycentric control points $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)$ is defined as

$$\boldsymbol{X}(\boldsymbol{\tau}) = \sum_{\boldsymbol{\lambda}} \boldsymbol{X}_{\boldsymbol{\lambda}} L^n_{\boldsymbol{\lambda}}(\boldsymbol{\tau}) \ . \tag{A.15}$$

The tuple $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ with $\lambda_i \in \{0, 1, \ldots, n\}$ labels the control points of the triangle (see Figure A.3) and the following relations hold

$$|\boldsymbol{\lambda}| = \sum_{i=1}^{3} \lambda_i = n \tag{A.16}$$
$$\boldsymbol{\lambda}! = \lambda_1! \cdot \lambda_2! \cdot \lambda_3! \ . \tag{A.17}$$

The point data to be interpolated is denoted by $\boldsymbol{X}_{\boldsymbol{\lambda}}$ and $L^n_{\boldsymbol{\lambda}}$ represents equidistant, barycentric Lagrange polynomials defined as

$$L^n_{\boldsymbol{\lambda}} = \frac{n^n}{\boldsymbol{\lambda}!} \prod_{a=0}^{\lambda_1-1} \prod_{b=0}^{\lambda_2-1} \prod_{c=0}^{\lambda_3-1} \left(\tau_1 - \frac{a}{n}\right) \left(\tau_2 - \frac{b}{n}\right) \left(\tau_3 - \frac{c}{n}\right) . \tag{A.18}$$

A.2.3 Mappings

There are several options to perform numerical integration on a triangle. One possibility is to map a tensor product of one-dimensional quadrature rules onto the triangle [9]. Therefore the mapping from a Cartesian quadrilateral index space to the barycentric index space is needed. This can be achieved by mapping the index quadrilateral (Figure A.4a) onto a collapsed quadrilateral as depicted in Figure A.4b

$$a\,(r,s) = N_i a_i \qquad \text{with} \quad a_i = \{0,1,0,0\} \tag{A.19a}$$

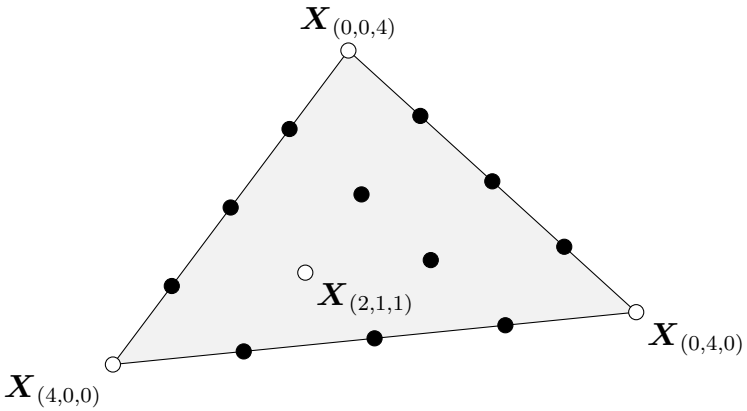
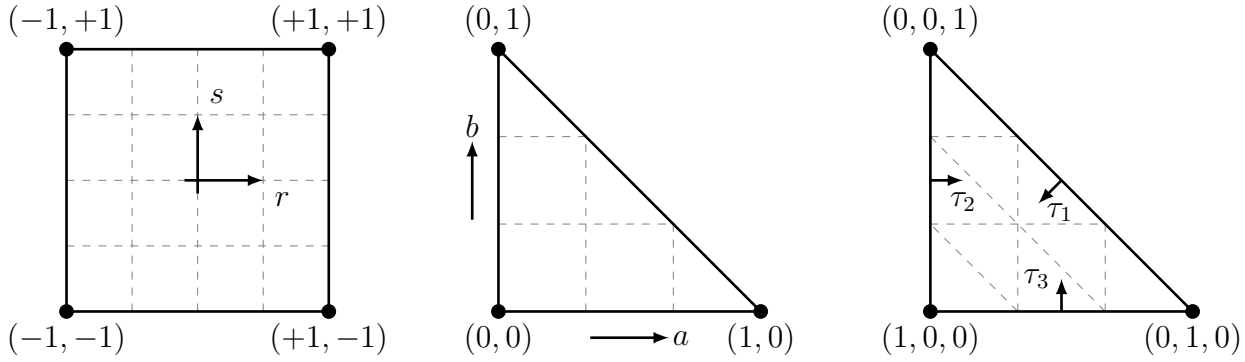


Figure A.3: Labeling of control points for a triangle of order $n = 4$.



(a) Cartesian index space for quadrilateral

(b) Collapsed Cartesian index space for triangle

(c) Barycentric index space for triangle

Figure A.4: Different index spaces for the triangle.

$$b\left(r,s\right) = N_i b_i \qquad \text{with} \quad b_i = \left\{0,0,1,1\right\} \ , \tag{A.19b}$$

where N_i are standard bilinear shape functions and a_i and b_i are the corner coordinates of the collapsed quadrilateral. The resulting mappings then read

$$a\left(r,s\right) = \frac{1}{4}\left(1+r\right)\left(1-s\right) \qquad b\left(r,s\right) = \frac{1}{2}\left(1+s\right) \ . \tag{A.20}$$

The Cartesian coordinates of the resulting triangle (a,b) coincide with the barycentric coordinates τ_2 and τ_3.

$$\tau_1\left(a,b\right) = 1-a-b \qquad \tau_2\left(a,b\right) = a \qquad \tau_3\left(a,b\right) = b \tag{A.21}$$

Another option is to employ quadrature rules, derived specifically for triangles. These can be given in full barycentric coordinates τ_i or for the right triangle in terms of a and b (i.e. τ_2 and τ_3) [117, 25].

A.2.4 Jacobian

Based on the index spaces introduced in the previous section the Jacobian can be formulated with respect to the index space of the standard quadrilateral

$$\boldsymbol{J}\left(r,s\right) = \begin{bmatrix} \frac{\partial X}{\partial r} & \frac{\partial Y}{\partial r} & \frac{\partial Z}{\partial r} \\ \frac{\partial X}{\partial s} & \frac{\partial Y}{\partial s} & \frac{\partial Z}{\partial s} \end{bmatrix} = \begin{bmatrix} \frac{\partial X}{\partial \tau_i}\frac{\tau_i}{\partial r} & \frac{\partial Y}{\partial \tau_i}\frac{\tau_i}{\partial r} & \frac{\partial Z}{\partial \tau_i}\frac{\tau_i}{\partial r} \\ \frac{\partial X}{\partial \tau_i}\frac{\tau_i}{\partial s} & \frac{\partial Y}{\partial \tau_i}\frac{\tau_i}{\partial s} & \frac{\partial Z}{\partial \tau_i}\frac{\tau_i}{\partial s} \end{bmatrix} \tag{A.22}$$

or with respect to the quadrilateral collapsed to a right triangle

$$\boldsymbol{J}\left(a,b\right) = \begin{bmatrix} \frac{\partial X}{\partial a} & \frac{\partial Y}{\partial a} & \frac{\partial Z}{\partial a} \\ \frac{\partial X}{\partial b} & \frac{\partial Y}{\partial b} & \frac{\partial Z}{\partial b} \end{bmatrix} = \begin{bmatrix} \frac{\partial X}{\partial \tau_i}\frac{\tau_i}{\partial a} & \frac{\partial Y}{\partial \tau_i}\frac{\tau_i}{\partial a} & \frac{\partial Z}{\partial \tau_i}\frac{\tau_i}{\partial a} \\ \frac{\partial X}{\partial \tau_i}\frac{\tau_i}{\partial b} & \frac{\partial Y}{\partial \tau_i}\frac{\tau_i}{\partial b} & \frac{\partial Z}{\partial \tau_i}\frac{\tau_i}{\partial b} \end{bmatrix} \ . \tag{A.23}$$

Here repeated indices denote summation (Einstein notation). The partial derivatives of the barycentric coordinates with respect to the Cartesian coordinates (a,b) and (r,s) read

$$\frac{\tau_1}{\partial r} = -\frac{1}{4}\left(1-s\right) \qquad \frac{\tau_2}{\partial r} = \frac{1}{4}\left(1-s\right) \qquad \frac{\tau_3}{\partial r} = 0 \tag{A.24a}$$

$$\frac{\tau_1}{\partial s} = \frac{1}{4}\left(1+r\right) - \frac{1}{2} \qquad \frac{\tau_2}{\partial s} = -\frac{1}{4}\left(1+r\right) \qquad \frac{\tau_3}{\partial s} = \frac{1}{2} \tag{A.24b}$$

and

$$\frac{\tau_1}{\partial a} = -1 \qquad \frac{\tau_2}{\partial a} = 1 \qquad \frac{\tau_3}{\partial a} = 0 \tag{A.24c}$$

$$\frac{\tau_1}{\partial b} = -1 \qquad \frac{\tau_2}{\partial b} = 0 \qquad \frac{\tau_3}{\partial b} = 1 \ , \tag{A.24d}$$

respectively. The partial derivatives of the (undeformed) global space with respect to the barycentric coordinates $\frac{\partial X}{\partial \tau_i}$,$\frac{\partial Z}{\partial \tau_i}$ and $\frac{\partial Z}{\partial \tau_i}$ can be computed using Equation A.15

$$\frac{\partial \phi}{\partial \tau_i}\left(\boldsymbol{\tau}\right) = \sum_{\boldsymbol{\lambda}} \phi_{\boldsymbol{\lambda}} \frac{\partial L^n_{\boldsymbol{\lambda}}}{\partial \tau_i}\left(\boldsymbol{\tau}\right) \ , \tag{A.25}$$

where $\phi \in \{X, Y, Z\}$ is a scalar quantity to be interpolated and $\phi_{\boldsymbol{\lambda}}$ are discrete values at the control points of the triangle. The derivatives of the equidistant, barycentric Lagrange polynomials can then be computed as

$$\frac{\partial L^n_{\boldsymbol{\lambda}}}{\partial \tau_1} = \frac{n^n}{\boldsymbol{\lambda}!} \frac{\partial P\left(\tau_1, \lambda_1, n\right)}{\partial \tau_1} \cdot P\left(\tau_2, \lambda_2, n\right) \cdot P\left(\tau_3, \lambda_3, n\right) \tag{A.26a}$$

$$\frac{\partial L^n_{\boldsymbol{\lambda}}}{\partial \tau_2} = \frac{n^n}{\boldsymbol{\lambda}!} P\left(\tau_1, \lambda_1, n\right) \cdot \frac{\partial P\left(\tau_2, \lambda_2, n\right)}{\partial \tau_2} \cdot P\left(\tau_3, \lambda_3, n\right) \tag{A.26b}$$

$$\frac{\partial L^n_{\boldsymbol{\lambda}}}{\partial \tau_3} = \frac{n^n}{\boldsymbol{\lambda}!} P\left(\tau_1, \lambda_1, n\right) \cdot P\left(\tau_2, \lambda_2, n\right) \cdot \frac{\partial P\left(\tau_3, \lambda_3, n\right)}{\partial \tau_3} \ , \tag{A.26c}$$

with

$$P\left(\tau_i, \lambda_i, n\right) = \prod_{a_i=0}^{\lambda_i - 1} \tau_i - \frac{a_i}{n} \tag{A.27a}$$

$$\frac{\partial P\left(\tau_i, \lambda_i, n\right)}{\partial \tau_i} = P\left(\tau_i, \lambda_i, n\right) \cdot \sum_{a_i=0}^{\lambda_i - 1} \frac{1}{\tau_i - \dfrac{a_i}{n}} \ . \tag{A.27b}$$

A.2.5 Metric

The area mapping (metric) needed for numerical integration is obtained from the metric tensor $\boldsymbol{a}$ as

$$\det \boldsymbol{J} = \sqrt{\det \boldsymbol{a}} \ . \tag{A.28}$$

The metric tensor is defined as

$$\boldsymbol{a} = \begin{bmatrix} \boldsymbol{J}_1 \cdot \boldsymbol{J}_1 & \boldsymbol{J}_1 \cdot \boldsymbol{J}_2 \\ \boldsymbol{J}_2 \cdot \boldsymbol{J}_1 & \boldsymbol{J}_2 \cdot \boldsymbol{J}_2 \end{bmatrix} \ , \tag{A.29}$$

where $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ are the two row vectors of the Jacobian in Equations (A.22) or (A.23).

A.2.6 Normal vector

The rows of the Jacobian represent two tangential vectors at any given point (r, s) (Figure A.4). Thus, the normal vector at an index position (r, s) can be computed from the cross product between the two row vectors $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ of the Jacobian in Equations (A.22) or (A.23)

$$\boldsymbol{n}\left(r, s\right) = \frac{\boldsymbol{J}_1 \times \boldsymbol{J}_2}{\left|\boldsymbol{J}_1 \times \boldsymbol{J}_2\right|} \ . \tag{A.30}$$

and normalized by its Euclidean length.

Appendix B

Closed form solutions for closest point projections

B.1 Projection onto line

The vector $\boldsymbol{v}_1$ between $\boldsymbol{x}_\mathrm{S}$ and the projected point $\boldsymbol{x}_\mathrm{M}$ has to be orthogonal to the vector spanning the line segment (see Figure B.1). Thus, the following relation holds

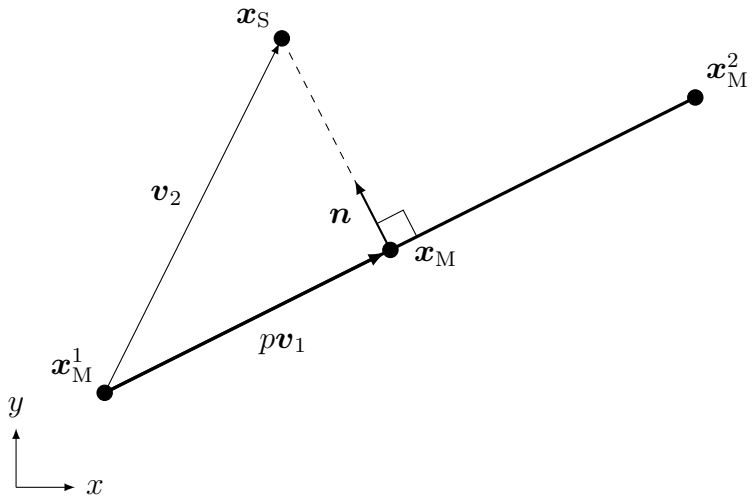


Figure B.1: Projection onto line.

$$\boldsymbol{v}_1 = \boldsymbol{x}^2_\mathrm{M} - \boldsymbol{x}^1_\mathrm{M} \tag{B.1}$$

$$\boldsymbol{v}_2 = \boldsymbol{x}_\mathrm{S} - \boldsymbol{x}^1_\mathrm{M} \tag{B.2}$$

$$\boldsymbol{v}_1 \cdot (\boldsymbol{v}_2 - p\boldsymbol{v}_1) \stackrel{!}{=} 0, \quad p = [0,1] \ , \tag{B.3}$$

from which the linear interpolation parameter p can be obtained as

$$p = \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{\boldsymbol{v}_1 \cdot \boldsymbol{v}_1} \ . \tag{B.4}$$

The projected point then follows as

$$\boldsymbol{x}_{\mathrm{M}} = (1-p)\,\boldsymbol{x}_{\mathrm{M}}^{1} + p\boldsymbol{x}_{\mathrm{M}}^{2}\ . \tag{B.5}$$

B.2 Projection onto triangle

A closed solution for the CPP on linear triangles has been proposed by Heidrich [49]. This formulation obtains the projected point $\boldsymbol{x}_{\mathrm{M}}$ in terms of local barycentric coordinates (see Appendix A). Consider therefore a linear triangle spanned by three points $\boldsymbol{x}_{\mathrm{M}}^{1}, \boldsymbol{x}_{\mathrm{M}}^{2}, \boldsymbol{x}_{\mathrm{M}}^{3}$ and define the following vectors (see Figure B.2)

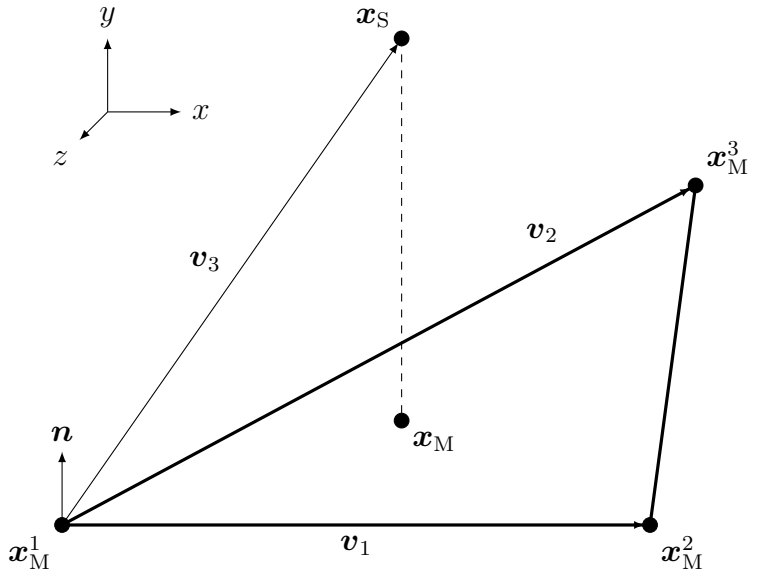


Figure B.2: Closest point projection onto triangle.

$$\boldsymbol{v}_1 = \boldsymbol{x}_{\mathrm{M}}^{2} - \boldsymbol{x}_{\mathrm{M}}^{1} \tag{B.6}$$
$$\boldsymbol{v}_2 = \boldsymbol{x}_{\mathrm{M}}^{3} - \boldsymbol{x}_{\mathrm{M}}^{1} \tag{B.7}$$
$$\boldsymbol{v}_3 = \boldsymbol{x}_{\mathrm{S}} - \boldsymbol{x}_{\mathrm{M}}^{1} \tag{B.8}$$
$$\boldsymbol{n} = \boldsymbol{v}_1 \times \boldsymbol{v}_2\ . \tag{B.9}$$

The local barycentric coordinates $\boldsymbol{\tau}_{\mathrm{M}}$ of the projected point then follow as

$$\tau_{\mathrm{M}}^{2} = \frac{(\boldsymbol{v}_1 \times \boldsymbol{v}_3) \times \boldsymbol{n}}{\boldsymbol{n} \times \boldsymbol{n}} \tag{B.10}$$
$$\tau_{\mathrm{M}}^{3} = \frac{(\boldsymbol{v}_2 \times \boldsymbol{v}_3) \times \boldsymbol{n}}{\boldsymbol{n} \times \boldsymbol{n}} \tag{B.11}$$
$$\tau_{\mathrm{M}}^{1} = 1.0 - \tau_{\mathrm{M}}^{2} - \tau_{\mathrm{M}}^{3}\ . \tag{B.12}$$

Using the barycentric coordinates and the triangle points, the projected point in global coordinates follows as

$$\boldsymbol{x}_{\mathrm{M}} = \sum_{i=1}^{3} \boldsymbol{x}_{\mathrm{M}}^{i} \cdot \tau_{\mathrm{M}}^{i} \tag{B.13}$$

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Dover books on mathematics. Dover, New York, NY, ninth dover printing edition, 2013.

[2] A. Andreykiv, F. van Keulen, D. J. Rixen, and E. Valstar. A level-set-based large sliding contact algorithm for easy analysis of implant positioning. *International Journal for Numerical Methods in Engineering*, 89(10):1317–1336, March 2012.

[3] ANSYS, Inc. *ANSYS Release 14.0, Help System, Element Reference*. 2011.

[4] I. Babuška and B. Guo. The h-p Version of the Finite Element Method for Domains with Curved Boundaries. *SIAM Journal on Numerical Analysis*, 25(4):837–861, August 1988.

[5] I. Babuška and J. M. Melenk. The Partition of Unity Method. *International Journal for Numerical Methods in Engineering*, 40(4):727–758, February 1997.

[6] I. Babuška and M. Suri. The p and h-p Versions of the Finite Element Method, Basic Principles and Properties. *SIAM Review*, 36(4):578–632, December 1994.

[7] I. Babuska, B. Szabo, and I. Katz. The p-Version of the Finite Element Method. *SIAM Journal on Numerical Analysis*, 18(3):515–545, June 1981.

[8] I. Babuška and W. C. Rheinboldt. Analysis of optimal finite-element meshes in R^1. Technical Report BN-869, University of Maryland, Institute for physical science and technology, March 1978.

[9] K. J. Bathe. *Finite Element Procedures*. Prentice Hall, New Jersey, 2007.

[10] M. S. Bazaraa. *Nonlinear Programming: Theory and Algorithms*. Wiley-Interscience, Hoboken, N.J, 3rd edition, 2006.

[11] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5–8):229–263, January 2010.

[12] T. Belytschko. *Nonlinear Finite Elements for Continua and Structures*. Wiley, Chichester, New York, 2000.

[13] T. Belytschko, R. Gracie, and G. Ventura. A review of extended/generalized finite element methods for material modeling. *Modelling and Simulation in Materials Science and Engineering*, 17(4):1–24, 2009.

[14] J.-P. Berrut and L. N. Trefethen. Barycentric Lagrange Interpolation. *SIAM Review*, 46(3):501–517, January 2004.

[15] T. Bog, N. Zander, S. Kollmannsberger, and E. Rank. Normal contact with high order finite elements and a fictitious contact material. *Computers & Mathematics with Applications*, 70(7):1370–1390, October 2015.

[16] T. Bog, N. Zander, S. Kollmannsberger, and E. Rank. Weak imposition of frictionless contact constraints on automatically recovered high-order, embedded interfaces using the finite cell method. *Computational Mechanics*, submitted, 2017.

[17] J. Bonet and R. D. Wood. *Nonlinear Continuum Mechanics For Finite Element Analysis*. Cambridge University Press, Cambridge, 2nd edition, 2008.

[18] P. Bourke. Polygonising a scalar field (Marching Cubes). `http://paulbourke.net/geometry/polygonise/`, 1994.

[19] E. Burman. A penalty free non-symmetric Nitsche type method for the weak imposition of boundary conditions. arXiv e-print 1106.5612, June 2011.

[20] E. Burman, S. Claus, P. Hansbo, M. G. Larson, and A. Massing. CutFEM: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501, November 2015.

[21] Q. Chen and I. Babuška. Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle. *Computer Methods in Applied Mechanics and Engineering*, 128(3–4):405–417, December 1995.

[22] F. Chouly, P. Hild, and Y. Renard. Symmetric and non-symmetric variants of Nitsche's method for contact problems in elasticity: Theory and numerical experiments. Technical Report hal-00776619, 2013.

[23] J. Cottrell, T. J. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley and Sons, New York, 2009.

[24] S. Duczek and U. Gabbert. Efficient integration method for fictitious domain approaches. *Computational Mechanics*, August 2015.

[25] D. A. Dunavant. High degree efficient symmetrical Gaussian quadrature rules for the triangle. *International Journal for Numerical Methods in Engineering*, 21(6):1129–1148, June 1985.

[26] A. Düster, J. Parvizian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197(45–48):3768–3782, August 2008.

[27] A. Einstein. Die Grundlage der allgemeinen Relativitätstheorie. *Annalen der Physik*, 354(7):769–822, 1916.

[28] M. Elhaddad, N. Zander, T. Bog, S. Kollmannsberger, M. Ruess, and E. Rank. Multi-level hp-adaptivity for embedded interface problems with application in biomechanics. in preparation, 2017.

[29] J. F. Epperson. On the Runge Example. *The American Mathematical Monthly*, 94(4):329, April 1987.

[30] C. Ericson. *Real-Time Collision Detection*. Morgan Kaufmann series in interactive 3D technology. Elsevier, Amsterdam, Boston, 2005.

[31] M. Fabre, J. Pousin, and Y. Renard. A fictitious domain method for frictionless contact problems in elasticity using Nitsche's method. *SMAI Journal of Computational Mathematics*, 2:19–50, 2016.

[32] G. Farin. Triangular Bernstein-Bézier patches. *Computer Aided Geometric Design*, 3(2):83–127, August 1986.

[33] K. A. Fischer and P. Wriggers. Frictionless 2D Contact formulations for finite deformations based on the mortar method. *Computational Mechanics*, 36(3):226–244, August 2005.

[34] K. A. Fischer and P. Wriggers. Mortar based frictional contact formulation for higher order interpolations using the moving friction cone. *Computer Methods in Applied Mechanics and Engineering*, 195(37–40):5020–5036, July 2006.

[35] A. Francavilla and O. C. Zienkiewicz. A note on numerical computation of elastic contact problems. *International Journal for Numerical Methods in Engineering*, 9(4):913–924, January 1975.

[36] D. Franke. *Investigation of Mechanical Contact Problems with High-Order Finite Element Methods*. PhD thesis, Technische Universität München, 2011.

[37] D. C. Franke, A. Düster, V. Nübel, and E. Rank. A comparison of the h-, p-, hp-, and rp-version of the FEM for the solution of the 2D Hertzian contact problem. *Computational Mechanics*, 45(5):513–522, January 2010.

[38] D. C. Franke, A. Düster, and E. Rank. The p-version of the FEM for computational contact mechanics. *PAMM*, 8(1):10271–10272, December 2008.

[39] T.-P. Fries and S. Omerović. Higher-order accurate integration of implicit geometries. *International Journal for Numerical Methods in Engineering*, 106(5):323–371, January 2015.

[40] F. Frischmann. *Topological and Geometric Healing on Solid Models*. Master's thesis, Technische Universität München, Chair for Computation in Engineering, November 2011.

[41] L. A. Galin and G. M. L. Gladwell. *Contact Problems: The Legacy of L. A. Galin*. Number v. 155 in Solid mechanics and its applications. Springer, Dordrecht, Netherlands, 2008.

[42] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 209–216. ACM Press/Addison-Wesley Publishing Co., 1997.

[43] R. Glowinski, T.-W. Pan, and J. Periaux. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering*, 111(3–4):283–303, 1994.

[44] W. J. Gordon and C. A. Hall. Transfinite element methods: Blending-function interpolation over arbitrary curved element domains. *Numerische Mathematik*, 21(2):109–129, April 1973.

[45] C. E. Green, S. A. Leadbetter, and M. H. Rheinfurth. Dynamic Testing for Shuttle Design Verification. In *NASA Space Shuttle Technol. Conf.*, pages 231–265, San Antonio, Texas, April 1972.

[46] Y. Guo, M. Ruess, and D. Schillinger. A parameter-free variational coupling approach for trimmed isogeometric thin shells. *Computational Mechanics*, 2016.

[47] J. O. Hallquist. *NIKE2D: An Implicit, Finite-Deformation, Finite-Element Code for Analyzing the Static and Dynamic Response of Two-Dimensional Solids.* Lawrence Livermore Laboratory, University of California, Berkeley, 1979.

[48] A. Hansbo, P. Hansbo, and M. G. Larson. A finite element method on composite grids based on Nitsche's method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(03):495–514, 2003.

[49] W. Heidrich. Computing the Barycentric Coordinates of a Projected Point. *Journal of Graphics, GPU, and Game Tools*, 10(3):9–12, January 2005.

[50] U. Heisserer, S. Hartmann, A. Düster, W. Bier, Z. Yosibash, and E. Rank. P-FEM for finite deformation powder compaction. *Computer Methods in Applied Mechanics and Engineering*, 197(6-8):727–740, January 2008.

[51] H. Hertz. Ueber die Berührung fester elastischer Körper. *Journal für die reine und angewandte Mathematik*, 92:156–171, 1882.

[52] K. Höllig, U. Reif, and J. Wipper. Weighted extended B-spline approximation of Dirichlet problems. *SIAM Journal on Numerical Analysis*, 39(2):442–462, 2001.

[53] G. A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering.* Wiley, Chichester, New York, 2000.

[54] J. F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Upper Saddle River, New Jersey, 3rd edition, 2014.

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

[56] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Mineola, NY, 2000.

[57] M. Joulaian and A. Düster. Local enrichment of the finite cell method for problems with material interfaces. *Computational Mechanics*, 52(4):741–762, October 2013.

[58] M. Joulaian, S. Hubrich, and A. Düster. Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Computational Mechanics*, pages 1–21, March 2016.

[59] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. page 339. ACM Press, 2002.

[60] N. Kikuchi and J. T. Oden. *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*. SIAM, Philadelphia, 1988.

[61] G. Kloosterman, R. M. J. van Damme, A. H. van den Boogaard, and J. Huétink. A geometrical-based contact algorithm using a barrier method. *International Journal for Numerical Methods in Engineering*, 51(7):865–882, July 2001.

[62] A. Konyukhov. *Geometrically Exact Theory for Contact Interactions*. PhD thesis, KIT Scientific Publishing, Karlsruhe, 2011.

[63] A. Konyukhov. Geometrically Exact Theory of Contact Interactions—Applications with Various Methods FEM and FCM. *Journal of Applied Mathematics and Physics*, 03(08):1022–1031, 2015.

[64] A. Konyukhov and R. Izi. *Introduction to Computational Contact Mechanics: A Geometrical Approach*. Wiley series in computational mechanics. John Wiley & Sons, Chichester, West Sussex, 2015.

[65] A. Konyukhov, C. Lorenz, and K. Schweizerhof. Various contact approaches for the finite cell method. *Computational Mechanics*, pages 1–21, June 2015.

[66] A. Konyukhov and K. Schweizerhof. Closest point projection in contact mechanics: Existence and uniqueness for different types of surfaces. *PAMM*, 7(1):4040053–4040054, December 2007.

[67] A. Konyukhov and K. Schweizerhof. Incorporation of contact for high-order finite elements in covariant form. *Computer Methods in Applied Mechanics and Engineering*, 198(13-14):1213–1223, March 2009.

[68] L. Kudela, N. Zander, T. Bog, S. Kollmannsberger, and E. Rank. Efficient and accurate numerical quadrature for immersed boundary methods. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):1–22, June 2015.

[69] L. Kudela, N. Zander, S. Kollmannsberger, and E. Rank. Smart octrees: Accurately integrating discontinuous functions in 3D. *Computer Methods in Applied Mechanics and Engineering*, 306:406–426, July 2016.

[70] L. Kuptsov. Einstein rule. `http://www.encyclopediaofmath.org/index.php?title=Einstein_rule&oldid=32064`, 2014.

[71] T. A. Laursen. *Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*. Springer, Berlin, 2002.

[72] C. Lehrenfeld. High order unfitted finite element methods on level set domains using isoparametric mappings. *Computer Methods in Applied Mechanics and Engineering*, 300:716–733, March 2016.

[73] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pages 163–169, New York, NY, 1987. ACM Press.

[74] D. G. Luenberger. *Linear and Nonlinear Programming*. Springer, New, NY, 2nd edition, 2005.

[75] L. E. Malvern. *Introduction to the Mechanics of a Continuous Medium*. Prentice Hall, Englewood Cliffs, June 1977.

[76] M. Mäntylä. *An Introduction to Solid Modeling*. Number 13 in Principles of computer science series. Computer Science Press, Rockville, 1988.

[77] C. Maple. Geometric design and space planning using the marching squares and marching cube algorithms. In *2003 International Conference on Geometric Modeling and Graphics, 2003. Proceedings*, pages 90–95. IEEE, 2003.

[78] F. Massarwi and G. Elber. A B-spline based framework for volumetric object modeling. *Computer-Aided Design*, 78:36–47, September 2016.

[79] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, September 1999.

[80] A. Mongeau. *Large Deformation Two- and Three- Dimensional Contact on Embedded Interfaces Using the Finite Cell Method*. Master's thesis, Technische Universität München, December 2015.

[81] B. Müller, F. Kummer, and M. Oberlack. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *International Journal for Numerical Methods in Engineering*, 96(8):512–528, November 2013.

[82] E. Nadal, J. J. Ródenas, J. Albelda, M. Tur, J. E. Tarancón, and F. J. Fuenmayor. Efficient Finite Element Methodology Based on Cartesian Grids: Application to Structural Shape Optimization. *Abstract and Applied Analysis*, 2013:1–19, 2013.

[83] E. Nadal Soriano. *Cartesian Grid FEM (cgFEM): High Performance h-Adaptive FE Analysis with Efficient Error Control. Application to Structural Shape Optimization*. PhD thesis, 2014.

[84] T. Netz, A. Düster, and S. Hartmann. High-order finite elements compared to low-order mixed element formulations. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 93(2-3):163–176, February 2013.

[85] V. Nübel, A. Düster, and E. Rank. An rp-adaptive finite element method for the deformation theory of plasticity. *Computational Mechanics*, 39(5):557–574, April 2007.

[86] J. T. Oden. Exterior Penalty Methods for Contact Problems in Elasticity. In W. Wunderlich, Erwin Stein, and Klaus Jürgen Bathe, editors, *Nonlinear Finite Element Analysis in Structural Mechanics*, pages 655–665. Springer, Berlin Heidelberg, January 1981.

[87] R. W. Ogden. *Non-Linear Elastic Deformations*. Dover Publications, Mineola, N.Y, 1997.

[88] S. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Number 153 in Applied mathematical sciences. Springer, New York, 2003.

[89] J. Parvizian, A. Düster, and E. Rank. Finite cell method. *Computational Mechanics*, 41(1):121–133, April 2007.

[90] L. Piegl and W. Tiller. *The NURBS Book*. Monographs in Visual Communication. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[91] A. Popp, M. W. Gee, and W. A. Wall. A finite deformation mortar contact formulation using a primal-dual active set strategy. *International Journal for Numerical Methods in Engineering*, 79(11):1354–1391, September 2009.

[92] A. Popp, M. W. Gee, and W. A. Wall. Finite Deformation Contact Based on a 3D Dual Mortar and Semi-Smooth Newton Approach. In F. Pfeiffer, P. Wriggers, G. Zavarise, and P. Wriggers, editors, *Trends in Computational Contact Mechanics*, volume 58, pages 57–77. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[93] A. Popp, M. W. Gee, and W. A. Wall. Mortar Methods for Single- and Multi-Field Applications in Computational Mechanics. In M. M. Resch, X. Wang, W. Bez, E. Focht, and H. Kobayashi, editors, *Sustained Simulation Performance 2012*, pages 133–154. Springer Berlin Heidelberg, 2013.

[94] A. Popp, A. Seitz, M. W. Gee, and W. A. Wall. Improved robustness and consistency of 3D contact algorithms based on a dual mortar approach. *Computer Methods in Applied Mechanics and Engineering*, 264:67–80, September 2013.

[95] M. A. Puso. A 3D mortar method for solid mechanics. *International Journal for Numerical Methods in Engineering*, 59(3):315–336, January 2004.

[96] M. A. Puso and T. A. Laursen. A mortar segment-to-segment contact method for large deformation solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(6–8):601–629, February 2004.

[97] I. Ramière, P. Angot, and M. Belliard. A fictitious domain approach with spread interface for elliptic problems with general boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 196(4-6):766–781, January 2007.

[98] A. C. Ramos, A. M. Aragón, S. Soghrati, P. H. Geubelle, and J.-F. Molinari. A new formulation for imposing Dirichlet boundary conditions on non-matching meshes. *International Journal for Numerical Methods in Engineering*, 103(6):430–444, August 2015.

[99] E. Rank, A. Düster, V. Nübel, K. Preusch, and O. Bruhns. High order finite elements for shells. *Computer Methods in Applied Mechanics and Engineering*, 194(21-24):2494–2512, June 2005.

[100] E. Rank. Adaptive remeshing and h-p domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 101(1–3):299–313, December 1992.

[101] B. D. Reddy. *Introductory Functional Analysis: With Applications to Boundary Value Problems and Finite Elements*. Number 27 in Texts in applied mathematics. Springer, New York, 1998 edition, November 1997.

[102] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank. 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, September 2013.

[103] M. Ruess, D. Schillinger, A. I. Özcan, and E. Rank. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269:46–71, February 2014.

[104] M. Ruess, D. Tal, N. Trabelsi, Z. Yosibash, and E. Rank. The finite cell method for bone simulations: Verification and validation. *Biomechanics and modeling in mechanobiology*, 11(3-4):425–37, March 2012.

[105] J. Sanders. *Stable Embedded Grid Techniques in Computational Mechanics*. PhD thesis, 2010.

[106] S. Schaefer and J. Warren. Dual Contouring: "The Secret Sauce". `https://people.eecs.berkeley.edu/~jrs/meshpapers/SchaeferWarren2.pdf`, 2002.

[107] S. Schaefer and J. Warren. Dual marching cubes: Primal contouring of dual grids. In *Computer Graphics Forum*, volume 24, pages 195–201. Wiley Online Library, 2005.

[108] D. Schillinger, A. Düster, and E. Rank. 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, 2012.

[109] D. Schillinger. *The P- and B-Spline Versions of the Geometrically Nonlinear Finite Cell Method and Hierarchical Refinement Strategies for Adaptive Isogeometric and Embedded Domain Analysis*. PhD thesis, Technische Universität München, Munich, May 2012.

[110] D. Schillinger, I. Harari, M.-C. Hsu, D. Kamensky, S. K. F. Stoter, Y. Yu, and Y. Zhao. The non-symmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 309:625–652, September 2016.

[111] D. Schillinger and E. Rank. 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, November 2011.

[112] D. Schillinger and M. Ruess. 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, May 2014.

[113] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, and E. Rank. Small and large deformation analysis with the p- and B-spline versions of the Finite Cell Method. *Computational Mechanics*, 50(4):445–478, February 2012.

[114] P. J. Schneider and D. H. Eberly. *Geometric Tools for Computer Graphics*. The Morgan Kaufmann series in computer graphics and geometric modeling. Boston : Morgan Kaufmann Publishers, Amsterdam, 2003.

[115] J. A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Number 3 in Cambridge monographs on applied and computational mathematics. Cambridge University Press, Cambridge, 1996.

[116] I. Y. Shtaerman. Contact problem of the theory of elasticity. Technical report, DTIC Document, 1970.

[117] P. Silvester. Symmetric quadrature formulae for simplexes. *Mathematics of Computation*, 24(109):95–100, 1970.

[118] T. Strouboulis, K. Copps, and I. Babuska. The generalized finite element method. *Computer methods in applied mechanics and engineering*, 190(32):4081–4193, 2001.

[119] Y. Sudhakar, J. P. Moitinho de Almeida, and W. A. Wall. An accurate, robust, and easy-to-implement method for integration over arbitrary polyhedra: Application to embedded interface methods. *Journal of Computational Physics*, 273:393–415, September 2014.

[120] B. A. Szabó and I. Babuška. *Finite Element Analysis*. John Wiley & Sons, New York, 1991.

[121] B. A. Szabó, A. Düster, and E. Rank. The p-version of the finite element method. In E. Stein, editor, *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Chichester, West Sussex, 2004.

[122] İ. Temizer and C. Hesch. Hierarchical NURBS in frictionless contact. *Computer Methods in Applied Mechanics and Engineering*, 299:161–186, February 2016.

[123] H. Theisel. Exact isosurfaces for marching cubes. In *Computer Graphics Forum*, volume 21, pages 19–32. Wiley Online Library, 2002.

[124] M. Tur, J. Albelda, J. M. Navarro-Jimenez, and J. J. Rodenas. A modified perturbed Lagrangian formulation for contact problems. *Computational Mechanics*, 55(4):737–754, April 2015.

[125] V. Varduhn, M.-C. Hsu, M. Ruess, and D. Schillinger. 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, January 2016.

[126] C. V. Verhoosel, G. van Zwieten, and B. van Rietbergen. Micromechanical modeling of trabecular bone using isogeometric analysis. In *Advances in Computational Mechanics, A Conference Celebrating the 70th Birthday of Thomas J.R. Hughes*, 2013.

[127] C. Verhoosel, G. van Zwieten, B. van Rietbergen, and R. de Borst. 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, February 2015.

[128] B. Wassermann, S. Kollmannsberger, T. Bog, and E. Rank. From geometric design to numerical simulation: A direct approach using the Finite Cell Method on Constructive Solid Geometry. *Computers & Mathematics with Applications*, 2017.

[129] Z. Wassouf. *Die Mortar Methode Für Finite Elemente Hoher Ordnung*. PhD thesis, Technische Universität München, 2010.

[130] B. I. Wohlmuth. A Comparison of Dual Lagrange Multiplier Spaces for Mortar Finite Element Discretizations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 36(6):995–1012, January 2003.

[131] B. I. Wohlmuth. Variationally consistent discretization schemes and numerical algorithms for contact problems. *Acta Numerica*, 20:569–734, April 2011.

[132] B. I. Wohlmuth and R. H. Krause. Monotone multigrid methods on nonmatching grids for nonlinear multibody contact problems. *SIAM Journal on Scientific Computing*, 25(1):324, 2003.

[133] P. Wriggers. *Computational Contact Mechanics*. Springer, Berlin, New York, 2nd edition, 2006.

[134] P. Wriggers, J. Schröder, and A. Schwarz. A finite element method for contact using a third medium. *Computational Mechanics*, 52(4):837–847, March 2013.

[135] P. Wriggers, T. Vu Van, and E. Stein. Finite element formulation of large deformation impact-contact problems with friction. *Computers & Structures*, 37(3):319–331, 1990.

[136] P. Wriggers and G. Zavarise. A formulation for frictionless contact problems using a weak form introduced by Nitsche. *Computational Mechanics*, 41(3):407–420, February 2008.

[137] Z. Yang, S. Kollmannsberger, A. Düster, M. Ruess, E. G. Garcia, R. Burgkart, and E. Rank. Non-standard bone simulation: Interactive numerical analysis by computational steering. *Computing and Visualization in Science*, 14(5):207–216, April 2012.

[138] V. A. Yastrebov. *Numerical Methods in Contact Mechanics*. Numerical methods in engineering series. ISTE ; Wiley, London : Hoboken, NJ, 2013.

[139] N. Zander, T. Bog, M. Elhaddad, F. Frischmann, S. Kollmannsberger, and E. Rank. 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, October 2016.

[140] N. Zander, T. Bog, S. Kollmannsberger, D. Schillinger, and E. Rank. Multi-level hp-adaptivity: High-order mesh adaptivity without the difficulties of constraining hanging nodes. *Computational Mechanics*, 55(3):499–517, February 2015.

[141] G. Zavarise and L. De Lorenzis. An augmented Lagrangian algorithm for contact mechanics based on linear regression. *International Journal for Numerical Methods in Engineering*, 91(8):825–842, August 2012.

[142] G. Zavarise and P. Wriggers. A superlinear convergent augmented Lagrangian procedure for contact problems. *Engineering Computations*, 16(1):88–119, February 1999.

[143] G. Zavarise. The shifted penalty method. *Computational Mechanics*, 56(1):1–17, July 2015.

[144] G. Zavarise, P. Wriggers, and B. A. Schrefler. A method for solving contact problems. *International Journal for Numerical Methods in Engineering*, 42(3):473–498, 1998.

[145] O. Zienkiewicz, R. Taylor, and J. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, 6th edition, 2005.