Dipl.-Ing. Samir Omerović, BSc

Higher-order finite element methods for implicitly defined interface problems

DOCTORAL THESIS

to achieve the university degree of

Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dr.-Ing. habil. Thomas-Peter Fries

Institute of Structural Analysis

Second Supervisor

Prof. Dr.-Ing. habil. Alexander Düster

Numerical Structural Analysis with Application in Ship Technology

Graz, February 2017

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

Graz, _____ _____

Date Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

Graz, am _____ _____

Datum Unterschrift

Abstract

The present thesis introduces a new framework to treat boundary value problems with interfaces, a mathematical model which occurs often in nature and engineering. The distinctive feature of interface problems is the inherent presence of discontinuities, which can be kinks in the primary variable, as for example for material interfaces, or even jumps, in case of cracks. In fact, a large class of unfitted problems belongs also in this category, when the domain boundary does not match the computational mesh.

The numerical treatment of interface problems is notoriously difficult, and in particular standard finite element methods are not readily suited to deal with this class of problems. Many of the limitations stem from the inability of the method to resolve the jumps and kinks within elements. The achieved convergence rates, especially in case of higher-order approximations, are significantly degraded compared to the optimal ones. One known remedy is to use fictitious domain methods, that is, to use an unfitted geometry and ignore the interfaces and to enforce the interface conditions e.g. by means of Lagrange multipliers or Nitsche's method. In this approach, however, the complexity is postponed from the remeshing to the integration of the weak form. Another possibility is to use tedious and often time-consuming remeshing of the domain.

The suggested methodology in this thesis facilitates both approaches, for one it simplifies greatly the numerical integration and can be used as a "black box" for fictitious domain methods, on the other hand it can be used as a standalone remeshing or a fictitious domain method itself, depending on the point of view. The procedure starts from a higher-order background mesh that is independent of the geometry of interest. The geometry or the interface is defined implicitly by means of a level set function. The basic idea is to automatically identify and remesh the zero level set to obtain a discrete interface representation. The interface mesh is then by default independent of the background mesh which is subsequently decomposed along the interface. Customized mappings are used to incorporate the interface elements into the background mesh, that is, to construct higher-order sub-cells.

This background mesh can be used as an "integration mesh" in order to facilitate numerical integration for a broad class of methods as e.g. Finite Cell Method, XFEM or CutFEM. The proposed method delivers then a higher-order representation of the interface, which is not necessarily conforming, as hanging nodes are allowed. This mesh can then easily be processed by any solver to obtain the integration points for any desired order, on the interface element or within the resulting sub-cells.

In case that the decomposed mesh is conforming, the resulting higher-order discretization can be used directly for simulation purposes. Major issues are then the robustness of the method and the quality of the automatically generated elements, which are not necessarily well-shaped. However, numerical studies on various benchmark problems show that optimal convergence rates are obtained.

Collectively, the proposed approach shows excellent agreement between the optimal and the calculated convergence rates in two and three dimensions, for problems with non-fitting interfaces and boundaries. It is very flexible with respect to the employed element type and not limited to structured background meshes. These features render the proposed approach a candidate for remeshing a broad class of problems with an implicit sharp interface representation.

Kurzfassung

Das Ziel der vorliegenden Arbeit ist die Entwicklung eines neuartigen Lösungsansatzes, um Randwertprobleme mit diskontinuierlichen Koeffizienten zu approximieren. Dieses mathematische Modell ist ein häufig vorkommendes Problem, sowohl im Ingenieurwesen als auch in der Natur. Die Besonderheit dieser Problemklasse ist das inhärente Vorhandensein von Diskontinuitäten. Dies kann ein Knick in der Primärvariablen sein, beispielsweise bei Problemen mit sprunghaften Materialkoeffizienten, oder aber ein Sprung, beispielsweise bei Rissmodellierung. Eine weitere Klasse von Problemen die in diese Kategorie einzuordnen ist, sind Randwertprobleme mit unangepassten Gebietsrändern.

Die numerische Modellierung von Grenzschichtphänomenen ist bekanntermaßen schwierig. Dies gilt insbesondere für die Standardformulierung der Finite Element Methode. Die größte Schwierigkeit lässt sich dabei auf das Unvermögen der Methode, Knicke und Sprünge innerhalb von Elementen darzustellen, zurückführen. Die Konvergenzraten die mit der Standardformulierung erreicht werden, liegen dann weit unter den optimalen Konvergenzraten. Eine Abhilfe schaffen hier sogenannte Fiktive Gebietsmethoden. Dabei werden meist Hintergrundnetze verwendet, in die das Gebiet vollständig eingebettet ist, ohne dass Gebietsränder und Elementgrenzen aufeinander abgestimmt sind. Randbedingungen werden mittels Lagrange-Multiplikatoren oder mit der Nitsche Methode erzwungen. Bei diesem Vorgehen wird die Schwierigkeit der Vernetzung auf die Integration der schwachen Form umgelagert. Eine andere Möglichkeit ist eine rechen- und zeitaufwendige Neuvernetzung durchzuführen.

Die in dieser Dissertation vorgestellte Vorgehensweise erleichtert beide Zugänge, zum einem bietet es eine Möglichkeit die numerische Integration zu erleichtern und kann als ein "Black Box"-Aufsatz für Fiktive Gebietsmethoden verwendet werden. Auf der anderen Seite kann das Vorgehen an sich als eine Fiktive Gebietsmethode mit automatischer Neuvernetzung gesehen werden. Das Verfahren startet mit einem Hintergrundnetz höherer Ordnung, welches vollkommen unabhängig von der aktuellen Geometrie ist. Die tatsächliche Geometrie wird hierbei mittels einer Levelsetfunktion beschrieben. Die Grenzfläche ist dann die Nullisolinie dieser Funktion, die identifiziert und vernetzt wird, um eine explizite Darstellung der Grenzfläche zu erhalten. Das Grenzflächennetz ist dabei immer unabhängig vom Hintergrundnetz, welches anschließend entlang der Grenzfläche unterteilt wird. Spezielle Abbildungsvorschriften werden dann benutzt, um das Grenzflächennetz in das restliche Gebietsnetz einzubetten.

Dieses "verbesserte" Hintergrundnetz kann nun als "Integrationsnetz" für eine ganze Klasse von Methoden wie z.B. Finite Cell Method, XFEM und CutFEM benutzt werden. Hierbei ist zu beachten, dass dieses Integrationsnetz nicht unbedingt konform sein muss, also hängende Knoten durchaus erlaubt sind. Dieses Netz kann anschließend benutzt werden um Integrationspunkte, unabhängig von der Sorte der numerischen Integration, in das Rechengebiet abzubilden.

Im Falle, dass das zerlegte Netz konform ist, kann das automatisch generierte Netz höherer Ordnung direkt für die Simulation herangezogen werden. Wichtige Themen sind dann die Robustheit der Methode und die Qualität der automatisch generierten Elemente. Numerische Studien auf den automatisch generierten Netzen bestätigen, dass optimale Konvergenzraten erhalten werden.

Zusammenfassend zeigt die Methode optimales Konvergenzverhalten, sowohl für unangepasste Ränder wie auch Grenzflächen. Das Verfahren ist für zwei und dreidimensionale Probleme geeignet und auf keinen Elementtyp oder auf strukturierte Hintergrundnetze limitiert. Diese Besonderheiten machen die Methode zu einem ernsthaften Anwärter für eine größere Klasse von Problemen mit impliziter und scharfer Darstellung von Grenzflächen.

Acknowledgements

First and foremost I would like to thank my supervisor, Prof. Dr.-Ing. Thomas-Peter Fries, for guiding me throughout the course of this thesis work. His enthusiasm for all topics and aspects in computational mechanics is truly infectious. I cannot thank him enough for his encouragement and his invaluable insights as well as providing new impulses whenever I got stuck. I feel deeply privileged for the opportunity to work with him.

Furthermore, I would like to express my gratitude to Prof. Dr.-Ing. Alexander Düster for providing his time to examine this thesis. His interest in my work is gratefully acknowledged.

I have also been fortunate to work with some great colleagues. I would like to thank all colleagues at the *Institute of Structural Analysis*, in particular Christian Dünser and Daniel Schöllhammer, for useful and interesting discussions.

Finally, on a personal level, I want to thank my family and my friends for their support and help at all times.

Last, but certainly not least, I would like to thank Julia. Not only for her emotional support but also for her patience and her encouragement throughout the process.

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Main contributions of the thesis . . . 3
1.3 Outline of the thesis . . . 3

2 Higher-order finite element methods on fitted meshes **5**
2.1 Model problems . . . 6
2.1.1 Poisson problem . . . 7
2.1.2 Governing equations for linear elasticity . . . 8
2.1.3 Governing equations for elastoplasticity . . . 9
2.2 Continuous variational formulation . . . 14
2.3 Discrete formulation . . . 17
2.3.1 Discrete variational formulation for Laplace and elasticity problems . . . 17
2.3.2 Discrete variational formulation for elastoplasticity problem . . . 19
2.3.3 Discrete formulation of local constitutive equations . . . 20
2.4 The finite element . . . 21
2.5 Spatial discretization . . . 26
2.6 Conditioning of system matrices . . . 30
2.7 Convergence rates . . . 31

3 Higher-order finite element methods on unfitted meshes **37**
3.1 Level set method . . . 38
3.2 Extended finite element method . . . 41
3.3 Finite cell method . . . 48
3.4 Cut finite element method . . . 50
3.5 Common issues associated with embedded domain methods . . . 51
3.5.1 Imposition of boundary and interface conditions . . . 52
3.5.2 Linear dependence due to slightly cut elements . . . 54
3.5.3 Numerical integration in cut elements . . . 54
3.6 Other methods . . . 55

4 Reconstruction of zero level sets **56**
4.1 Reconstruction strategy in a nutshell . . . 56
4.2 Detection of cut elements . . . 58
4.3 Valid level set data . . . 60
4.4 Element topologies . . . 62
4.5 Subdivision strategy for triangles . . . 63
4.5.1 Local subdivisions . . . 63
4.5.2 Non-local subdivisions . . . 66
4.6 Subdivision strategy for quadrilaterals . . . 67
4.6.1 Local subdivisions . . . 67
4.6.2 Non-local subdivisions . . . 68

4.7 Topologies in cut tetrahedral elements . . . 69
4.7.1 Local subdivisions . . . 69
4.8 Topologies in cut hexahedral elements . . . 69
4.9 Notes on implementational issues . . . 71
4.10 Reconstruction of zero level set by Newton-Raphson procedure . . . 71
4.10.1 Newton–Raphson scheme with prescribed search directions . . . 72
4.10.2 Starting points for the Newton-Raphson iteration . . . 73
4.10.3 Search paths . . . 73
4.10.4 Search paths in two dimensions . . . 74
4.10.5 Search paths in three dimensions . . . 75
4.11 Examples for reconstructed zero level sets . . . 79
4.11.1 zero level sets in two-dimensional reference elements . . . 79
4.11.2 zero level sets in three-dimensional reference elements . . . 79
4.11.3 zero level sets by recursive refinement . . . 80
4.11.4 zero level sets in meshes . . . 81

5 Higher-order integration techniques in/on implicitly defined domains 86
5.1 State of the art for discontinuous integrands . . . 88
5.1.1 Methods that use subdivision techniques . . . 89
5.1.2 Methods that use modified integration rules. . . . 89
5.2 Higher-order accurate integration of discontinuous functions using a conformal decomposition technique . . . 90
5.3 Coordinate mappings . . . 90
5.3.1 Mapping for triangular elements . . . 94
5.3.2 Mapping for quadrilateral elements . . . 98
5.3.3 Mapping for tetrahedra . . . 99
5.3.4 Mapping for prisms . . . 103
5.3.5 Mapping for hexahedra . . . 104
5.4 Integration rules in cut elements . . . 105

6 Conformal higher-order decomposition finite element method 106
6.1 Outline of the method . . . 106
6.2 A modified version of the method . . . 108
6.3 Treatment of sharp features . . . 110
6.4 Influence of the decomposition scheme on the condition number . . . 111
6.5 Remedies to ill-conditioning of the linear system of equations . . . 112
6.5.1 Nodal moving procedures . . . 113
6.6 Invalid topologies/meshes . . . 115
6.7 Conformal higher-order decomposition finite element method for elastoplasticity . 116
6.8 Detection of the elastoplastic zone . . . 117
6.9 Mesh update strategy . . . 118

7 Numerical studies **120**
7.1 Interface reconstruction . . . 121
7.1.1 Numerical results in two dimensions - reconstruction of a circular interface 121
7.1.2 Numerical results in two dimensions - reconstruction of a flower-shaped inclusion . . . 127
7.1.3 Numerical results in three dimensions - reconstruction of a spherical interface 131
7.1.4 Numerical results in three dimensions - Goursat's surface . . . 133
7.1.5 Summary . . . 134
7.2 Integration of functions and their interpolation in/on zero level sets . . . 137
7.2.1 Numerical results in two dimensions . . . 137
7.2.2 Integrating functions *on* the zero level set . . . 137
7.2.3 Error for integrating functions *in* implicitly defined domains . . . 139
7.2.4 Numerical results in three dimensions . . . 141
7.2.5 Error for integrating functions *on* the zero level set . . . 142
7.2.6 Error for integrating functions *inside* implicitly defined domains . . . 142
7.2.7 Summary . . . 143
7.3 Boundary value problems I - Interface problems . . . 144
7.3.1 Numerical results in two dimensions - bi-material scalar-valued problem . 144
7.3.2 Numerical results in two dimensions - bi-material vector-valued problem . 148
7.3.3 Spherical elastic inclusion under uniaxial tension in an infinite domain . . . 150
7.4 Boundary value problems II - Fictitious domain problems . . . 155
7.4.1 Infinite plate with a circular hole . . . 155
7.4.2 Spherical cavity under uniaxial tension in an infinite domain . . . 158
7.5 Elastoplastic problems . . . 159
7.5.1 Elastoplastic hollow cylinder . . . 159
7.5.2 Elastoplastic hollow sphere under internal pressure . . . 162

8 Conclusions and outlook **165**
8.1 Summary . . . 165
8.2 Open problems and outlook . . . 165

References **167**

1 Introduction

1.1 Motivation

After more than half a century of intensive research in finite element methods, the most fundamental questions regarding (linear) FE analysis are now to a great extent answered. In the last couple of years in many engineering and industrial applications as e.g. biomechanics, modeling of image based geometries etc. the mathematical and the geometrical models became more challenging and by now the question of *efficient* numerical solution methods is of paramount importance. A particularly attractive class of methods, due to the higher accuracy reached at a lower computational cost and additional robustness, are higher-order finite elements, the so-called p-version of the FEM. There are several (non-academic) examples, where only a p-FEM approach is capable to solve a problem on a required level of accuracy within reasonable computational time and effort, see [168, 157]. The application field of the p-FEM is wide, however, the method can only exploit its strength given certain smoothness requirements on the field variables and an accurate description of boundaries and interfaces. In fact, in this case it can be shown that the solution converges exponentially with the order of the approximating polynomial.

Nowadays, a major field of interest are the so-called interface problems, a widely spread class of problems with discontinuous coefficients. The discontinuities may occur in the primary field of interest (e.g. displacement, temperature), or alternatively in the secondary field (stresses, flux). These types of problems arise frequently and in various contexts, e.g. to represent bi-material structures [165] in solid mechanics, but also for a variety of other problems across different fields of research as shape and topology optimization [27], two-phase flow problems [88], free surface flows, phase transition problems [43], etc. We mention that problems with unfitting domain boundaries also belong to this category. There, the formulation of a problem is extended to a larger domain containing the original domain, turning the boundary conditions rather to interface conditions. The numerical treatment of interface problems, especially in a higher-order context, presents a great challenge, which is due to the requirements on both the accuracy and the robustness of the underlying algorithms. As mentioned, standard higher-order finite element methods are not readily suited for approximation of interface problems, as the inherent discontinuities in the domain demand tedious remeshing procedures. If the interface is not accounted for, that is, if it is not resolved by the mesh, the convergence rates are significantly degraded compared to the optimal rates. In fact, the convergence rates are then even *independent of the polynomial degree p* of the finite element space, which makes any higher-order procedure obsolete.

A remedy to this problem, which has lately experienced a lot of attention in the computational mechanics community, is offered by the class of the fictitious domain methods (FDM). A large number of FDMs emerged in the last few years, in most instances with the precise goal to minimize the meshing costs, which is from the early days of FEM one of the bottlenecks of the classic method. These methods operate with a geometry independent mesh, therefore facilitating or even avoiding mesh generation in a classical sense. The basic idea is to embed the possibly complex geometry in a simple surrounding, "fictitious" domain and apply the boundary conditions (and sometimes interface conditions) by some tailored approaches. There is a vast amount of literature to this topic, see [53, 91, 66]. Among the huge variety of methods in this category are the early approaches with the Immerse Boundary Method [145], the Penalty Method [8] as well

as the Embedding Domain Method [30]. The boundary conditions are enforced with specialized techniques as the penalty method or Lagrange multipliers in various versions [4]. The boundary conditions are particularly significant also in another context of FDMs, the numerical integration. It is important to emphasize that the accuracy of the FDMs directly correlates with the accuracy of the numerical integration scheme. When applying boundary conditions e.g. via Nitsche's method additional surface terms arise. Now, these surface terms may be defined over curved interfaces, which means that the numerical integration becomes quite demanding and complex. The application of the boundary conditions represents one drawback of the fictitious domain methods, another disadvantage concerns the implementation. Especially industrial solvers rely rather on a standard FEM formulation, which needs to be extended and heavily modified to work as a solver for FDMs. For instance, massive changes are necessary in order to deal with a variable number of unknowns, incorporation of additional constraints, non-standard application of boundary conditions, specialized solvers in case of a non-standard structure of the system of equations etc. Another point is that most of the fictitious domain methods, for some or another reason, are only applied in a low-order context. Notable exceptions are the finite cell method [142], which was actually developed already with the higher-order context in mind, and a recently introduced derivative of the extended finite element method, the CutFEM [29]. Both methods are described in the remainder of the thesis.

The motivation for this thesis stems from the ambition to develop an *efficient* and *conceptually simple* higher-order finite element method for interface problems in two and three dimensions. As common in FDMs, the interface is described implicitly by means of a zero level set, and is interpolated by means of standard FE functions on the higher-order background mesh. As mentioned before, for the integration of the weak form, an accurate explicit description of the interface is inevitable. This is the first crucial point of the thesis: the detection and meshing of the zero level set, in the thesis referred to as reconstruction, offers a higher-order accurate explicit representation of the interface. This representation may then be used to generate any desired integration rule on the interface mesh. To construct higher-order sub-cells, it is necessary to decompose the cut elements, such that the reconstructed interface elements match one element face. For any point within the sub-cells a customized mapping is constructed, and it is of utmost importance that the mappings fulfill some regularity requirements. One of the benefits of the approach is that the whole procedure may be used as an additional module to existing codes, as a "black-box" which delivers a higher-order geometry representation which aligns with all interfaces and boundaries. This procedure allows for the user to decide which integration rule and which integration order to choose. If the mesh is used purely for integration purposes, it is not necessary that it is conforming, hanging nodes are allowed, which is convenient for local refinement.

If the mesh is supposed to be used in the context of standard FEM simulations, some additional measures need to be taken. This is along the lines of the so-called *conformal decomposition FEM* (CDFEM) [133, 107], which, herein, is extended to higher-order elements. Hanging nodes must be avoided, therefore refinement is possible only globally or in a neighborhood, but not exclusively in one element. The main point concerns the quality of the element shapes, as it is not ensured that the elements are always well-shaped. As shown in the thesis, the sometimes awkwardly shaped elements, at least the type which may occur in the automatically decomposed sub-cells, do not hinder optimal accuracy properties. However, they affect the condition number

which means that the accuracy is indirectly influenced. For this reason some mesh manipulations are introduced, and as shown, relatively little modifications on the mesh result in a considerable improvement of the condition number of the system matrix. In terms of convergence properties, in any investigated norm, the calculated rates are optimal or close-to-optimal to the theoretically ideal rates. This applies for pure interpolation problems as well as boundary value problems with unfitted boundaries and boundary value problems with unfitting interfaces within the domain.

1.2 Main contributions of the thesis

This thesis focuses on the development of a novel, higher-order method to treat interface and unfitted domain problems within one unified framework. The individual contributions are summarized thematically below.

- Higher-order extension of the conformal decomposition finite element method. The method was, up to now, applied only to low-order approximations. This work extends the method to higher-order accurate approximations for two and three dimensional problems in linear elasticity.
- Development of higher-order accurate integration schemes for discontinuous integrands. The accuracy of any fictitious domain method relies heavily on the performance of the employed quadrature method. We present a new algorithm which can approximate integrands with curved discontinuities with arbitrary high-order accuracy.
- Use of customized mapping procedures to construct curved higher-order sub-cells. The application of specialized mappings is of utmost importance for the CDFEM as well as for the integration within cut domains. We show that the regularity of the map strongly influences the interpolation properties of the resulting sub-cells and describe mappings that perform optimally.

The minor contribution concerns the application of the CDFEM to elastoplastic problems. Even if for some special cases optimal rates can be achieved, the general procedure is restricted to low-order accurate approximations. This is less due to the CDFEM, but can rather be traced back to the detection and approximation of the elastoplastic interface. It is noted that the above mentioned contributions make profit of shared efforts at the Institute of Structural Analysis to develop and investigate higher-order methods for problems with internal boundaries and interfaces.

1.3 Outline of the thesis

Following this introduction, **Section 2** gives an overview on the standard, i.e. fitted finite element method, meaning that the computational mesh approximately matches the domain boundary. It is an introduction into higher-order finite element methods and outlines important aspects which are detailed in the next sections. **Section 3** presents briefly the basics of embedded domain methods. The focus is on the principles of the standard extended finite element method (XFEM), the CutFEM and the finite cell method (FCM). In **Section** 4 we focus on the identification and the remeshing of the zero level set in two and three dimensions. Given an arbitrary background mesh, the implicitly given zero level set is transformed into an explicit representation. **Section** 5

deals with integration techniques. The focus is on a novel numerical integration technique of discontinuous integrands on zero level sets as well as inside the resulting volumes. Moreover, tailored element mappings are introduced and thoroughly examined. In **Section 6** a new methodology is presented to construct higher-order accurate meshes given an implicitly defined interface on a background mesh. Furthermore, some stability issues are addressed when using these automatically generated meshes in FE analyses. In **Section** 7 the newly proposed method is analyzed using numerical examples. The focus is on the accuracy of the reconstruction, the interpolation properties and the performance for boundary value problems with interfaces in two and three dimensions. Finally, in **Section** 8, the present work is summarized in the conclusions and some possible future researches and extensions of the work are discussed.

2 Higher-order finite element methods on fitted meshes

The standard finite element method (FEM) is nowadays already a classical method for finding approximate solutions to boundary value problems of partial differential equations (PDEs). Even though the name "finite element method" was coined just at the end of the 1950s, the origins of the method have a long history, depending on what is considered the essential part of the finite element method. Some of the ideas can clearly be traced back even to the 17^{th} century. Back then, the ideas include utilization of minimization principles and piecewise linear approximation in order to *derive* a differential equation (G.W. Leibnitz, 1696 [106]), utilizing variational calculus to obtain approximate solutions (W. Ritz, 1908 [9]), and the ingenious insight to use orthogonality conditions rather then exploiting minimum principles to form the resulting system of equations (B. Galerkin, 1915 [158]).

In 1943 R. Courant proposed to approximate a Poisson problem using piecewise linear functions via minimization of a corresponding energy functional [45]. This approach clearly had all attributes of the modern FEM and might be considered as the first "real" finite element analysis. In fact, already in 1922, in a long footnote in [101] he introduced a triangular mesh and described how to construct convergent approximation spaces for a Dirichlet problem. End of the 1950s, without being aware of Courant´s work, structural engineers started to work on the Stiffness Matrix Method, which was later on recognized to be a solid mechanics analogy to the FEM. In a landmark paper on linear plane elasticity problems in 1960, R.W. Clough used the name "finite element method" in a publication [39] for the first time. In the 1960s and 1970s the purely mechanical approach was supplemented by mathematical considerations. A rigorous mathematical foundation was developed in [5, 161, 37], including convergence proofs, a priori error estimates and also establishing important results regarding the interpolation theory. After that the method expanded in practically every research direction and offers nowadays almost unlimited possibilities in various application fields. For more on the history of the finite element method see [111] and the overview papers in [135, 179].

Today, numerous versions of the finite element method are available. When classifying the approaches based on refinement strategies, two different procedures emerge. For one, there is the classical h-version of the FEM, with the polynomial degree being rather low (usually $p = \{1, 2\}$) and the refinement realized by increasing only the element number, e.g. [15, 180]. On the contrary, there is the p-version, utilizing high-order polynomials with the refinement carried out only by increasing the polynomial degree. The p-version attracted a lot of attention in research in the last two decades [168] and is also in focus for this section. There is also a combination of the two approaches, referred as hp-FEM, where a careful selection of h- and p-refinement delivers exponential convergence rates in terms of degrees of freedom, even for some cases of non-smooth problems as corner singularities. We emphasize, that all methods described in this section utilize bodyfitted meshes, meaning that the discretization resolves the actual domain boundaries and interfaces. Another nowadays popular variant is the large class of methods applying non-bodyfitting meshes, which is the main topic of Section 3

The objective of this section is to give a reasonably self-contained overview of the regular finite element method using the Poisson problem and linear elasticity as well as elastoplasticity as prime examples. In all cases it is assumed that the spatial discretization, further on referred

to as mesh, aligns with all internal and external interfaces. For related approaches and similar notation see [82, 24, 156]. First, the governing equations for the problems are introduced, together with the associated boundary conditions. Next, the transition from the strong form to the weak form is shown, reducing the requirements on the regularity of the involved functions. A standard Galerkin finite element formulation is then presented together with the concept of isoparametric elements. Then an introduction into the important topic of mesh generation is given, explaining the standard approaches. Finally, the topic of conditioning of the system matrices is discussed together with the convergence rates in different norms which will be important in the following sections.

2.1 Model problems

Three different boundary value problems are introduced, ordered in increasing complexity. First a scalar problem is shown, represented by the well-known Poisson problem. Then a vectorized version of the same linear elliptic type of equations is described, utilizing the equations of linear elasticity. Finally, the last problem describes elastoplasticity, and represents a nonlinear and non-smooth problem. In the latter case the problem is confined to small strains and a linear material model in the elastic regime. Thus, the non-linearity is found completely in a restriction of the stresses to an admissible set.

For all explanations in this section only 2D problems are considered, however the (formal) extension to the more general case of dim $= 3$ is mostly self-explanatory. The model example including the used notation is shown in Fig. 1. For all equations holds that the body is made up by two different materials, denoted generically as Ω^+ and Ω^-, with $\Omega := \Omega^+ \cup \Omega^-$. The boundary of the computational domain is considered further as external interface Γ_{ext} and any interface inside the body, e.g. material interfaces, the boundary separating elastic and plastic zones or an interface describing two sides of a crack path as internal interface Γ_{int}.

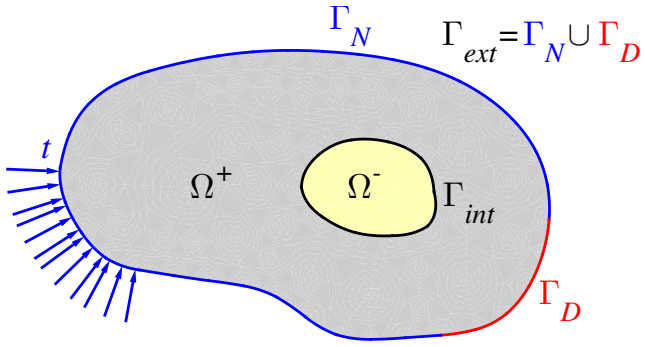


Figure 1: This sketch introduces basic notations and demonstrates the initial state of the body as well as the boundary conditions. The two domains Ω^+ and Ω^- represent zones with generically different material properties, as e.g. two completely different materials or the same material but Ω^+ behaving elastically and Ω^- elastoplastically.

2.1.1 Poisson problem

The Poisson equation is *the* canonical example for a linear elliptic partial differential equation (PDE) and the main reason for its popularity is the comparably simple structure of the equations and the existence of a number of analytical solutions. The governing field equations together with the corresponding boundary conditions describe a wide variety of physical phenomena. In particular, this includes the description of elastic membranes, heat transfer, gravitational fields, diffusion processes, magnetostatics, electrostatics and many other physical processes.

Let Ω represent some open bounded set in $\mathbb{R}^2$, and $\Gamma = \partial\Omega$ the boundary. The unknown scalar function, denoted as $u(\boldsymbol{x})\colon \Omega \to \mathbb{R}$, is defined for every point $\boldsymbol{x} := [x_1, x_2]^\intercal$ inside the domain Ω. Furthermore a source (or load) function $f(\boldsymbol{x})$ is given on Ω. The Poisson problem is then defined in Ω as

$$\begin{aligned} -\nabla \cdot (k^+(\boldsymbol{x})\nabla u(\boldsymbol{x})) &= f(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \Omega^+ \\ -\nabla \cdot (k^-(\boldsymbol{x})\nabla u(\boldsymbol{x})) &= f(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \Omega^- \end{aligned} \tag{2.1}$$

with the $\nabla := [\partial/x_1, \partial/x_2]^\intercal$, denoting the gradient operator and the material conductivity $k^+(\boldsymbol{x}) \in \Omega^+$ as well as $k^-(\boldsymbol{x}) \in \Omega^-$. From here on, for simplicity reasons we take $k^+ = k^- = \text{const} = 1$. The assumption of constant unity material coefficients reduces Eq. (2.1) to the well known form

$$-\nabla^2 u(\boldsymbol{x}) = f(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \Omega. \tag{2.2}$$

To complete the boundary value problem, the field equations are supplemented by two sets of boundary conditions (BCs) as

$$u(\boldsymbol{x}) = 0 \qquad \forall \boldsymbol{x} \in \Gamma_D \subseteq \partial\Omega \tag{2.3}$$

$$t(x) = \nabla u(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \Gamma_N \subset \partial\Omega \tag{2.4}$$

with the unit outward normal $\boldsymbol{n}(\boldsymbol{x})$ in point $\boldsymbol{x}$ on $\partial\Omega$. Notice that the function value on Γ_D is for simplicity reasons prescribed as zero, however, it is possible to impose any other (smooth) function $g(\boldsymbol{x})$ which would rather complicate only the formal treatment. For all the problems we consider from here on, the relations $\partial\Omega := \Gamma_D \cup \Gamma_N$, $\emptyset := \Gamma_D \cap \Gamma_N$ and $\Gamma_D \neq \emptyset$ are *always* valid in order to have a unique solution and a well-posed problem, see e.g. [61]. In case of different material conductivities, it is necessary to prescribe interface conditions or internal boundary conditions along the internal interface Γ_{int} as

$$\begin{aligned} [\![u]\!] &= 0 \qquad \forall \boldsymbol{x} \in \Gamma_{int} \\ [\![k\nabla u \cdot \boldsymbol{n}]\!] &= 0 \qquad \forall \boldsymbol{x} \in \Gamma_{int} \end{aligned}$$

with the jump operator $[\![f(\boldsymbol{x})]\!] = \lim_{\varepsilon \to 0} f(\boldsymbol{x} + \varepsilon) - f(\boldsymbol{x} - \varepsilon)$. This condition enforces continuity in the primary field and in the flux.

To give the equations a physical meaning: the divergence part of Eq. (2.2) denotes the density of the field variable $u(\boldsymbol{x})$ (e.g. temperature) in an equilibrium state. The right hand side of Eq. (2.2)

are body forces inside the domain Ω and correspond to some sources inside the domain. In case that the right hand side is zero, the equation is called *Laplace equation*. The boundary condition in Eq. (2.3) is also known as *Dirichlet boundary condition* or *essential boundary condition* and assigns prescribed values of the primary variable on the boundary. This corresponds to e.g. prescribed temperature in heat transfer problems. The boundary condition Eq. (2.4) is known as *Neumann boundary condition* or *natural boundary condition* and specifies the value of the normal derivative of the primary unknown, e.g. prescribed flux in case of heat transfer problems.

Regarding the continuity requirements on the involved functions, the function $u \in C^2(\Omega) \cap C^1(\bar{\Omega}) \cap C^0(\bar{\Omega})$ with $\bar{\Omega} := \Omega \cup \Gamma$. The space $C^k(\Omega)$ is defined to be the set of all continuous functions $u(\boldsymbol{x})$ on Ω, such that $u(\boldsymbol{x})$ and all partial derivatives up to order k exist and are continuous on Ω. The continuity requirement on the unknown function $u(\boldsymbol{x})$ inside the domain Ω is higher than on the closure of the domain $\bar{\Omega}$, which can be explained by the reduced regularity requirements on the boundary conditions in Eqs. (2.3) and (2.4). The proper setting for the loading function $f(\boldsymbol{x})$ is in $L^2(\Omega)$. This function space consists, loosely spoken, of functions that are more rough that the functions from $C^k(\Omega)$, a precise definition will be given later.

2.1.2 Governing equations for linear elasticity

A typical example for a vector-valued version of a linear elliptic PDE are the equations of linear elasticity. The unknown or dependent variable is the vector-valued displacement field $\boldsymbol{u}(\boldsymbol{x})$, with the components u_i, $i \leq 3$, which in general depend on the spatial position $\boldsymbol{x}$. The underlying physical assumptions leading to the linearity of the model are restrictions on the magnitude of the deformations (infinitesimal strain theory, therefore small displacements and small rotations) as well as homogeneous and isotropic material. Furthermore, a linear relationship between stresses and strains (linear material law) is assumed. The system of resulting field equations, valid for every point $\boldsymbol{x}$ inside the computational domain Ω, is given as

$$-\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \Omega \tag{2.5}$$

$$\boldsymbol{\sigma}(\boldsymbol{u}) = 2\,\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda\,\mathrm{tr}[\boldsymbol{\varepsilon}(\boldsymbol{u})]\,\mathbf{I} \qquad \forall \boldsymbol{x} \in \Omega \tag{2.6}$$

$$\boldsymbol{\epsilon}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\mathsf{T}}\right) \qquad \forall \boldsymbol{x} \in \Omega \tag{2.7}$$

The first equation, Eq. (2.5), describes the balance of internal and external forces and comprises a vector with entries for every dimension. The tensor $\boldsymbol{\sigma}$ is the (Cauchy) stress tensor with components σ_{ij}, $i, j \leq 3$, which is symmetric due to the balance of angular momentum; the vector $\boldsymbol{f}$ is the applied body force per unit volume with the components f_i, $i \leq 3$.

The second equation, Eq. (2.6), represents the material model and is hereby taken for simplicity as the linear elastic Hooke's law. The equation can also be rewritten in the matrix form $\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}$, and consequently the elasticity tensor $\mathbb{C}$ can be interpreted as a linear mapping from the strain to the stress space. The elasticity tensor is typically assumed to possess major and minor symmetries (see [100]), which reduces the number of independent components significantly. Note that $\mathbb{C}$ is here given in terms of the two Lamé-Navier parameters $\mu > 0$ and $\lambda > 0$, their connection with the more familiar Young's modulus and the Poisson ratio is given as

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{and} \quad \mu = \frac{E}{2(1+\nu)}. \tag{2.8}$$

A key factor when formulating the equations for *plasticity* is the manner how kinetic quantities are described, i.e. how the split into the dilatational (volumetric) and deviatoric properties is performed. The trace of the strain tensor is defined as $\mathrm{tr}[\boldsymbol{\epsilon}(\boldsymbol{u})] := \nabla \cdot \boldsymbol{u}$, and leads to the stress part in Eq. (2.6) that is responsible for volumetric changes. The deviatoric stress is then the remaining part of the equation, which can also be written as

$$\boldsymbol{\sigma}(\boldsymbol{u}) = 2\,\mu\boldsymbol{\epsilon}(\boldsymbol{u}) + \lambda\,\mathrm{tr}[\boldsymbol{\epsilon}(\boldsymbol{u})]\,\mathbf{I} = \mathrm{dev}[\boldsymbol{\sigma}] + \mathrm{vol}[\boldsymbol{\sigma}] \tag{2.9}$$

The small strain tensor $\boldsymbol{\epsilon}$ is defined in terms of the deformation gradients via Eq. (2.7). Observe that only the symmetric part, $\nabla^s \boldsymbol{u} = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^\intercal\right)$, of the displacement gradient is used, the skew-symmetric part $\nabla^w \boldsymbol{u} = \frac{1}{2}\left(\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^\intercal\right)$ gives rise to infinitesimal rotations and is neglected in small-strain theories.

Again, the governing differential equations are supplemented by boundary conditions on the primary unknowns, i.e. prescribed displacements, and on the normal derivatives, which is a condition on the projection of the stress tensor normal to the boundary, as

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}) &= \mathbf{0} && \forall \boldsymbol{x} \in \Gamma_D \subseteq \partial\Omega \\ \boldsymbol{t}(\boldsymbol{x}) &= \boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) && \forall \boldsymbol{x} \in \Gamma_N \subset \partial\Omega \end{aligned}$$

For one-dimensional problems the Poisson equation and linear elasticity both clearly degenerate to the same ordinary differential equation with different material parameters.

2.1.3 Governing equations for elastoplasticity

Whereas elasticity is intimately connected with energy conservation and arises from a reversible deformation of crystal lattices, elastoplastic phenomena describe thermodynamically dissipative processes. Plasticity implies immediately *irreversible* processes and consequently the arising discontinuities occurring on a micro-structural level (further on referred as plastic deformations) as well as the path dependence must be reflected in the mathematical model.

The governing physical features of an elastoplastic model may be explained taking reference to the graphs in Fig. 3, representing the idealized behaviour of an one-dimensional bar with a uniform cross section under axial loading as in Fig. 2. The only non-zero stress component is σ_x, which is then interchangeable with the normal force N_x, for our example $q(x) = 0$, therefore, the only load is a single force on the right boundary of the bar.

The given data are the idealized stress-strain curve as depicted in Fig. 3(a), with the Young's modulus E and the hardening modulus K, as well as the initial yield stress σ_{Y_0} and the associated strain ϵ_{Y_0}. The loading history of the bar is represented by a variation of the load factor λ as a function of a pseudo-time τ, as shown in Fig. 3(b). The final load-deflection behaviour of the bar is depicted in Fig. 3(c) for an incremental deformation model (*Prandtl-Reuss plasticity*) with isotropic hardening and in Fig. 3(d) for a total deformation model (sometimes also referred to as

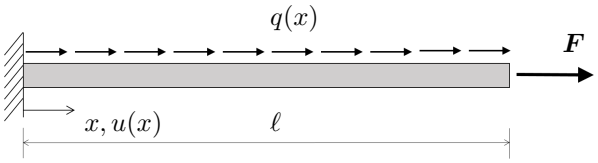


Figure 2: One-dimensional elastoplastic bar with a constant cross sectional area under axial stress. The rod is built-in at the left boundary and a single force acts at the right boundary.

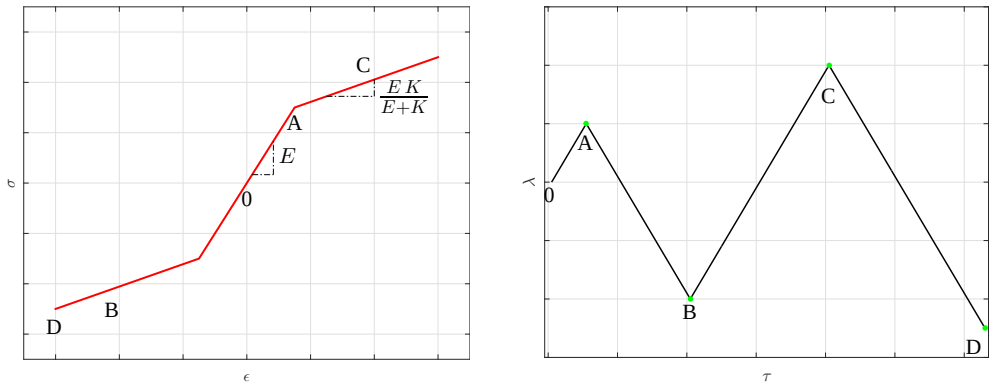
Hencky-plasticity).

As mentioned, one of the key features of plasticity, even in case of quasi-static response, is the dependence on a time-like variable. This time-like variable corresponds to some generic parametrization for the applied load. Note that in this explanation we are referring to the theory of incremental plasticity, Fig. 3(c), the total deformation theory is in fact independent of a pseudo-time.

Starting from an unloaded configuration in $\mathbf{O}$ ($\lambda = 0$), the load is increased to $\mathbf{A}$ ($\lambda = 1$). The current stress is still smaller than the yield stress σ_{Y_0}, the first yielding happens at $|\lambda| > 1.5$, therefore the material behaves completely elastic. This stress path is also referred as elastic domain and any incremental (small) change of load (e.g. unloading of the body from $\mathbf{A}$ to $\mathbf{O}$ or further loading in the other direction to $\bar{\mathbf{A}}$) would result in a purely elastic response of the system. However, now the loading direction is reversed and the body is compressed to point $\mathbf{B}$ ($\lambda = -2.0$). At this point the limit stress is exceeded, that is, the absolute value of the current stress $|\sigma_x| > \sigma_{Y_0}$, therefore leading to an inadmissible stress state. Just at the transition from elastic to elastoplastic behaviour ($\lambda = -1.5$) it is necessary to introduce a new variable, the plastic (or permanent) strain $\boldsymbol{\epsilon}^{pl}$. The growth in ϵ^{pl} decreases the elastic strain ϵ^{el} (note that the total strain ϵ *must* remain constant) and consequently also the current stress σ_x. Note that also the slope of the stress-strain curve is now significantly reduced to $\frac{E+K}{E\,K}$, instead of E as in the purely elastic domain. However, there are also restrictions on the slope of the stress-strain curve, for stable materials it must hold that $\frac{E+K}{E\,K} > 0$. Now the stress is reversed once more (tensile load to $\lambda = 2.0$) and apparently the whole domain, from $\mathbf{B} - \mathbf{C}$, is now completely elastic. Therefore, increasing the yield stress in compression regime has also increased the yield stress for tensile behaviour, this behaviour is also called isotropic hardening. Now, reversing the load direction a last time to $\lambda = -2.5$, we see that the load behaviour curve snaps back to the initial plastic behaviour at $\mathbf{B}$ and continues with the decreased slope of the stress-strain curve.

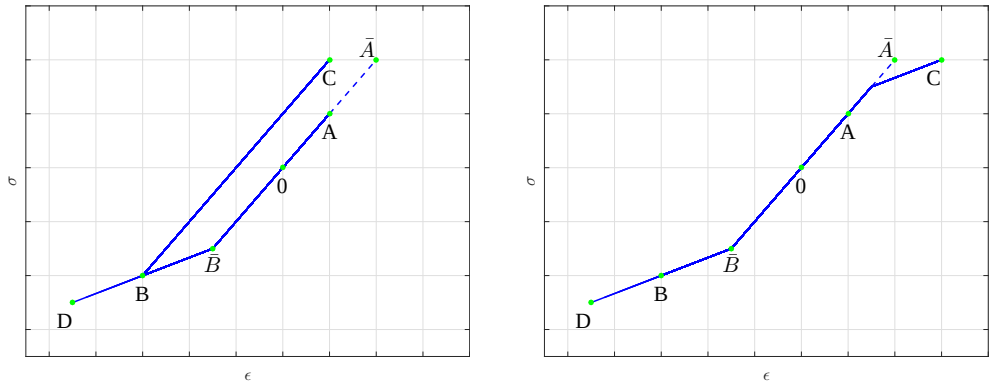
Analyzing the behaviour we notice two key points. (i) there is an unambiguous, sharply defined limit stress which bounds the elastic domain (σ_Y) and (ii) additional variables need to be introduced describing the evolution of the elastoplastic process (plastic flow and hardening processes). In the standard elastoplastic models these are the plastic strain $\boldsymbol{\epsilon}^{pl}$ and a set of so-called internal stress-like variables (e.g. the backstress in cycling deformation processes) further denoted by $\boldsymbol{q}$ and the associated energy conjugated strain-like variables $\boldsymbol{\alpha}$. The size of this set varies depending on the emerging plastic process, that is, in isotropic hardening it is simply a scalar, in kinematic hardening it is a second order tensor describing a stress variable.

From a mathematical point of view, the resulting equations gives rise to *variational inequalities* (in contrast to variational equations in elasticity). The system of equations consists of standard



(a) Stress-strain relationship

(b) Load factor-pseudo time diagram



(c) Incremental plasticity in 1D

(d) Deformation plasticity in 1D

Figure 3: Characteristics of elastoplastic behaviour shown on the example of a stress-strain curve as defined in (a) and a given loading history in (b). Two different types of plasticity can be differentiated: for one a material exhibiting incremental plasticity in (c) and deformation plasticity in (d), both with hardening.

elasticity equations and is further complemented by a threshold in stresses as well as additional evolution equations in *every* point of the domain. These equations describe constraints on the admissible stresses, bounding them by what is known as yield surface. A further complexity is that this yield surface might change (in shape and position) and evolution equations are necessary to describe the process. This results in partial differential equation in space (basically the same equations as in linear elasticity) and additional ordinary differential equations in any point of the domain. Under certain assumptions the governing equations for plasticity even have a variational structure, which facilitates the numerical treatment.

As mentioned, two distinctive types of plasticity can be found in the literature: the incremental and the total plasticity. The *incremental plasticity* is closer to experimental behaviour and exhibits true path dependence, as seen in Fig. 3(c). The theory is very general and it is necessary to introduce a pseudo-time in order to capture cycling loading and various hardening phenomena. The other type is the *total deformation plasticity*, shown in Fig. 3(d), which for one load step gives the same results as for incremental plasticity. Even so, there is no true history dependence and the model might be considered rather a nonlinear elastic type of material model. Nevertheless,

the model gives a good approximation to true elastoplastic behaviour and from a mathematical and implementational point of view, the total plasticity theory is considerably easier to handle. Note that the model has also a bound on admissible stresses and therefore similar smoothness properties on the displacement field.

We will now work out the basic equations for the continuous formulation of elastoplasticity and point out the differences between incremental and total plasticity.

Additive decomposition of strain tensor We start with the fundamental assumption of elastoplasticity (at least when considering small strains), that is, the *additive* decomposition of strains in an elastic and a plastic part. The total strain tensor is decomposed in

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^{el} + \boldsymbol{\epsilon}^{pl} \tag{2.10}$$

where both $\boldsymbol{\epsilon}^{el}$ and $\boldsymbol{\epsilon}^{pl}$ are defined as symmetric, second order tensors. It is also possible to deduce the additive decomposition from thermodynamical considerations rather than only from observations on an one-dimensional model, see [120].

Elastic response Notice that the total strain is given as the symmetric part of the deformation gradient, $\boldsymbol{\epsilon} := \nabla^s \boldsymbol{u}$. That is, Eq. (2.10) basically defines the elastic part of the strain due to the fact that the plastic part will be determined later using evolution equations. It must be emphasized that the elastic response of a body in terms of the stress tensor $\boldsymbol{\sigma}$ is a function of elastic strain only, and consequently given as

$$\boldsymbol{\sigma} := \mathbb{C} : \boldsymbol{\epsilon}^{el} \tag{2.11}$$

Eqs.(2.10) and (2.11) imply the following

$$\boldsymbol{\sigma} := \mathbb{C} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^{pl}) \tag{2.12}$$

It remains to define an elastic threshold, therefore separating elastic and elastoplastic behaviour and to define the plastic strain tensor, which characterizes the irreversible part of the deformation and is the primary unknown in plasticity theory.

Elastic domain and yield condition The threshold on the stress is mathematically described using the zero level set of a scalar indication function, usually referred as *yield function* or *yield criterion* and further denoted with $f(\boldsymbol{\sigma}, \boldsymbol{q}) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$. The function depends on the current stresses and also on a set of history variables $\boldsymbol{q}$ (except for the case of perfect plasticity). The elastic domain is now defined using this function as

$$\mathbb{E}_{\boldsymbol{\sigma}} := \{\boldsymbol{\sigma} | f(\boldsymbol{\sigma}, \boldsymbol{q}) < 0\} \tag{2.13}$$

The boundary of the elastic domain is just the zero level set of $f(\boldsymbol{\sigma})$ and represents the transition zone between elastic and elastoplastic behaviour, therefore

$$\partial\mathbb{E}_{\boldsymbol{\sigma}} := \{\boldsymbol{\sigma} | f(\boldsymbol{\sigma}, \boldsymbol{q}) = 0\} \tag{2.14}$$

Stresses outside of the hypersurface, which is defined by $f(\boldsymbol{\sigma})$, are by definition *not allowed*.

The yield function itself depends strongly on the phenomenological behaviour and can also undergo changes in shape and position, in dependence on the loading history. The simplest material model is the *von Mises* yield function, also known as J_2-plasticity, and is commonly used for mild steels. The set of admissible stresses, therefore the yield function, is then defined as

$$f(\boldsymbol{\sigma}) := |\mathrm{dev}(\boldsymbol{\sigma})| - \sigma_y \tag{2.15}$$

It is emphasized that this yield function is only a function of the *deviatoric* part of the stress tensor, which corresponds well with experimental results for metals.

Flow rule and hardening laws The key notions of plasticity, the irreversibility of plastic flow and the evolution equations for the internal variables are described by the flow rule and the hardening laws. Using arguments from thermodynamics (maximum of plastic dissipation, see [153]), one can establish the necessity for $\boldsymbol{\epsilon}^{pl}$ being *normal* to the yield surface, resulting in

$$\dot{\boldsymbol{\epsilon}}^{pl}(\boldsymbol{\sigma}, \boldsymbol{q}) := \gamma \boldsymbol{r}_p \quad \text{and} \quad \dot{\boldsymbol{\alpha}}(\boldsymbol{\sigma}, \boldsymbol{q}) := -\gamma \boldsymbol{r}_q \tag{2.16}$$

with $\boldsymbol{r}_p$ and $\boldsymbol{r}_q$ as two flow functions defining the direction of the plastic flow and the type of hardening, respectively. The evolution equation for plastic strain is called *flow rule* whereas the equation defining the evolution of strain-line hardening variables $\dot{\boldsymbol{\alpha}}$ is referred as the *hardening law*. The direction of plastic flow is approximated reasonably well by $\boldsymbol{r}_p := \nabla_\sigma \phi$ for most metallic materials. Note that for brevity reasons we used $\nabla_{\boldsymbol{\sigma}} := \frac{\partial}{\partial \boldsymbol{\sigma}}$. The assumption of *normality*, is also referred to as *associated flow rule*. This hypothesis has also important consequences on the properties of the applied numerical scheme. That is, the plastic strain is in this case *incompressible*, which introduces the well known effect of *volumetric locking* in low-order nodal-based displacement finite element methods. The factor γ is referred to as plastic multiplier due to the fact that γ represents the well known Lagrange multiplier associated with minimization problems. For deformation plasticity the plastic strain is given simply by $\dot{\boldsymbol{\epsilon}}^{pl}(\boldsymbol{\sigma}, \boldsymbol{q}) := c \cdot \nabla_{\boldsymbol{\sigma}} \phi$, with c as an arbitrary *constant* instead of a rate equation in γ. Finally, the evolution equations are complemented by the loading/unloading conditions. Following standard arguments from optimization theory they are given as

$$\gamma \geq 0, \quad f \leq 0, \quad \text{and} \quad \gamma f = 0 \tag{2.17}$$

defining *when* the evolution of plastic strains and internal variables may occur.

We stress a particular fact, that is crucial for the numerical treatment using higher-order finite elements: the additive decomposition of the strain may induce weak or even strong discontinuities into the displacement field. The presence of weak discontinuities is explained with the simultaneously appearance of two different strain fields, $\boldsymbol{\epsilon}^{el}$ and $\boldsymbol{\epsilon}^{pl}$. The integration of these two independent fields yields the displacement field and posses naturally a weak discontinuity or a kink. The presence of strong discontinuities in the displacements (the stresses are still continuous) however, can be due to the presence of slip lines, as theoretically devised by [155]. This fact reduces considerably the performance for standard finite element methods. Some types of problems even lead to stress discontinuities, e.g. as the limiting case of a narrow stress transition zone in a fully plastified beam. Notice that additional difficulties occur if the Dirichlet boundary plas-

tifies, then the boundary condition, $\boldsymbol{u} = \boldsymbol{u}_g$ will in general not be satisfied. Taking into account all the difficulties associated with the discontinuities it is clear that a more advanced mathematical setting of the appropriate solution spaces is required.

One point, that is not often mentioned, is that plasticity represents mathematically a variational inequality as e.g. incompressibility or contact problems. The consequence is that a more descriptive interpretation of plasticity might be that plasticity represents a free boundary value problem, with the plastic zone transmitting though the elastic zone. This interpretation makes the elastoplastic boundary value problem accessible to more advanced solution methods as discussed in [23]. In fact, the presented approach transforms any type of constraint problem, as for example on the stress or strain field, into a shape equilibrium problem. However, to be able to apply the approach to plasticity problems, especially for higher-order elements, some additional considerations are necessary. For one there is the inherent path dependency of the problem, the other challenge is an accurate spatial description of the elastoplastic interface based on information given only in integration points. We will pursue this topic further in Section 6.

2.2 Continuous variational formulation

Solving the introduced systems of equations in an analytical form is most of the time an impossible task. For one, sometimes exact solutions may not exist for particular problems e.g. due to an involved geometry and complicated boundary conditions. For the other, the requirements on the smoothness of the solutions are often quite strong (e.g. as shown for Laplace equation or the elasticity problem $u(\boldsymbol{x}) \in C^2(\Omega)$). In order to deal with these problems efficiently, it is necessary to relax the smoothness requirements and to find an easier way to deal with the boundary conditions. This is achieved by introduction of the so-called *weak formulation* or *variational formulation*. We restrict ourselves to a formulation where the solely unknown is the primary variable, as e.g. temperature in heat transfer respectively the displacement field for solid mechanics problems. However, note that for some application fields, in solid mechanics in particular J_2-plasticity, it is recommended to utilize mixed methods to prevent the notoriously difficult locking effects, see [31, 41]. For a detour to mixed methods, including the mathematical setting, implementational issues and numerous applications, see [21, 177]. We notice that another remedy to the locking effects is found in using higher-order approximation spaces, see [166].

To arrive at the weak formulation it is necessary to introduce the concept of *trial functions* and *test functions*. The trial functions $u(\boldsymbol{x})$ represent admissible solutions and are defined as an element of some vector space $\mathcal{S}$, the test functions are further on denoted by $w(\boldsymbol{x})$ and are elements of some vector space $\mathcal{V}$. The requirements on $\mathcal{S}$ and $\mathcal{V}$ will be specified when needed. We start with the Poisson equation as given in Eq. (2.2) and some boundary conditions as in Eq. (2.3) and (2.4). In the first step both sides of the equation of the are multiplied by the test function and integrated over the posed domain. This results in

$$-\int_\Omega \nabla^2 u(\boldsymbol{x})\, w(\boldsymbol{x}) = \int_\Omega f(\boldsymbol{x})\, w. \tag{2.18}$$

Given a specific differential equation, the idea is to "test" whether the local statement is also valid in an integral sense. Note that both formulations, the local and the integral, are so far completely equivalent. Assuming that all included functions are smooth enough, the next step

includes application of Green's identity (the multidimensional analogy to partial integration) to the left-hand side of Eq. (2.18) which gives

$$-\int_\Omega \nabla^2 u\, w = \int_\Omega \nabla u \cdot \nabla w - \int_\Omega \nabla \cdot (\nabla u\, w). \tag{2.19}$$

Note that from here on, for the sake of simplicity, the argument of spatial dependence has been omitted and is implicitly assumed for all involved functions. One derivative was shifted from u to w and an additional term appears containing mixed derivatives. Invoking the divergence theorem to this additional term, the flux on the boundary can be related to the domain terms, resulting in

$$\int_\Omega \nabla \cdot (\nabla u\, w) = -\int_{\partial\Omega} \nabla_{\boldsymbol{n}} u\, w \tag{2.20}$$

Assuming that the test function w is non-homogeneous on the Dirichlet boundary, this integral can be split on parts on the Dirichlet and Neumann boundary, giving

$$\int_{\partial\Omega} \nabla_{\boldsymbol{n}} u\, w = \int_{\Gamma_N} \nabla_{\boldsymbol{n}} u\, w + \int_{\Gamma_D} \nabla_{\boldsymbol{n}} u\, w. \tag{2.21}$$

We simplify the problem demanding that the vector space $\mathcal{V}$ contains only functions with a vanishing trace on the Dirichlet boundary, $w|_{\Gamma_D} = 0$, therefore the second term on the right hand side vanishes. The remaining part is actually the weak formulation of the traction boundary condition. We see, that the continuity requirement for u dropped globally from $u \in C^2(\Omega)$ to $u \in C^1(\Omega)$. Summing up all terms gives rise to the weak formulation of the Poisson problem

$$\int_\Omega \nabla u \cdot \nabla w = \int_\Omega f\, w + \int_{\Gamma_N} t\, w \tag{2.22}$$

for all suitable test functions w from a vector space $\mathcal{V}$. The left hand side of Eq. (2.22) is in mathematical notation referred to as a bi-linear form, the right hand side as linear form. Note that, as was the actual intention, the requirements on the regularity of the function u are significantly reduced compared to the strong formulation. It can be shown that the solutions to practical problems might be even less regular, possibly having kinks or jumps, even singular points. As a consequence, the solution must be sought in other spaces then the spaces of continuous functions $C^m(\Omega)$, containing more general functions and allowing to capture this non-smooth features.

These spaces form the backbone of modern functional analysis and are of paramount importance for the theoretical treatment of partial differential equations. The functions in these spaces are linear and infinite dimensional. The key point is that the functions are defined only in an *integral* sense, therefore allowing for non-smooth functions. We define the space of *square-integrable* functions $L^2(\Omega)$, named after the French mathematician H. Lebesgue as

$$L^2(\Omega) := \left\{ v : \Omega \to \mathbb{R} \Big| \int_\Omega v^2 < \infty \right\}.$$

Other spaces that are of interest, especially in the realm of solid mechanics, are the following *Sobolev-spaces* with the function v and all spatial derivatives of order m being continuously differentiable and elements of $L^2(\Omega)$, resulting in

$$H^m(\Omega) := \left\{ v : \Omega \to \mathbb{R} \middle| v \in L_2(\Omega), \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, ... \in L_2(\Omega) \right\}.$$

In solid mechanics the space $H^1(\Omega)$ is also referred to as *space with finite energy* as the product of the derivatives, which is essentially the strain energy, must be bounded. Instead of requiring the functions to be continuous, rather only the square of the functions is desired to be integrable. In short, these function spaces are defined in a way such that only functions that are well behaved are included. To simplify the treatment of boundary conditions, the following spaces are defined

$$H^1_D(\Omega) := \{ v : \Omega \to \mathbb{R} \Big| v \in H^1(\Omega); v = g \text{ on } \Gamma_D \}.$$

This is a Sobolev space satisfying the Dirichlet boundary conditions on Γ_D, and

$$H^1_0(\Omega) := \{ v : \Omega \to \mathbb{R} \Big| v \in H^1(\Omega); v = 0 \text{ on } \Gamma_D \}$$

is similar to the previous one, with the difference that the trace of the elements vanish on the Dirichlet boundary. For the sake of completeness we mention, that the regular Sobolev spaces are sufficient as long as the solution is smooth enough, which is the case for Laplace- and elasticity problems. For plasticity problems, however, especially for perfect plasticity, this is no longer applicable. The possible weak and strong discontinuities demand for less restrictive spaces, and they are given by the *space of bounded deformations*, defined as

$$\mathbf{BD}(\Omega) := \{ v : \Omega \to \mathbb{R} \Big| v \in L^1(\Omega); \nabla^s v \in M^1(\Omega) \}$$

$M^1(\Omega)$ is defined as the dual space of $C^0(\Omega)$, a so-called space of Radon measures. For the exact definition of the spaces, see e.g. [58]. We mention that existence and uniqueness proofs as well as regularity results are available for a few selected cases. In particular, it can be shown that for perfect plasticity the equations loose the coercivity property and therefore the uniqueness of the solution is lost.

With the definition of the proper spaces we can specify the sought collection of functions. The trial functions are in case of non-homogeneous Dirichlet boundary conditions elements of $\mathcal{S} := H^1_D(\Omega)$. For our case, however, the trial and the test functions coincide and are elements of $\mathcal{S} = \mathcal{V} := H^1_0(\Omega)$. Therefore we can write Eq. (2.18) as

$$\begin{aligned} &\text{find } u \in \mathcal{S} \quad \text{such that} \\ &-\int_\Omega \nabla u \cdot \nabla w = \int_\Omega f\, w - \int_{\Gamma_N} t\, w \qquad \forall w \in \mathcal{V} \end{aligned}$$

It can easily be shown that the strong form always implies the weak form, the vice versa, however, is valid only under certain regularity conditions on the involved functions, for details see [182]. An alternative way, valid for symmetric bi-linear forms, to derive the weak form is due to a rather physical reasoning. An integral formulation is reached by postulating the existence of an energy potential, e.g. for linear elasticity the potential energy is defined as

$$\mathcal{U}(\boldsymbol{u}) := \frac{1}{2} \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\epsilon}(\boldsymbol{u}) - \int_\Omega \boldsymbol{f} \cdot \boldsymbol{u}. \tag{2.23}$$

The governing equations of the weak form result then as a necessary condition for a stationary point, if fact, the minimum of $\mathcal{U}(\boldsymbol{u})$. This formulation offers a conceptual simplification and in addition easier mathematical proofs of uniqueness and existence, however, the procedure is in general applicable only for conservative problems.

Applying the same scheme as before to the equations of linear elasticity (multiplication by a test function, partial integration, divergence theorem) gives rise to the following system of equations, in the field of structural mechanics also known as *principle of virtual work*

$$\begin{aligned} &\text{find } \boldsymbol{u} \in \mathcal{S} \quad \text{such that} \\ &\int_\Omega 2\mu \nabla^s \boldsymbol{u} \cdot \nabla^s \boldsymbol{w} + \lambda \mathrm{tr}(\nabla^s \boldsymbol{u}) \mathrm{tr}(\nabla^s \boldsymbol{w}) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{w} - \int_{\Gamma_N} \boldsymbol{t} \cdot \boldsymbol{w} \qquad \forall \boldsymbol{w} \in \mathcal{V} \end{aligned} \tag{2.24}$$

$\nabla^s \boldsymbol{u}$ as well as $\boldsymbol{f}$ are both vector valued functions, therefore the test functions also need to be vector-valued. As mentioned before, for conservative problems an alternative formulation is possible. An application of the Gâteaux-derivative, defined as

$$\delta \mathcal{U}[\boldsymbol{u}](\boldsymbol{v}) = \lim_{\varepsilon \to 0} \frac{\mathcal{U}[\boldsymbol{u} + \varepsilon \boldsymbol{v}] - \mathcal{U}[\boldsymbol{u}]}{\varepsilon}$$

yields a stationary point of $\mathcal{U}[\boldsymbol{u}]$ in direction of $\boldsymbol{v}$ and consequently exactly the terms in Eq. 2.24. The following notation for the involved spaces is used

$$[H^1(\Omega)]^{dim} := \left\{ v \in L^2(\Omega) \middle| \frac{\partial v}{\partial x_i} \in L^2(\Omega) \right\}.$$

Taking again into account homogeneous Dirichlet boundary conditions we arrive at the following space for the test functions

$$[H_0^1(\Omega)]^{dim} := \left\{ v \in L_2(\Omega) \middle| \frac{\partial v}{\partial x_i} \in L_2(\Omega); v = 0 \text{ on } \Gamma_D \right\}.$$

Later on, it will be important to measure the "size" of the functions, for this reason some appropriate norms are introduced. So for a function $u(\boldsymbol{x}) \in L^2(\Omega)$, the suitable norms are

$$||u||_{L^2} := \sqrt{\int_\Omega u^2} \qquad \text{and} \qquad ||\boldsymbol{u}||_{L^2} := \sqrt{\int_\Omega \boldsymbol{u} \cdot \boldsymbol{u}} \tag{2.25}$$

for a scalar function $u(\boldsymbol{x})$ and a vector valued function $\boldsymbol{u}(\boldsymbol{x})$, respectively. For functions $u(\boldsymbol{x}) \in H^1(\Omega)$, the suitable norms (in this case rather a semi-norm) are given as

$$|u|_{H^1} := \sqrt{\int_\Omega |\nabla u|^2} \qquad \text{and} \qquad |\boldsymbol{u}|_{H^1} := \sqrt{\int_\Omega |\nabla \boldsymbol{u} \cdot \nabla \boldsymbol{u}|} \tag{2.26}$$

2.3 Discrete formulation

2.3.1 Discrete variational formulation for Laplace and elasticity problems

As seen before, in the continuous formulation the problem has infinitely many parameters (designated as degrees of freedom in solid mechanics). To transform the problem into a discrete

formulation, the procedure by Galerkin suggests to choose suitable finite dimensional subspaces for the trial ($\mathcal{S} \approx \mathcal{S}^h$) and the test functions ($\mathcal{V} \approx \mathcal{V}^h$) and then to search for the "best" solution in these subspaces. The subspaces in the standard Galerkin method are usually build of piecewise polynomials, but also other choices exist.

An ansatz (a trial solution) for the displacement field as well as for the field of test functions is chosen and the fields can be written as a linear combination of basis functions $N_i(\boldsymbol{x})$ and unknown coefficients u_i and w_i,

$$u^h(\boldsymbol{x}) = \sum_{i \in I} N_i(\boldsymbol{x})\, u_i \qquad w^h(\boldsymbol{x}) = \sum_{i \in I} N_i(\boldsymbol{x})\, w_i. \tag{2.27}$$

In doing so, the shape functions fulfill $\{N_i(\boldsymbol{x})\}_{i \in I} \subset \mathcal{V}^h$. The functions $N_i(\boldsymbol{x})$ are the shape functions or ansatz functions. The functions possess a local support and are non-zero only in a small region of Ω. Once this ansatz is inserted in the equations we end up with

$$\begin{aligned} &\text{find } u^h \in \mathcal{S}^h \quad \text{such that} \\ &-\int_\Omega \nabla(N_i(\boldsymbol{x})u^h) \cdot \nabla(N_i(\boldsymbol{x})w^h) = \int_\Omega f\,(N_i(\boldsymbol{x})w^h) - \int_{\Gamma N} t\,(N_i(\boldsymbol{x})w^h) \qquad \forall w^h \in \mathcal{V}^h \end{aligned} \tag{2.28}$$

This procedure results in a system of equations with the n^{th}-row of the matrix corresponding to the n^{th}-test function w_n^h. The test function is associated with the n^{th} node in our discretization and physically represents the "testing" of the governing differential equation (the balance equation) on the support of w_n^h. If the basis functions are linearly independent, this provides N equations for the N unknown coefficients. The left hand side is further on referred to as *stiffness matrix* and the right hand side as *load vector*. One matrix entry is then defined by

$$K_{ij} = \int_\Omega \nabla N_i(\boldsymbol{x}) \cdot \nabla N_j(\boldsymbol{x}) \qquad F_i = \int_\Omega f_i N_i(\boldsymbol{x}) - \int_{\partial\Omega} g_i N_i(\boldsymbol{x}) \tag{2.29}$$

The system of equations is defined by subsequently assembling all entries as

$$\begin{bmatrix} K_{11} & K_{12} & \dots & K_{1m} \\ K_{21} & K_{22} & \dots & K_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ K_{m1} & K_{m2} & \dots & K_{mm} \end{bmatrix} \begin{bmatrix} u_1^h \\ u_2^h \\ \vdots \\ u_m^h \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix}$$

or in short notation

$$\boldsymbol{K}\boldsymbol{u}^h = \boldsymbol{F}_{ext} \tag{2.30}$$

This system of linear equations is named Galerkin system and the solution $\boldsymbol{u}^h$ is referred to as Galerkin solution. Once this system of equations is solved, the value of the unknowns can be interpolated in any part of the domain using Eq. (2.27). Note that the trial and the test functions were from the same vector space, resulting in a symmetric and positive definite system matrix. Furthermore, due to the local support of the ansatz, the matrix is sparse, therefore facilitating numerical handling. The shape functions $N_i(\boldsymbol{x})$ are usually not defined in terms of global coor-

dinates $\boldsymbol{x}$, but rather in terms of local coordinates $\boldsymbol{r}$. The idea is particularly suitable to define element nodes (especially for higher-order elements) and to facilitate numerical integration. For each element type a suitable reference domain is defined where the shape functions are constructed. The discrete handling of elasticity is analogous to the handling of the Laplace problem, and results in a similar system of equations as in Eq. (2.29).

2.3.2 Discrete variational formulation for elastoplasticity problem

Compared to the pure elastic problem, for elastoplastic problems the solution procedure becomes far more intricate. The domain discretization is still done using regular finite elements and is similar to the formulation for elastic problems, however, some additional algorithmic considerations are necessary. As mentioned before, in order to treat plasticity problems it is inevitable to introduce a time-like variable defining the load level. Assuming we are interested in the solution over the interval of interest λ, usually $\lambda := [0, 1]$, this interval is divided into sub-intervals

$$\lambda = \bigcup_{n=0}^{m} [t_n, t_{n+1}].$$

Knowing all variables at a time t_n the task is to find the solution for the next time step t_{n+1}. Notice that this corresponds to the external load $\boldsymbol{F}_{ext}(t_n)$ for which the solution is known and $\boldsymbol{F}_{ext}(t_{n+1})$ for which the solution is sought. Rewriting the weak form in Eq. (2.24) in a slightly different way as

$$\begin{aligned} &\text{find } \boldsymbol{u} \in \mathcal{S} \quad \text{such that} \\ &\int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \nabla^s \boldsymbol{w} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{w} - \int_{\Gamma_D} \boldsymbol{t} \cdot \boldsymbol{w} \qquad \forall \boldsymbol{w} \in \mathcal{V} \end{aligned} \tag{2.31}$$

Considering that the expression $\boldsymbol{\sigma}(\boldsymbol{u})$ is highly nonlinear due to the constraints on the admissible stresses, we end up with a nonlinear weak form (nonlinear in $\boldsymbol{u}$). Rewriting the discrete weak form using an analogy from structural mechanics and interpreting the left hand side of Eq. (2.31) as the internal forces that are equivalent with the stresses in an integral sense, therefore $\boldsymbol{F}_{int} := \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \nabla N$. Then, an incremental load approach gives

$$\boldsymbol{F}_{int}(\boldsymbol{u}_{n+1}) = \boldsymbol{F}_{ext}(t_{n+1})$$

with the displacement $\boldsymbol{u}_{n+1}$ corresponding to the load level t_{n+1}. The residual vector $\boldsymbol{R}(\boldsymbol{u}_{n+1})$, with the entries as the unbalanced forces is introduced. This vector can be interpreted as the forces preventing the structure from being in equilibrium as

$$\boldsymbol{R}(\boldsymbol{u}_{n+1}) := \boldsymbol{F}_{int}(\boldsymbol{u}_{n+1}) - \boldsymbol{F}_{ext}(t_{n+1}) \tag{2.32}$$

The solution of the discrete form of Eq. (2.31) is equivalent to finding the root of Eq. (2.32). Standard solution methods for nonlinear equation systems can be applied, the most widespread being the Newton-Raphson procedure. This amounts to

$$\boldsymbol{R}(\boldsymbol{u}^j_{n+1}) + \nabla_{\boldsymbol{u}} \boldsymbol{R} \, \Delta \boldsymbol{u} = \boldsymbol{0}$$

followed by the update of the unknown displacement field with

$$\boldsymbol{u}_{n+1}^{j+1} = \boldsymbol{u}_{n+1}^{j+1} + \Delta\boldsymbol{u}.$$

This iteration procedure is continued until some norm of the residual vector is smaller than some user defined tolerance. The linearization of the system matrix $\nabla_{\boldsymbol{u}}\boldsymbol{R}$ is referred as tangent stiffness matrix.

Remark 1 It is noted that the convergence of the Newton-Raphson procedure is guaranteed only provided that the initial guess is in a close neighborhood to the solution. Especially for solutions with non-smooth properties as elastoplasticity, or problems involving contact conditions, the solution may diverge and it is necessary to control the way the solution is sought. For elastoplasticity problems that are close to system failure, it is mandatory to have e.g. a line search procedure to control the growth of the residual vector.

Remark 2 In order to get optimal quadratic convergence of the Newton-Raphson method in elastoplasticity, some additional considerations regarding the tangent stiffness matrix are necessary. That is, the linearization of the tangent stiffness $\nabla_{\boldsymbol{u}}\boldsymbol{R}$ must be realized when the discretization in space *and* time is done, otherwise sub-optimal convergence rates are the consequence.

2.3.3 Discrete formulation of local constitutive equations

In addition to the discretization in space, which is done via finite elements and resulting in a global (nonlinear) system of equations, it is necessary to transfer the (local) constitutive formulation in a discrete one. A widespread scheme to handle the local time stepping procedure is due to [153] and referred to as *return mapping algorithm,* given the striking geometric interpretation of the procedure. The algorithm is an unconditionally stable backward difference in time (backward-Euler) resulting in an algebraic set of equations yielding the sought variables.

First, it is assumed that no plastic flow occurs during the current time step (which represents the trial state), meaning $\boldsymbol{\varepsilon}_{n+1}^{pl,tr} = \boldsymbol{\varepsilon}_{n}^{pl}$. The known initial data at t_n are $\boldsymbol{\varepsilon}_n, \boldsymbol{\varepsilon}_n^{pl}$ and $\boldsymbol{\varepsilon}_{n+1}$. The trial quantities for the algorithm are then given as

$$\begin{aligned}
\mathrm{dev}[\boldsymbol{\sigma}_{n+1}^{tr}] &= 2\mu(\mathrm{dev}[\boldsymbol{\varepsilon}_{n+1}] - \boldsymbol{\varepsilon}_n^{pl}) \\
f_{n+1}^{tr} &= ||\mathrm{dev}[\boldsymbol{\sigma}_{n+1}^{tr}]|| - \frac{2}{3}\sigma_y \\
\boldsymbol{n}_{n+1}^{tr} &= \frac{\mathrm{dev}[\boldsymbol{\sigma}_{n+1}^{tr}]}{||\mathrm{dev}[\boldsymbol{\sigma}_{n+1}^{tr}]||}
\end{aligned}$$

Using f_{n+1}^{tr}, it is evaluated whether a stress correction is necessary. If $f_{n+1}^{tr} \leq 0$, all state variables, that is, the stresses, strains and history variables are updated using the trial values, otherwise a correction (along the normal vector to the yield surface) is necessary. This results in the following equations for the update of the state variables

$$\begin{aligned}
\Delta\gamma &= \frac{f^{tr}_{n+1}}{2\mu} \\
\mathrm{dev}[\boldsymbol{\sigma}_{n+1}] &= \mathrm{dev}[\boldsymbol{\sigma}^{tr}_{n+1}] - 2\mu\Delta\gamma\boldsymbol{n}^{tr}_{n+1} \\
\boldsymbol{\varepsilon}^{pl}_{n+1} &= \boldsymbol{\varepsilon}^{pl}_{n} - \Delta\gamma\boldsymbol{n}^{tr}_{n+1} \\
\boldsymbol{\sigma}^{tr}_{n+1} &= \mathrm{dev}[\boldsymbol{\sigma}_{n+1}] + \kappa\mathrm{tr}(\boldsymbol{\varepsilon}_{n+1})\mathbf{I}
\end{aligned}$$

The bulk modulus κ is used to describe dilatational deformations, and is defined as

$$\kappa = \frac{E}{3(1-2\nu)}.$$

2.4 The finite element

According to the commonly used formal definition of a standard finite element due to P. Ciarlet [37], any finite element is composed by the triplet $[\Omega_{\boldsymbol{r}}, P_\Omega, \Sigma_\Omega]$. In detail, the components are defined as

Domain [$\Omega_{\boldsymbol{x}}$] Defined as a closed polygon in $\mathbb{R}^{dim}$. Depending on the spatial dimension, the element can take various shapes. At this point we introduce two different domains (or elements) which are linked by a mapping $\Omega_{\boldsymbol{r}} = \phi(\Omega_{\boldsymbol{x}})$. For one, there is the real "physical" element, defined in the coordinate system $\mathbf{x}$ with possibly curved edges or faces. For another, a reference or master element is introduced, with the corresponding coordinate system defined by $\boldsymbol{r}$. Notice that for all elements in the physical domain only one master element exists and only the mapping changes. The importance of this concept, physical and master element, can not be overestimated and is crucial for element-by-element assembling, numerical integration, definition of nodal points, etc. For each element class the reference domain is defined in a classified range, see Figs. 4, 5 and 6, which is by no means unique. The choice of the reference domain is not unique, it should support a simple numerical integration and facilitate the definition of the shape functions. In an one-dimensional setting, the obvious choice of reference domain is depicted in Fig. 4. Note that the reference domain is defined between the bounds of $r \in [-1, 1]$.

(a) 1D-domain in r-coordinate system (b) 1D-domain in x-coordinate system

Figure 4: Reference and real domain for 1D-elements.

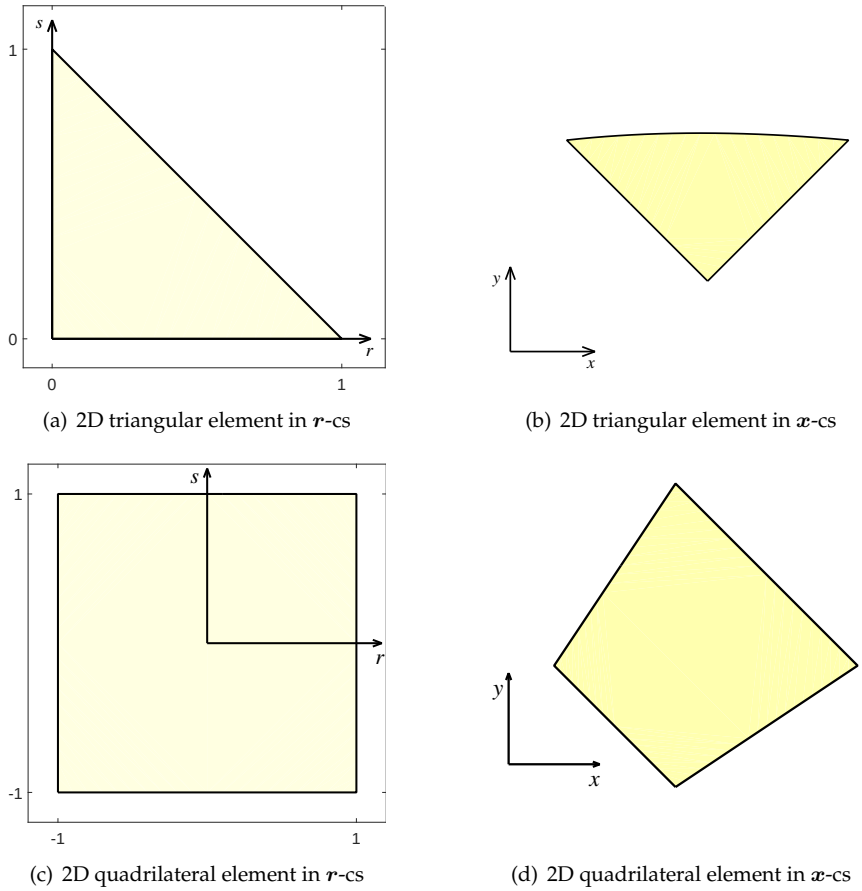
The only thing left is to define the mapping ϕ, which is conveniently done using the same class of functions as used for the approximation of unknowns in Eq. (2.27). That is, the approximated spatial coordinate x^h is then given as a function of reference coordinates r as

$$x^h = \phi(x) := \sum_{i \in I} N_i(r)\, x_i, \tag{2.33}$$

with x_i as the global coordinates of the nodal point i, corresponding to the shape function N_i. This procedure is referred to as *isoparametric concept* and was introduced by B. Irons in 1968, see [103]. Especially for curved domains this approach introduces a discretization error. Nevertheless the error decreases with the element size and the approach is nowadays an absolute standard procedure. The advantages of this formulation are obvious, e.g. increased flexibility of the elements as well as guaranteed compatibility between the elements. Furthermore, it tremendously simplifies the implementation of the finite element method.

In two dimensions there are two common classes of elements; the triangular and quadrilateral elements. The isoparametric mapping is then the vectorized version of Eq. (2.33) and given as

$$\boldsymbol{x}^h = \boldsymbol{\phi}(\boldsymbol{x}) := \sum_{i \in I} N_i(\boldsymbol{r})\, \boldsymbol{x}_i. \tag{2.34}$$



(a) 2D triangular element in $\boldsymbol{r}$-cs

(b) 2D triangular element in $\boldsymbol{x}$-cs

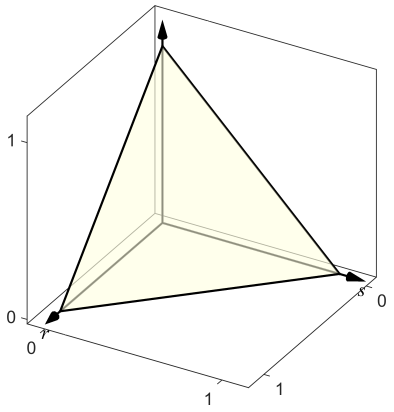
(c) 2D quadrilateral element in $\boldsymbol{r}$-cs

(d) 2D quadrilateral element in $\boldsymbol{x}$-cs

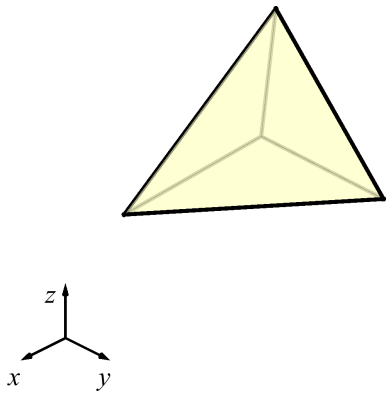
Figure 5: Reference and physical domains in two dimensions, for triangular (a),(b) and quadrilateral (c),(d) elements, respectively. Note that the reference domain for triangles is defined differently as that for quadrilaterals.

The mapping in Eq. (2.34) is in general nonlinear, it remains linear only in case of affine mappings. However, in order to guarantee that all elements in the physical domain are

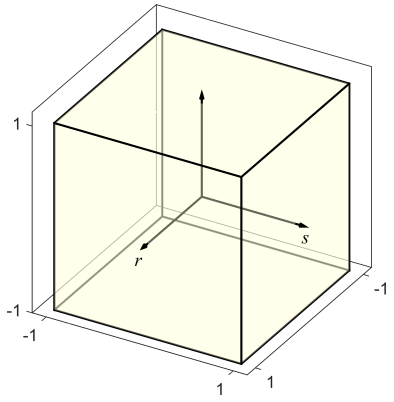
geometrically valid, it has to be ensured that the mapping ϕ remains unique. This is done by means of the Jacobian matrix, we will revisit the topic later when dealing with transformation of derivatives.
In three dimensions the following elements are used: the tetrahedron as the 3D-version of the triangle, the hexahedron or brick as a generalization of a quadrilateral element and the prism as a tensor product of a triangular and a quadrilateral element. The mapping in three dimensions remains the same as in the two-dimensional case, see Eq. (2.34).



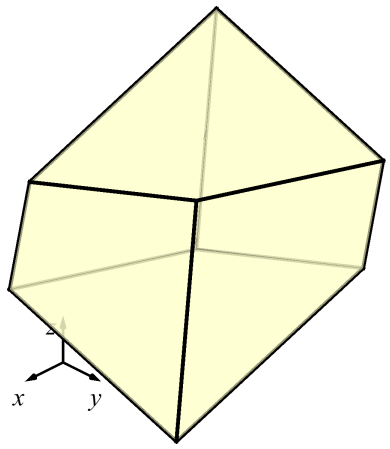
(a) Tetrahedral element in r-cs



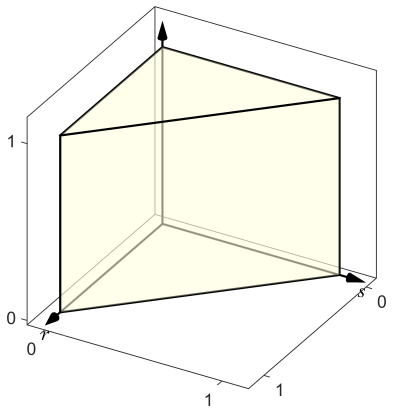
(b) Tetrahedral element in x-cs



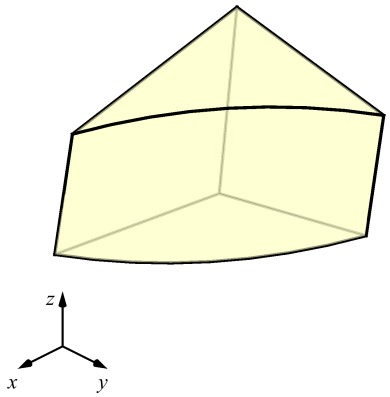
(c) Hexahedral element in r-cs



(d) Hexahedral element in x-cs



(e) Prism element in r-cs



(f) Prism element in x-cs

Figure 6: Reference and physical domains for 3D-elements, for tetrahedral (a),(b), hexahedral (c),(d) and prismatic (e),(f) element types.

Referring to Eq. (2.28), notice that the weak form is defined in terms of the coordinate system $\boldsymbol{x}$. In order to use the reference elements defined in $\boldsymbol{r}$-coordinates, it is necessary to define some transformations between the two coordinate systems. Two possibilities exist: to relate everything to the $\boldsymbol{x}$-coordinates and to work in the physical element or to work in

the $\boldsymbol{r}$-coordinates, then it is necessary to pull all variables back to the reference element
If the function values itself are transformed from $\boldsymbol{r} \rightarrow \boldsymbol{x}$, e.g. the approximate solution u^h, this is particularly simple as

$$u^h(\boldsymbol{r}) = u^h(\boldsymbol{x}(\boldsymbol{r})) \tag{2.35}$$

Obviously, shape function derivatives with respect to the global coordinates are present in the weak form. As a consequence, when working in another coordinate system, it is also necessary to transform the function derivatives. Note that now $u^h(\boldsymbol{r})$ is given and we are interested in $\nabla_{\boldsymbol{x}} u^h(\boldsymbol{r})$, therefore application of the standard chain rule and some algebraic manipulations yield

$$\nabla_{\boldsymbol{r}} u^h(\boldsymbol{r}) = \nabla_{\boldsymbol{r}} \boldsymbol{x} \, \nabla_{\boldsymbol{x}} u^h. \tag{2.36}$$

Using the definition of the *Jacobian matrix* as $\nabla_{\boldsymbol{r}} \boldsymbol{x} := \boldsymbol{J}$ yields the following equation

$$\nabla_{\boldsymbol{x}} u^h = \boldsymbol{J}^{-1} \, \nabla_{\boldsymbol{r}} u^h(\boldsymbol{r}) \tag{2.37}$$

which requires that the inverse of $\boldsymbol{J}$ exists. This inverse exists, provided the mapping ϕ is one-to-one between $\boldsymbol{r}$ and $\boldsymbol{x}$. And this can be checked easily using the determinant of the Jacobian matrix (called *Jacobian determinant* or simply *Jacobian),* which is given as

$$J := det[\boldsymbol{J}]. \tag{2.38}$$

A strictly positive Jacobian over Ω implies uniqueness of the mapping ϕ.

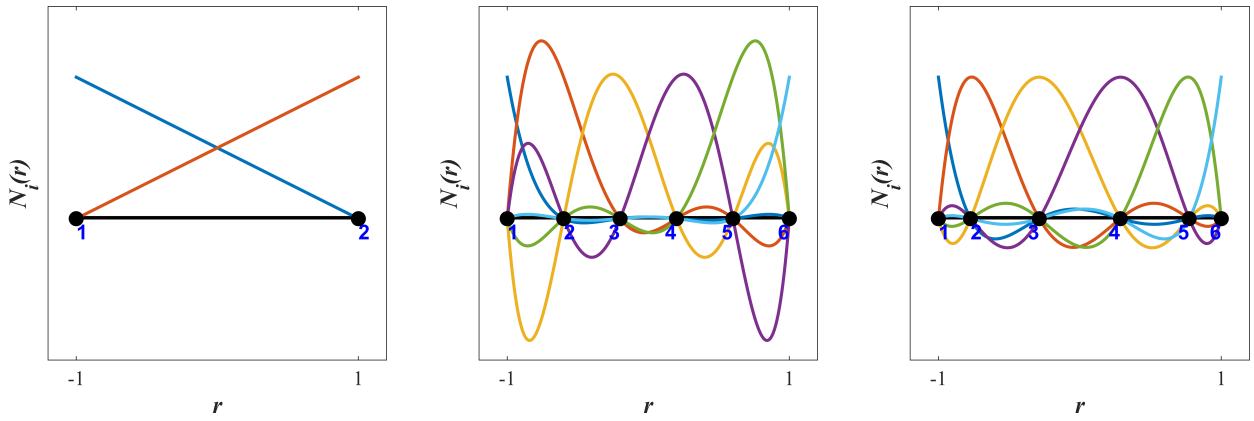
A polynomial space [P_Ω] The choice of a particular class of shape functions includes the definition of a basis for the problem. We focus here on nodal based elements, however, other choices are possible. A family of elements is known as *Lagrange-type* or *nodal* if the following is fulfilled

$$N_i(\boldsymbol{r}_j) = \delta_{ij}. \tag{2.39}$$

Hereby $\boldsymbol{x}_j$ are the coordinates of the j^{th}- node in the reference element. This characteristic is referred to as Kronecker-delta property. In fact, this feature enables a direct correlation of physical values to the coordinates and is paramount for the implementation.
Employing the Lagrange interpolation polynomial yields the explicit expression for 1D-polynomials defined as

$$N_i(r) = \prod_{1 \leq j \leq i+1, i \neq j} \frac{r - x_j}{x_i - x_j} \tag{2.40}$$

Unfortunately, the Lagrange polynomial suffers from one serious drawback, which can be seen in Fig. 7. Especially for polynomials of high degree oscillations at the edges of the interval appear. For this reason, Lagrange polynomials are used rather for modest orders up to 5 and for higher orders special functions are useful. Then, the so-called Chebychev polynomials or Legendre polynomials are used, the method is referred as *spectral element method*. But it is also possible to use Lagrange polynomials with non-uniformly spaced nodes to overcome this problem.
Note that the polynomials in real space are typically rational functions, depending on the deformation of the element. The magnitude of the deformation is hereby given by the



(a) 1D-domain in r-coordinate system (b) 1D-domain in x-coordinate system (c) 1D-domain in x-coordinate system

Figure 7: Shape functions for $m = 1$ in (a) and for $m = 5$ with an equidistant nodal distribution in (b) and Chen-Babuška nodal distribution for $m = 5$.

Jacobian as defined in Eq. (2.38).

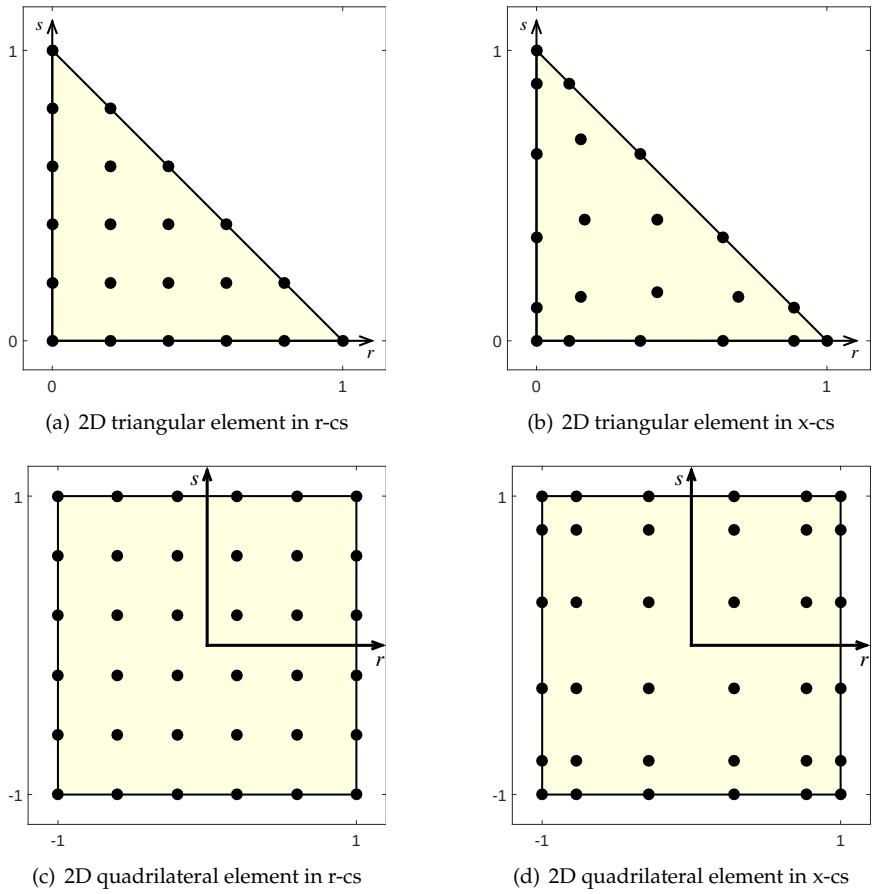
A set of functionals [Σ_Ω] The functionals act hereby on the polynomials. Usually, in the standard finite element formulations, these are only function evaluations in certain points even though other possibilities exist, e.g. normal derivatives. Notice that this step actually defines the degrees of freedom. Clearly, the choice of points, that is, the location of the nodes, influences the accuracy of the procedure. For further developments we focus on the relatively widely used equidistant point distribution. A theoretical optimum is given by the Chen-Babuška distribution of the nodes, see [32]. The advantages include better interpolation properties, therefore a smaller error, and also a reduction of Runge´s phenomenon. A comparison of an equispaced nodal distribution and the Chen-Babuška location for triangular and quadrilateral elements of polynomial order $m = 5$ is depicted in Fig. 8.

Note that now the continuity of the solution function $u(\boldsymbol{x})$ is locally $C^\infty(\Omega^{el})$ but globally in general only $C^0(\Omega)$. However, we conclude that weak discontinuities (as kinks in the displacement field) can be represented naturally with the standard finite element method, as long as the interface is just on the element boundary.

2.5 Spatial discretization

The finite element method is a numerical method and is therefore by nature an approximation method, its accuracy depends on several important aspects. Among them are the smoothness of the solution field, the choice of the finite element, the applied numerical integration scheme and the spatial discretization. The construction of suitable, *fully automatic*, spatial discretizations, further on referred as meshes, is still an open topic, in industry as well as in academia. In fact, the spatial subdivision of complex, possibly curved domains in elements suitable for analyses remains a bottleneck in the analysis. In 1992 Joe F. Thompson, one of the pioneers of grid generation techniques, wrote in [170]

At present it can take orders of magnitude more man-hours to construct the grid



(a) 2D triangular element in r-cs

(b) 2D triangular element in x-cs

(c) 2D quadrilateral element in r-cs

(d) 2D quadrilateral element in x-cs

Figure 8: Reference domains for 2D-elements, for an equidistant distribution of nodes in (a),(c) and the optimal distribution according to Chen-Babuška in (b),(d).

> than it does to perform and analyze the PDE solution on the grid. This is especially true now that PDE codes of wide applicability are becoming available, and grid generation has been cited repeatedly as being a major pacing item. The PDE codes now available typically require much less esoteric expertise of the knowledgeable user than do the grid generation codes.
>
> ...[T]he need for a highly efficient and easily used grid generation system is obvious to everyone ...

This statement is nowadays as true as it was in the 1990s. In between, a lot of research was conducted on mesh generation, see e.g. [67, 180, 118] and the references therein. Other related approaches to avoid (or at least facilitate) mesh generation include i.a. the concept of isogeometric analysis [44] and the large spectrum of embedded domain methods. The number of publications dealing with this topic is still increasing, which shows conclusively that there is still a demand for robust and fully automated meshing procedures.

We consider now the case that the physical domain Ω is approximated well enough by the computational domain Ω^h. The main idea is to cover the domain with a finite number of cells with simple shapes, such that the integrals defined in Ω^h are decomposed into a sum of integrals

defined over Ω^{el}. We can hereby choose any of the elements described in the last section, while meeting some constraints on their sizes and shapes.

The union of the non-overlapping elements is the computational mesh mesh $\mathcal{T}^h$, with the properties

$$\Omega^h = \bigcup_{i=1}^{N_{el}} K_i \qquad \text{and} \qquad \text{int}(K_m) \cap \text{int}(K_j) = \emptyset \quad \text{for} \quad m \neq j$$

The tessellation (in case of triangular elements sometimes also called triangulation) is consistent, if the intersection of any two elements is either empty, a common vertex, a common edge or a common face.

A rough classification of mesh types can be done according to various criteria, widely used are categorizations based on.

Conformality When the faces and the edges of neighboring elements fit perfectly together, continuity is automatically fulfilled and the discretization is referred as conformal; otherwise the discretization is referred to as *non-conformal*. Non-conformal meshes have usually additional nodes existing only on one element but not on the other, called *hanging nodes*.

Topology The most basic classification of meshes is based upon the connectivity of the mesh. If the connectivity is regular, such that for any element and any vertex the adjacent vertices are known implicitly and can be determined based on simple arithmetic, the mesh is categorized as *structured*. If this is not the case, it is necessary to explicitly store the connectivity and in this case the mesh is referred as *unstructured*.

Alignment If the mesh matches the body boundary (line in 2D, surface in 3D) approximately, the mesh is referred as *body-fitted*. If the faces or the edges of the mesh cross the boundaries, the mesh is denoted as *non-bodyfitted* or *non-aligned*.

The literature on mesh generation is vast and various different techniques have been proposed over the last decades. Considering an one-dimensional grid, the discretization is straightforward, at least compared to the 2D and 3D counterparts. The only fact to consider is the *spatial resolution* of the elements and the mesh generation can be done by defining (either manually or adaptively) the boundary nodes and utilizing the isoparametric mapping to create internal nodes.

For higher dimensions, in addition to the spatial resolution also the *shape* of the elements plays an important role. The early structured mesh generation methods (around the 1970s) presumed simple and structured geometries, therefore the first mesh generators worked by constructing global mappings. Among them are the *elliptic grid generators* (solve differential equations to transform a Cartesian mesh in parametric space to a structured mesh in physical space, e.g. a Laplace equation), *isoparametric mappings* or utilization of the interpolation technique as introduced by Gordon and Hall (denoted as *transfinite mapping*) and combinations of them. The clear drawback to this approach is that to be able to apply the technique, it is necessary that the whole domain can be mapped in the parametric space. However, over the years no fully satisfying automatic mesh generator, based on global mappings was constructed, see also the references in [180].

Having only straight edges in a domain, the unstructured mesh generation is nowadays (at least in 2D) straightforward. Two standard approaches for simplex elements are either the *Delaunay triangulation* or the *advancing front method* (or a combination of the two methods).

Delauney triangulation algorithm. This procedure was invented in the 1970s and investigated for a long time, especially in the community of computational geometry. The starting point of the Delauney triangulation is an *existing* point cloud. It must be emphasized, that the Delauney method provides no means to include new points, but only provides an algorithm to form the topology of the domain. A triangle in a mesh is known as Delauney element, if the circumcircle associated with the nodes of the element is empty, therefore contains no other points. A simple approach to construct a triangulation is based on the dual graph of the Delauney triangulation, the Voronoi-diagram. The Delauney triangulation is finally the dual graph of the Voronoi-diagram. However, there are also iterative procedures, avoiding the need to construct Voronoi cells. Once the initial mesh is constructed, either edge swapping or alternatively point insertion is applied in order to create a valid Delauney triangulation. Usually also mesh modification is necessary until the element qualities are sufficient. From a quality point of view, the elements close to the boundary are usually the worst, and the elements in the interior of the meshed domain the best. The procedure works in a hierarchical fashion, by meshing points, curves, surfaces and finally the volumes. The method is especially suitable for 2D-problems, as it has the striking advantage of maximizing the minimum angles of the triangles. This is not any more true for 3D, but even there several geometrical criteria are optimized.

Advancing front method. The method works for simplex as well as rectangular elements equally. It uses an existing boundary (therefore a boundary mesh is necessary) and creates a mesh along a front moving into the computational domain. The difference to the Delauney triangulation is that the elements are constructed one by one, therefore no (global) mesh exists until the process is completed. The emerging mesh is usually characterized by high quality elements near the boundary and rather ill-shaped elements (especially in $3D$) where the element front coincides. Similar to the Delaunay triangulation the approach works dimensionally hierarchical, therefore, from points up to volumes. The critical points are the choice of the face which is used as initial point of advance and the choice of new (or existing) vertices.

In general, the Delauney algorithm is much faster than the advancing front method, however, the problem remains that the final mesh does not align with the domain boundaries. Both methods work quite well for straight domain boundaries, when dealing with curved boundaries additional considerations need to be made. In order to be able to benefit from a higher-order discretization, it is inevitable to take into account eventually occurring curved boundaries.

For curvilinear meshes a different approach compared to linear meshes is necessary: Usually one of these two methods is applied for the generation of curved high-order meshes.

- Direct methods: One can either try to create directly a valid high-order mesh through a high-order version of a classical meshing algorithm. For this, it is necessary to describe very accurately the surface of the computational domain. This can be done either by using analytical descriptions, as now used in IGA (Splines, B-Splines, NURBS) or by generating point clouds and triangulating them. The advantage of the approach is the a priori validity of the resulting mesh. This surface triangulation however, is not trivial, and has been intensive subject of research for a while.

- Indirect methods: As a first step a straight-sided (linear) mesh is generated that approximates the domain well enough. The order of the elements is increased and subsequently the elements are curved to fit the boundary. And usually also some regularization procedure is applied, if invalid elements are found. The problem with the approach is that the higher-order curved mesh might become invalid due to an overlap of the new elements or due to an unacceptable degeneration of the elements, resulting in too high variations of the Jacobian inside the regarded element.

Both approaches are actually related, as they share the necessity of detecting and repairing the invalid parts of the mesh. At some point it becomes relevant to improve the mesh quality as abrupt variations in element shape or size might cause problems. This improvement can be done using mesh quality modifications or mesh smoothing methods.

Mesh smoothing methods. In general good quality meshes can not be obtained directly from the meshing generators. A post processing step is often necessary in order to optimize the mesh with respect to element size or (more often) the element shape. The methods include e.g. Laplacian smoothing (therefore solve a PDE), functional optimization, selective mesh movement or diagonal swapping.

2.6 Conditioning of system matrices

A system of equations is termed *ill-conditioned*, if small perturbations in the entries of the system matrix (in our case the stiffness matrix $\mathbf{K}$) or in the coefficients of the right hand side (the load vector $\mathbf{f}$) lead to significant changes of the solution vector. The condition number of a symmetric system matrix $\boldsymbol{K}$ is defined as

$$\kappa(\boldsymbol{K}) := \frac{\lambda_{max}(\boldsymbol{K})}{\lambda_{min}(\boldsymbol{K})} \tag{2.41}$$

with the minimal and maximal eigenvalues λ_{min} and λ_{max}, respectively. A system of equations with a very high condition number is perceptive to numerical errors. In practice, the condition number is important for two different reasons:

(i) For very large and sparse systems the most convenient way to solve the system of equations is to use iterative methods. In this case the condition number is directly correlated with the robustness and the efficiency of the solver.

(ii) Even when using direct methods to solve the system of equations, the condition number directly effects the accuracy of the solution. In this case the negative influence of a high condition number might prevail over the benefits of higher-order discretizations.

However, assuming that the system matrix is a stiffness matrix, we can give some physical interpretations to the occurring phenomena. The condition number depends on multiple factors, among other things on the BVP, the spatial discretization (size and shape of elements) and the choice of the approximating subspace.

In a well-posed problem the eigenvalues are always positive, and the minimal and maximal eigenvalues correspond to some stiffness values. Provided that the condition number is relatively

small, meaning that the stiffness is relatively equally distributed over the domain, the system is called well-conditioned. Otherwise, if the condition number is very large and e.g. if there are neighbour elements with a stiffness differing by several orders of magnitude, the system is ill-conditioned and behaves as singular or quasi-singular.

The mathematical framework in [68] showed that $\lambda_{min}(\boldsymbol{K})$ is directly associated with the area or the volume of the smallest element. In the same paper some bounds for the largest system eigenvalue $\lambda_{max}(\boldsymbol{K})$ were given as

$$\lambda^E_{max} \leq \lambda_{max} \leq n\, \lambda^E_{max}.$$

Hereby λ^E_{max} denotes the largest eigenvalue of the *element stiffness matrices* and n is the maximum numbers of elements meeting at a single vertex. The conclusion is that one awkwardly shaped element in an otherwise acceptable mesh could lead to an ill-conditioned system matrix. Another factor that can significantly influence the condition number is the element shape. We consider two limiting cases, an element with very small angles, as in Fig. 9(a), and an element with one large angle, as in Fig. 9(b). It is not surprising, that it can be shown that both limiting cases

(a) (b)

Figure 9: Elements with rather extreme interior angles. In (a) an element with one acute interior angle (α) and in (b) an element with one obtuse interior angle (β) is shown.

influence the condition number negatively.

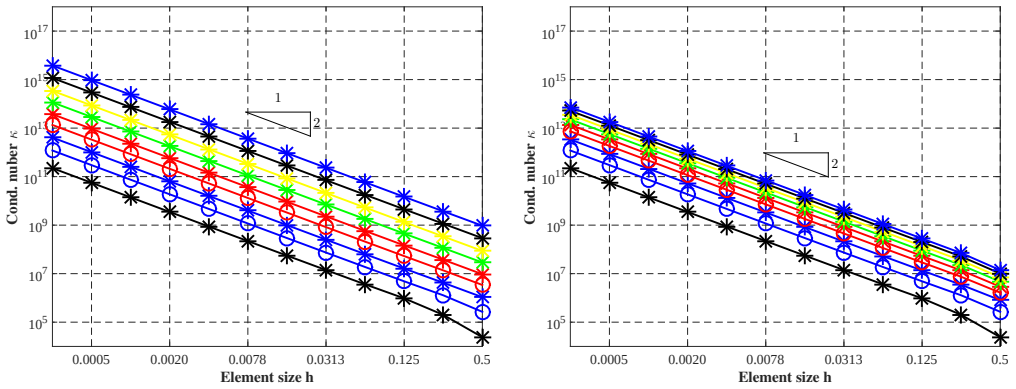
The condition number for nodal based finite elements is $\mathcal{O}(h^2)$, with h as the characteristic element size. This means, that the system becomes increasingly ill-conditioned, the smaller the element size is, which is clearly reflected in Fig. 10. For illustration purposes we consider now an one-dimensional problem, a bar with uniform cross section under axial loading. The loading function is given by $f(x) = \sin(\frac{\pi x}{\ell})$, the discretization is done using an equidistant distribution of nodes, in Fig. 10(a), and a Chen-Babuška distribution in Fig. 10(b).

The proportion of the condition number to the element size is for both node distributions the same, however, the absolute value of the condition number is better for the Chen-Babuška nodes. This reflects the fact that the basis is becoming linearly dependent with an increasing p. Another remedy to this problem is to use elements defined on a hierarchical basis, see [157].

2.7 Convergence rates

Except for very few special cases the finite element solution in Eq. (2.28) is only an approximation of the exact solution. The accuracy of the solution depends on several conditions:

(i) The first condition is related to the spatial discretization, meaning the degeneration of the elements in the physical space. As seen, the condition number is highly influenced by ele-



(a) Elements with equispaced nodes

(b) Elements with Chen-Babuška nodes

Figure 10: Condition number κ in dependence of the mesh size parameter h for element orders $p = 1$ to $p = 9$.

ments with either very small or very large angles. Interestingly, the interpolation property depends only on large angles, therefore a triangular element, as in Fig. 9(a) or quadrilaterals that degenerate to triangles pose no problems, for more details see [52] and the references therein.

(ii) The chosen polynomial degree of the elements. It is expected that the accuracy increases with a higher p. For one, curved boundaries can be described better, for the other, a higher polynomial has a smaller interpolation error for general functions. Closely related to the chosen polynomial degree is the chosen numerical integration scheme. Usually the integrals in Eq. (2.29) are evaluated numerically, however, if for some reason the numerical integration is not accurate enough, we expect the whole approximation scheme to underperform.

(iii) The last point concerns the regularity of the exact solution. If the solution contains singularities (as e.g. singularities in the stresses near re-entrant corners) or if the solution function is rough (kinks or jumps), it is expected that the accuracy deteriorates.

There are two general classes of error estimates for the finite element method. The first are *a posteriori* estimates, which give a precise measure on the total error, given a computed approximate solution $\boldsymbol{u}^h$. These estimates are important to design adaptive procedures and also to point out domains with a disproportional large error. The other class of methods are *a priori* estimates. They tell us the estimates on the convergence rates, which is loosely spoken the speed at which the error decreases as the number of degrees of freedom is increased. The bounds on the error ϵ are expressed in the following form

$$||\epsilon|| \leq C\, h^m p^n \approx C\, \mathcal{O}(\text{DoF})$$

with h as characteristic length of the element, p as polynomial degree of the elements and C, m, n are generic constants independent of the discretization parameters, see [24]. The constant C is usually not of interest, and the rate $\mathcal{O}(\text{DoF})$ as a function of the degrees of freedom depends on the regularity of the solution and the employed norm. As mentioned, the mathematical theory of

finite element methods was established in the 1970s starting with the h-version of the method, [6, 13]. The refinement proceeds simply by introducing more elements (or reducing the characteristic element size h), while keeping the usually low order of the polynomial ansatz functions. The other possibility is to keep the number of elements constant but to increase the number and order of basis functions over each element, which is analogous to increasing the polynomial order p; therefore this strategy is referred as p-refinement. These two methods differ significantly in terms of convergence properties; while the h-refinement reduces the error rather slowly the p-refinement gives an exponential decay of the error. The third possibility is to combine the two approaches and to change the mesh and the number of basis functions over the elements at the same time; this is referred to as hp-refinement.

We investigate now the *a priori* error in the primary variables (in our case displacements measured in the L^2-norm) as well as in the gradient of the primary unknowns (strains or stresses measured in the H^1-norm) on three simple examples. The first one has a smooth solution over the whole domain, the second a weak discontinuity (or a strong discontinuity in the gradient). The first two examples are one-dimensional, the last example shows the decay of the convergence rates for a two-dimensional example, where the geometrical features (a curved boundary) are not captured appropriately.

Regarding the employed error measures for the convergence studies for this and also for the next example, using the analytical solution $\boldsymbol{u}$ and the finite element solution $\boldsymbol{u}^h$, the L^2-norm is defined similar to Eq. (2.25) as

$$\varepsilon_{L^2} = ||\boldsymbol{u} - \boldsymbol{u}^h||_{L^2(\Omega)} = \sqrt{\int (\boldsymbol{u} - \boldsymbol{u}^h)^\intercal (\boldsymbol{u} - \boldsymbol{u}^h) \, \mathrm{d}\Omega}. \tag{2.42}$$

To investigate the convergence behaviour of function derivatives, we use the strain energy H^1, defined as

$$\varepsilon_{H^1} = ||\boldsymbol{u} - \boldsymbol{u}^h||_{H^1(\Omega)} = \sqrt{\int (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^h)^\intercal \mathbf{C} (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^h) \, \mathrm{d}\Omega} \tag{2.43}$$

with $\mathbf{C}$ being the elastic stiffness tensor and $\boldsymbol{\epsilon}(\boldsymbol{u})$ and $\boldsymbol{\epsilon}^h(\boldsymbol{u}^h)$ the strain fields of the exact solution and the finite element approximation, respectively. We use relative versions of these two norms, therefore $\varepsilon_{L^2(\Omega)} = ||\boldsymbol{u} - \boldsymbol{u}^h||_{L^2(\Omega)} / ||\boldsymbol{u}||_{L^2(\Omega)}$ and $\varepsilon_{H^1(\Omega)} = ||\boldsymbol{u} - \boldsymbol{u}^h||_{H^1(\Omega)} / ||\boldsymbol{u}||_{H^1(\Omega)}$.

The example is once more a bar with a uniform cross section under axial loading, fixed on the left boundary. The loading function is given by $f(x) = \sin(\frac{\pi x}{\ell})$, and the discretization is done using an equidistant distribution of nodes. We refine by keeping the polynomial ansatz p constant and construct a sequence of nested meshes with the refinement parameter h as element length. The solution for displacements and displacement derivatives is given as

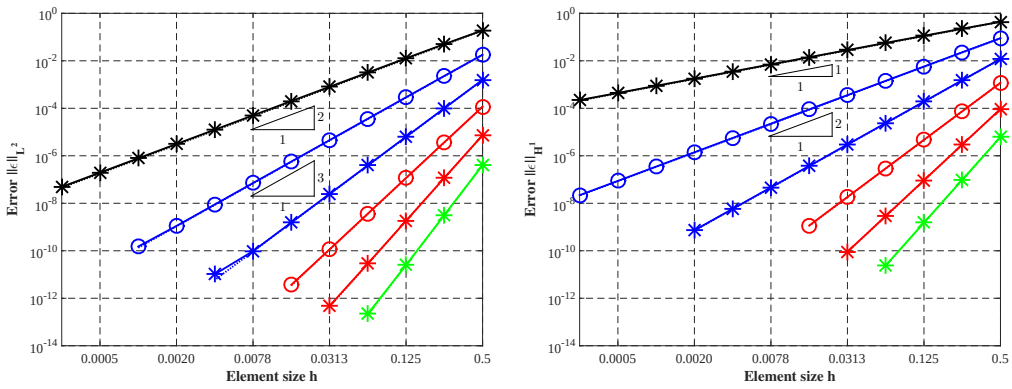
$$u(x) = \frac{1}{EA} \left(\frac{\ell^2}{\pi^2} \sin(\frac{\pi x}{\ell}) \right) \quad \text{and} \quad u_{,x}(x) = \frac{1}{EA} \left(\frac{\ell}{\pi} \cos(\frac{\pi x}{\ell}) \right).$$

The functions are obviously $C^\infty(\Omega)$-smooth, under these conditions the expected optimal convergence rates are given as

$$||\boldsymbol{u} - \boldsymbol{u}^h||_{L^2(\Omega)} = \mathcal{O}(h^{p+1}) \quad \text{and} \quad ||\boldsymbol{u} - \boldsymbol{u}^h||_{H^1(\Omega)} = \mathcal{O}(h^p), \tag{2.44}$$

see [168, 161].

The results in Fig. 11 confirm the statements in Eq. (2.44), optimal convergence rates in both



(a) (b)

Figure 11: Error analysis in the L^2-norm in (a) and the H^1-norm in (b) for smooth solution fields for displacements and normal forces.

norms are achieved for the problem.

The next example is regarding geometry, boundary conditions and discretization similar to the first one, the only difference is the material model. The bar consists of an elastoplastic material, with the material parameters *chosen* such that the interface between elastic and elastoplastic regions is located just in the middle. As the system is statically determined, it suffices to calculate the normal force in the middle, given as

$$\sigma_{Y_0} := \int_{x=\frac{\ell}{2}}^{\ell} \sin\left(\frac{\pi x}{\ell}\right)$$

The analytical solution for the displacements is given as

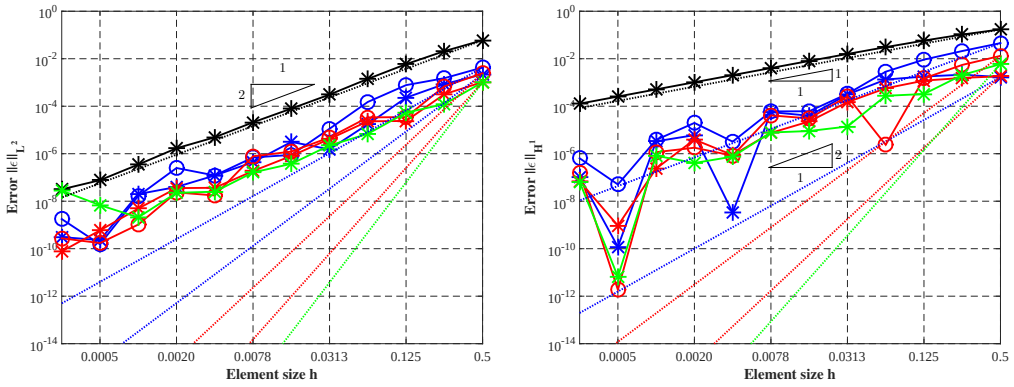
$$u(x) = \begin{cases} \frac{\ell^2 \sin\left(\frac{\pi x}{\ell}\right) + \pi \ell x - \sigma_0 \pi^2 x}{E_T \pi^2}, & \text{if } \boldsymbol{x} \in \Omega^+ \\ u\left(x = \frac{\ell}{2}\right) + \frac{\ell^2 \sin\left(\frac{\pi x}{\ell}\right) + \pi \ell x - \sigma_0 \pi^2 x}{E \pi^2} - \frac{\frac{\pi \ell^2}{2} + \ell^2}{E \pi^2} & \text{if } \boldsymbol{x} \in \Omega^- \end{cases}$$

with the elastoplastic tangent modulus $E_T = \frac{K \cdot E}{K+E}$ and K as plastic modulus. Notice that the displacements are continuous, but have a kink in the interface, that is, $u(x) \in C^0(\Omega)$. In contrast to the displacement field, the stress field is continuous and smooth, and defined as

$$\sigma_x(x) = \frac{\ell}{\pi} + \frac{\ell}{\pi} \cdot \cos\left(\frac{\pi x}{\ell}\right) \qquad \text{if } \boldsymbol{x} \in \Omega$$

Subject to these limitations, the convergence rates are depicted in Fig. 12. It is important that the discretization is chosen in a way that the elastoplastic interface is always inside of the element, and not on the boundary between two elements. The convergence rates drop dramatically, we end up with convergence orders of

$$||\boldsymbol{u} - \boldsymbol{u}^h||_{L^2(\Omega)} = \mathcal{O}(h^2) \quad \text{and} \quad ||\boldsymbol{u} - \boldsymbol{u}^h||_{H^1(\Omega)} = \mathcal{O}(h^1), \tag{2.45}$$



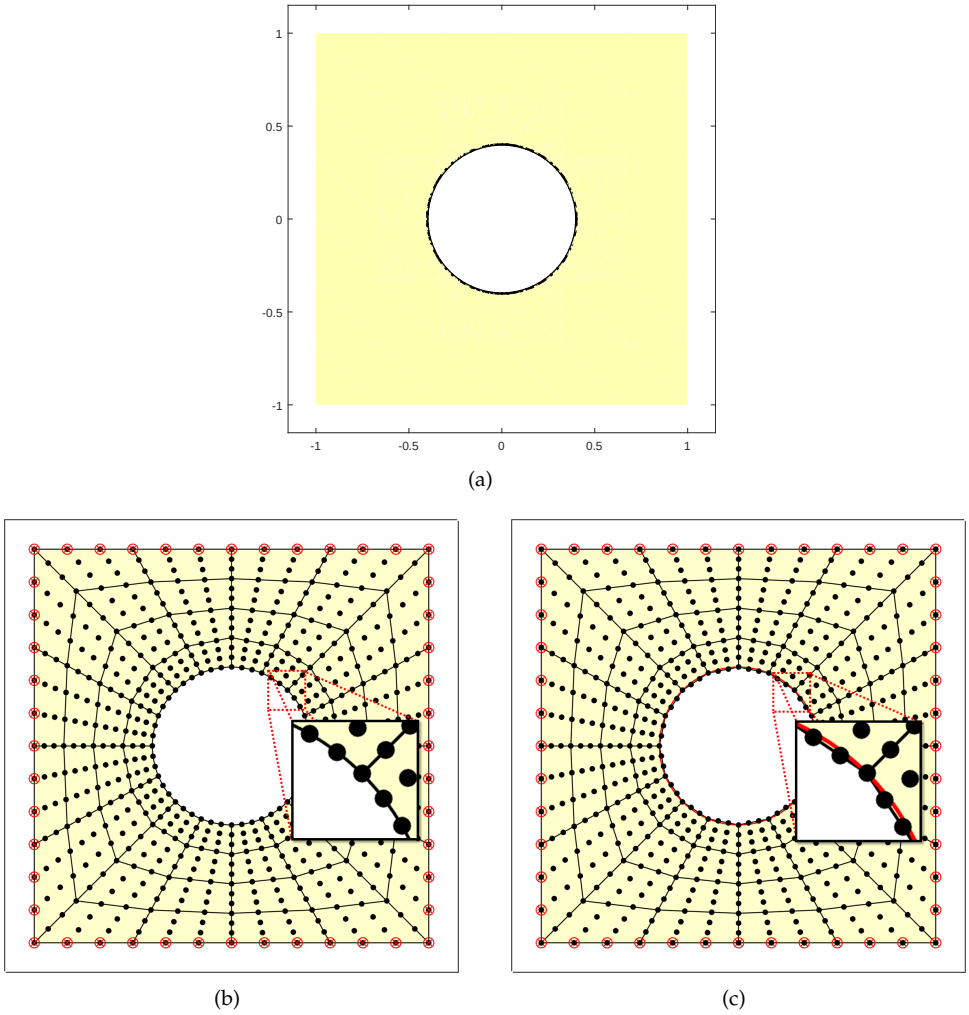
(a) (b)

Figure 12: L^2-error and H^1-error for a solution field with an elastoplastic interface within an element for displacements in (a) and normal forces in (b), respectively.

Therefore, the convergence rates do not change by increasing the element orders, which is due to the reduced continuity of the elastoplastic interface within an element.

The last example is a plate with a circular inclusion, as depicted in Fig. 13(a). All material properties as well as the analytical solution for displacements, strains and stresses are given in Section 7, as this example will be further used as an benchmark problem for our proposed scheme. We compare a conforming mesh as in Fig. 13(b) with a piecewise straight approximation of the curved boundaries, as visualized in Fig. 13(c).

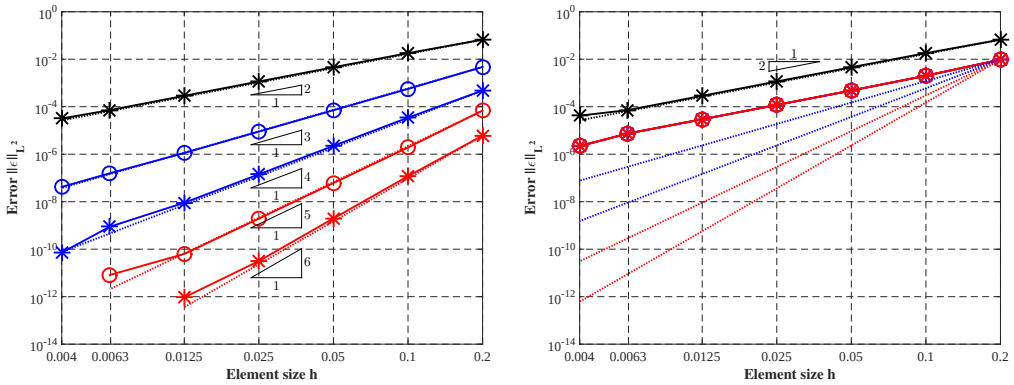
It is seen in Fig. 14 that the convergence rates of the non-fitted example are independent of the ansatz order of the finite element functions and are limited by the accuracy of the geometry description. All element orders with $p > 1$ are visually indistinguishable and converge with a very low-order as characteristic for linear elements.



(a)

(b) (c)

Figure 13: Sketch of the considered problem in (a), discretization using conforming elements in (b) and a discretization with a linear description of the boundary in (c).



(a) (b)

Figure 14: The L^2-error for the discretizations with a conforming boundary in (a) and the L^2-error for a non-conforming boundary in (b).

3 Higher-order finite element methods on unfitted meshes

As was demonstrated in the last section, the fully automatic representation of geometries with external or internal interfaces constitutes one of the main challenges in body-fitted finite element methods. Instead of the computationally costly mesh generation, which requires a large amount of human interaction, an alternative approach was already presented in the 1970s.

C. Peskin introduced in his thesis [144] and later on in a series on papers a novel method to study flow patterns around heart valves. The distinguished feature of the method was the manner how the space discretization was executed. The numerical simulation was performed on a Cartesian grid, with the boundaries not aligning to the physical domain. The domain of interest, the current geometry, was embedded into a geometrically bigger and simpler one. The crucial point of the technique was the enforcement of the boundary or interface conditions. This was done by modifying the resulting system of equations, namely, the coupling between the solid and the fluid was taken into account first by considering the solid a part of the fluid and then applying additional forces via a Dirac delta function on the fluid. The original idea relied on the use of a finite difference method to solve the equations, but in the meantime the method was also extended to the finite element regime, see [22, 178]. Nowadays, the method is applied mainly for fluid-structure interaction problems and, as in the early approaches, low-order finite difference schemes prevail.

Another, fairly similar, approach that came up in the 1990s, was introduced by R. Glowinski in [81]. In contrast to the approach by Peskin, one domain is not embedded in the other one, but rather the two domains (the real and the fictitious) are coupled using Lagrange multipliers, either on the boundary or alternatively inside the fictitious domain, see [80].

Thereafter a whole group of methods came into existence with the same distinctive features, which is a non-aligned mesh and subsequent treatment of interface conditions, but depending on the application fields, different names were coined, e.g. fictitious domain method, embedded domain method, immersed boundary method, unfitted finite element method, etc. This list is by no means complete as there is no distinct definition of unfitted methods. One drawback of the methods presented so far is that the extension to higher-order approximations is difficult and has, therefore, not been achieved for most of the methods.

In particular in the last two decades various approaches were developed to deal with elliptic interface problems. In the remaining section, first the concept of the level set method is introduced. The level set method is used to describe the location and possibly the time evolution of interfaces. After that we introduce three different fictitious domain methods and show the concepts as well as the key points and the finite element formulation on the simple 1D example with the material interface, as introduced in the last section. The three techniques are quite different in the methodology but share the common property to be extendable to higher-order approximations. The presented methods are the *eXtended finite element method* (XFEM), the *finite cell method* (FCM) and the *cut finite element method* (CutFEM). Despite the fact that the methods are quite different, there are some difficulties shared by all of them. This is further elaborated in the last part of the section.

3.1 Level set method

As described in Section 1, the domain is divided by (possibly moving) *sharp* external interfaces Γ_{ext} and/or internal interfaces Γ_{int}. To describe the location of the interfaces in body-fitted methods a standard approach is to use an *interface tracking method*. Specifically, a Lagrangian perspective is taken and the mesh is updated in order to follow the interface, commonly deployed methods are e.g. the *elastic mesh update method*, see [59]. It is essential that the representation of the interface is explicit, either using standard isoparametric elements or cubic splines [76] or non-uniform rational B-splines (*NURBS*) [60].

Notwithstanding that, for fictitious domain methods another approach, known as *interface capturing*, is advantageous. The idea is to proceed with an Eulerian perspective and to introduce a scalar interface function, further referred to as level set function, $\phi(\boldsymbol{x}, t) : \mathbb{R}^{dim+1} \rightarrow \mathbb{R}^{dim}$. The zero value of the function $\phi(\boldsymbol{x}, t)$ is employed to define the location of the interface. In case of a potential transport of the interface, in addition an evolution equation of the level set function is necessary. It is noteworthy that the level set function is defined in a space which is one dimension higher than the interface. The zero level of the level set function is then for one-dimensional problems simply a zero-iso*point*, in two dimensions a zero-iso*line* and in three dimensions a zero-iso*surface*. Consider two regions Ω^+ and Ω^-, such that $\Omega^+ \cup \Omega^- = \Omega$ and $\Omega^+ \cap \Omega^- = \Gamma$, then, under a sufficiently smooth level set function ϕ, the interface is described *implicitly* by the zero level of this level set function as

$$\phi(\boldsymbol{x}, t) := \begin{cases} < 0, & \text{if } \boldsymbol{x} \in \Omega^+ \\ > 0 & \text{if } \boldsymbol{x} \in \Omega^- \\ = 0 & \text{if } \boldsymbol{x} \in \Gamma \end{cases} \tag{3.1}$$

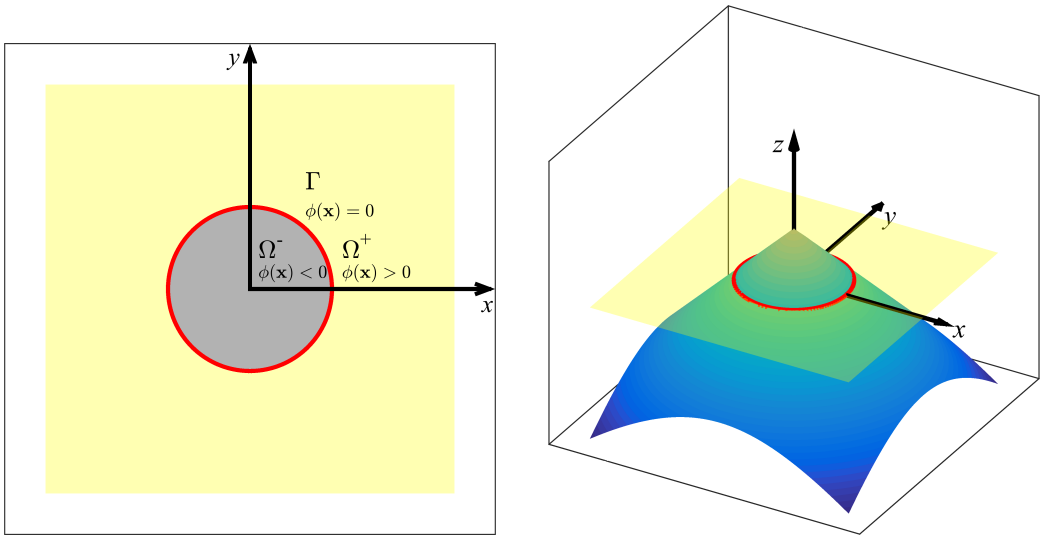
In addition, it is possible to assign a *physical* meaning to the sign of the level set function, namely to represent different loading functions, material properties or even different physical models in each of the subdomains Ω^+ and Ω^-. Another possibility is to assign *geometrical* meaning to the level set function, e.g. if only one of the aforementioned domains is of interest; the zero level set would herein represent the boundary of the computational domain.

In the considered examples in here, all the interfaces are static, the explicit dependence on the time t is omitted. However, in general the interface is time dependent and additional (transport) models are described by means of hyperbolic conservation laws, see [140, 151]. The introduced implicit description of interfaces may seem disproportionate at the first glance, but the versatility and the efficiency of the method are undisputed. The extension of the method to higher-dimensions (including sharp features) is quite straightforward and there exist higher-order accurate computational schemes to solve the governing evolution equations. Additional advantages of the level set approach are that the description of arising topological changes of the interface, such as merging and splitting of the fronts, is facilitated. Intrinsic geometrical properties, for instance normal vectors or the curvature of the interface, can be determined by the level set function [152, 139]. In fact, no other method has been as successful for capturing the geometry of evolving interfaces as the level set method. The application field of the level set method spans widespread problems in engineering, material science, computer graphics etc.

A very common choice of a level set function is the signed distance function, given as a

$$\phi(\boldsymbol{x}) := \pm \min_{\boldsymbol{x}_\Gamma \in \Gamma} ||\boldsymbol{x} - \boldsymbol{x}_\Gamma||, \qquad \forall \boldsymbol{x} \in \Omega$$

Let us study a concrete example: we describe a circular inclusion in 2D as $\phi(\boldsymbol{x}) := |\boldsymbol{x}|^2 - r^2, \forall \boldsymbol{x} \in \mathbb{R}^2$, the boundary of the inclusion is given with the zero-isolevel of the function ϕ, see Fig. 15.



(a) (b)

Figure 15: Representation of a circular inclusion separating the domains Ω^+ and Ω^- with a zero level set in (a), a three-dimensional plot of the zero level set in (b).

For the approximation of smooth level set functions the standard FE-error estimates apply, therefore the error in the approximation of the zero level is of order $\approx \mathcal{O}(p)$. It is noted, that other methods exist to represent interfaces in evolution (e.g. the volume of fluid methods), however, the level set method is easily combined with e.g. the XFEM as seen below in Section 3.2.

Remark 1 Frequently level set functions with signed-distance property are employed. However, in this work, we make no use of the special properties that are implied for such functions, such as $||\nabla\phi|| = 1$ for all $\boldsymbol{x} \in \Omega$.

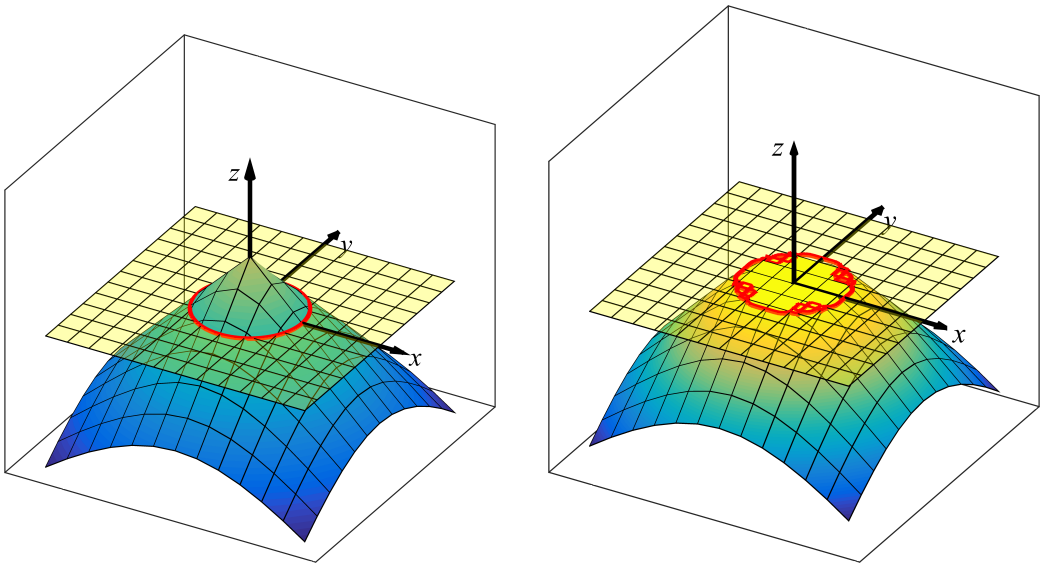
Remark 2 The level set method may also be used for the definition of open interfaces such as crack surfaces. Additional information is required to define the position of the crack front, for example, by additional level set functions.

If the level set function is described in a finite element setting, the values of the level set function are stored only at the nodes. Than, using the element shape functions, any value inside the element can be interpolated. The discretized level set function reads as

$$\phi^h(\boldsymbol{x}) = \sum_{i \in I} N_i(\boldsymbol{x})\, \phi_i. \tag{3.2}$$

Herein I is the set of all nodes in the domain of interest, and $\phi_i = \phi(\boldsymbol{x}_i)$ are the level set values at the element nodes. If the level set function is discretized using finite elements, the error of

the discretization is bounded by the standard interpolation error of the FEM, see Section 2.7. Therefore, as long as the level set function is smooth, as the example in Fig. 15, the discretized zero level set might be used for an accurate geometry representation. Otherwise, if some non-smooth features as edges or intersections occur, it may be necessary to place elements in a way that their edges align with the discontinuity; the topic is further discussed in Section 4.



(a) Discretized signed distance level set function

(b) Discretized scaled level set function

Figure 16: Comparison of discretized level set functions for a signed distance level set function in (a) and a scaled version in (b). It is evident that for the scaled version Runge's phenomenon appears and may worsen the result significantly.

We note that due to the fact that Lagrangian type shape functions may become negative, even in the case that all nodal values are positive, a situation might can occur as depicted in Fig. 16. Here the level set function, which is the signed distance function, is scaled by a function $\delta(\boldsymbol{x})$. This scaling function is defined as

$$\delta(\boldsymbol{x}) := \begin{cases} 100, & \text{if } \boldsymbol{x} \in \Omega_1 \\ \frac{1}{100} & \text{if } \boldsymbol{x} \in \Omega_2 \end{cases}. \tag{3.3}$$

If the initial level set function is multiplied by $\delta(\boldsymbol{x})$, clearly the same analytic zero level set is produced. This stems from the strict positivity of the function $\delta(\boldsymbol{x})$. The effect of the scaling does not influence the smoothness of the function ϕ but it influences the gradient of ϕ. Note that just at the zero level set a kink in the function appears, which results in a jump in the gradient. When the scaled level set function is discretized using Eq. (3.2), it becomes apparent that especially for higher-order discretizations Runge's phenomenon may appear resulting in some oscillations close to the zero level set. This pollution is more pronounced the larger the factor is.

3.2 Extended finite element method

The extended finite element method may be seen as the result of various approaches and existing methods in the 1990s to treat evolving discontinuities (e.g. free surfaces, cracks, phase interfaces) and simultaneously avoid global remeshing. Until then (strong) discontinuities were incorporated using interface elements with a double set of nodes, commonly combined with remeshing schemes, see [20]. Alternatively, embedded discontinuity elements were utilized and the displacement jump was incorporated as an incompatible strain mode [155].

Some ideas used in the XFEM have originated in mesh-free approximations. In fact, in the first related versions the XFEM was understood as a meshless method. Two examples are M. Fleming [65], who included asymptotic crack tip fields to *enrich* the displacement field in a crack analysis and J. Melenk and I. Babuška, [125, 12], who introduced the *partition of unity* (PU) framework. This concept allows to "glue" together locally defined approximation spaces or patches to define globally conforming approximation spaces of arbitrary regularity. The partition of unity concept constitutes a key ingredient to the framework of extended/generalized finite element methods as it allows to use possibly discontinuous approximation functions that mimic the behaviour of the solution. To apply the PU method it is essential to have some a priori knowledge regarding the smoothness, that is, the continuity properties of the solution. A set of functions $\Psi_i(\boldsymbol{x})$ has the partition of unity property if

$$\sum_{i\in I} \Psi_i(\boldsymbol{x}) = 1, \qquad \forall \boldsymbol{x} \in \Omega \tag{3.4}$$

with I being the set of nodes in the domain. In the milestone papers [125, 12] there is no assumption on the set Ψ_i; the functions can be chosen completely arbitrary as long as they satisfy the PU requirements, in particular, they can be meshfree. Notice that the standard finite element shape functions of the Lagrange-class as introduced in Section 2.4, also fulfill the PU requirement. Consequently

$$\sum_{i\in I} N_i(\boldsymbol{x})\psi = \psi, \qquad \forall \boldsymbol{x} \in \Omega \tag{3.5}$$

applies. Eq. (3.5) is valid for *any function* ψ, hence even functions with kinks or jumps can be reproduced *exactly*. If the function ψ can represent the solution of some boundary value problem, the product on the left hand side of Eq. (3.5) also will. The initial idea was to improve the performance of classical finite elements by augmenting the standard shape functions *globally* by some special shape functions. In [125] for example, a global harmonic polynomial function is added to the approximation space to deal with a (globally) non-smooth solution. Utilizing the PU framework, *any* enrichment function can be embedded in a globally conforming approximation space.

At the end of the 1990s a research group in Texas (T. Strouboulis and others in [162]) coined the name *generalized finite element methods* (GFEM) for a class of methods using the partition of unity concept and a *global* enrichment in combination with the classical FEM. The PU concept was realized using standard FE functions, with the motivation to solve elliptic problems on domains with voids. As most non-smooth phenomena, e.g. jumps, kinks or singularities, are rather local phenomena, soon in the GFEM the idea of *local* enrichments was introduced.

At the same time the group at the Northwestern university (J. Dolbow, T. Belytschko) used

the same concepts as mentioned before, but applied a *local* enrichment strategy from the very beginning. Their motivation was to handle strong discontinuities occurring in crack propagation problems. The method was termed *extended finite element method* (XFEM), and later on the approach was prolonged for general interface problems including voids, solidification, fluid-structure interaction problems, etc.

M. Stolarska proposed in [160] to use the level set method in an XFEM approach to represent the crack path as well as to model the crack growth. The combination of the level set method with the XFEM was the final piece and from there on a rewarding collaboration for both methods. Beside providing a method to describe and update the interfaces, the use of the level set method offers additional capabilities such as simplifying the selection of the enriched nodes and more important: the definition of the enrichment functions.

Nowadays the GFEM and the XFEM both exist for local enrichments and differ mainly in the used PU functions and enrichment functions. In order to explain the method in more detail we proceed with a scalar problem. The XFEM approximation of a scalar function $u(\boldsymbol{x})$ can be written in the following form

$$u^h(\boldsymbol{x}) = \sum_{i \in I} N_i(\boldsymbol{x}) \cdot u_i + \sum_{i \in J} N_i^\star(\boldsymbol{x}) \cdot \psi(\boldsymbol{x})\, a_i \tag{3.6}$$

The first part on the right hand side are the standard finite element shape functions and u_i the standard nodal degrees of freedom defined on the set of *all* nodes I in the computational domain Ω. The second part represents the enrichment term, with some globally defined enrichment function $\psi(\boldsymbol{x})$ multiplied by standard finite element shape functions and the additional degrees of freedom a_i. The functions $N_i^\star$ are standard FE shape functions that are defined over a subdomain $\Omega^\star \in \Omega$. The partition of unity is realized only over a *reduced* set of nodes J, that is, $J \in I$. Choosing a subset of enriched nodes means from an element point of view that the entire domain is divided into three disjoint parts. According to which of the subdomains the elements belong, they are classified as

(i) Reproducing elements: The set of elements with exclusively enriched nodes. The corresponding shape functions $N_i^\star$ form a partition of unity over the domain and are able to *exactly* represent the global enrichment function.

(ii) Standard (or unenriched) elements: The set of elements affiliated to the nodes with the index I consists of elements with no enriched nodes.

(iii) Blending elements: The set of elements where only a subset of the nodes is enriched, namely, these elements are the transition elements between reproducing and standard elements. The N_i are not zero but *they also do not form a PU*. Therefore, the elements can only partially, but not fully, reproduce the global enrichment function $\psi(\boldsymbol{x})$.

The fact that blending elements cannot represent the enrichment function does not present a problem, especially as we are only concerned with reproducing the global enrichment in elements that contain a discontinuity. However, the enrichment terms introduce some parasitic terms into the approximation and can considerably diminish the accuracy of the solution. Indeed, the reason for the existence of different types of enrichment functions to treat the same kind of discontinuity

are actually the blending elements. As we shall see, a special choice of enrichment functions can prevent problems with the blending elements.

A global enrichment, meaning $I = J$ (therefore there are no blending elements) would destroy one of the main advantages of the finite element method and would result in a much larger stiffness matrix. It is therefore a natural choice to enrich only some of the nodes as the discontinuous phenomena are often local, in particular in the presence of interfaces.

The discretized weak form, including the enrichment terms, is basically identical to the standard FEM. In case that only weak discontinuities occur, e.g. as in bi-material problems, the displacement field (as well as the stress field) is actually continuous across the whole domain. This results in a continuous weak form for the extended finite element method, which is formally identical with the one for a standard FE method. However, individual terms in the resulting system of equations look slightly different due to the enrichment part that is defined in Eq. (3.6). The weak form can be rewritten as

$$\begin{aligned} &\text{find } \bar{u}^h \in H^1(\Omega) \quad \text{such that} \\ &-\int_\Omega \nabla(\bar{N}_i(\boldsymbol{x})\bar{u}^h) \cdot \nabla(\bar{N}_i(\boldsymbol{x})\bar{w}^h) = \int_\Omega f\,(\bar{N}(\boldsymbol{x})\bar{w}^h) - \int_{\partial\Omega} t\,(\bar{N}_i(\boldsymbol{x})\bar{w}^h) \qquad \forall \bar{w}^h \in H_0^1(\Omega) \end{aligned} \tag{3.7}$$

The left hand side is a slightly modified version of the standard stiffness matrix, the right hand side the load vector. If the element is not cut and is not adjacent to a cut element, the shape functions and the degrees of freedom are according to the standard finite element method, therefore

$$\bar{N}_i = [N_i(\boldsymbol{x})] \qquad \bar{u}_i^h = [u_i] \quad \forall \boldsymbol{x} \in \Omega_I. \tag{3.8}$$

However, if the element is cut, the modified shape functions and the solution vector are defined as

$$\bar{N}_i = [N_i(\boldsymbol{x}) \quad N_i^\star(\boldsymbol{x})] \qquad \bar{u}_i^h = [u_i \quad a_i] \quad \forall \boldsymbol{x} \in \Omega_J. \tag{3.9}$$

For a more comprehensive overview, including extensive description on enrichment and integration strategies, as well as the treatment of Dirichlet boundary conditions, see [72]. It is interesting to note that the basic idea of the XFEM is related to the concept of hierarchical finite elements. It is possible to introduce known solution properties to the approximation space, without affecting further the subjacent space, which are in our case the uncut elements.

As mentioned earlier, the choice of the enrichment functions is directly related to the nature of the eventually discontinuous field behavior. The specific choice of the enrichment is very important in order to avoid sub-optimal rates of convergence, which is mostly due to the blending elements. To illustrate the influence of different enrichment functions we present and discuss the XFEM analysis of an one-dimensional problem bi-material, thus containing a weak discontinuity only. The examined enrichment functions are the standard abs-function, a modified abs-function as well as the Heaviside function (therefore introducing a strong discontinuity and enforcing continuity of the primal field using Lagrangian multipliers) and the so-called corrected XFEM.

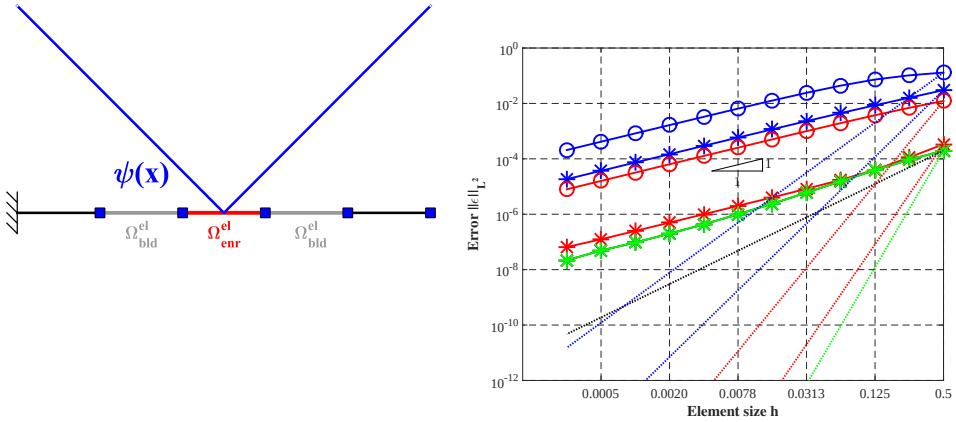
Abs enrichment. We start with the simplest choice of the enrichment function, as introduced in [165]. The function is commonly used to describe holes and inclusions and is given as

$$\psi(\boldsymbol{x}) = |\phi(\boldsymbol{x})| \tag{3.10}$$

The function $\psi(x)$ gives the absolute distance to the interface location and forms a reversed hat over the domain, see Fig. 17(a). The enrichment has a weak discontinuity in the function itself and hence a strong discontinuity in the function derivative. Considering Eq. (3.6), it becomes immediately clear that the approximation lacks the Kronecker-delta property, meaning that $u^h(\boldsymbol{x}_i) \neq u_i$. The key problem is that the imposition of Dirichlet boundary conditions gets more difficult when compared with the standard FEM. However, to avoid additional difficulties, in [17] a conceptually simple remedy to this problem was suggested that became standard from there on. The enrichment function is chosen of the following form

$$\psi(\boldsymbol{x}) = |\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}_i)|. \tag{3.11}$$

The basic idea is to relocate the enrichment function in a way that $\psi(\boldsymbol{x})$ becomes zero at each node. This approach is also known as *shifting* of the enrichment functions.
Nonetheless, there is one serious drawback with this enrichment function as evident from the results in Fig. 17(b). The convergence rates are far from optimal, which is due to the blending elements. As theoretically devised in [35], the blending or transition elements introduce some undesired (parasitic) terms into the approximation space that pollute the solution. For this example these terms even dominate the approximation error and reduce the convergence rates to perform just as bad as linear elements where the kink is within elements.



(a) Global (unshifted) enrichment function (b) Convergence study for the displacements

Figure 17: Bi-material rod using the abs-function as enrichment in (a) and convergence properties of the considered enrichment in the L^2-norm in (b).

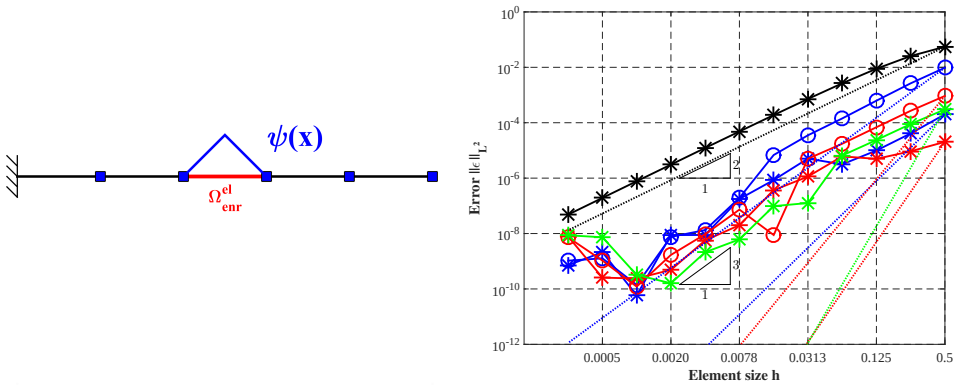
Curiously, in case that the enrichment function in the blending elements is not integrated, which amounts to neglect the enrichment function there, the one-dimensional case gives optimal convergence results, even for higher-orders. This result is rather an anomaly and unfortunately not transferable to the more general case of curved boundaries in multiple dimensions. Given the results, it is clear that the approach with the abs-enrichment, for 1D and for multiple dimensions, is not suitable for higher-order approximations.

Modified abs enrichment. This version of the enrichment functions was introduced in [126].

The global enrichment function is defined as

$$\psi(\boldsymbol{x}) = \sum_{i \in I} N_i(\boldsymbol{x}) |\phi_i| - |\sum_{i \in I} N_i(\boldsymbol{x}) \phi_i| \tag{3.12}$$

It is important to note that the enrichment function vanishes at the nodes of the reproducing element, this is visualized in Fig. 18(a), and therefore needs not to be shifted. As a consequence the problems with blending elements do not occur as there are, in fact, no blending elements for this approach. Nevertheless, when considering the convergence study in Fig. 18(b) it becomes evident that this type of enrichment is likewise not suited for higher-order approximations.



(a) Global modified abs-enrichment function (b) Convergence study for the displacements

Figure 18: Bi-material rod using the modified abs-function as enrichment (a) and convergence properties of the considered enrichment in the L^2-norm in (b).

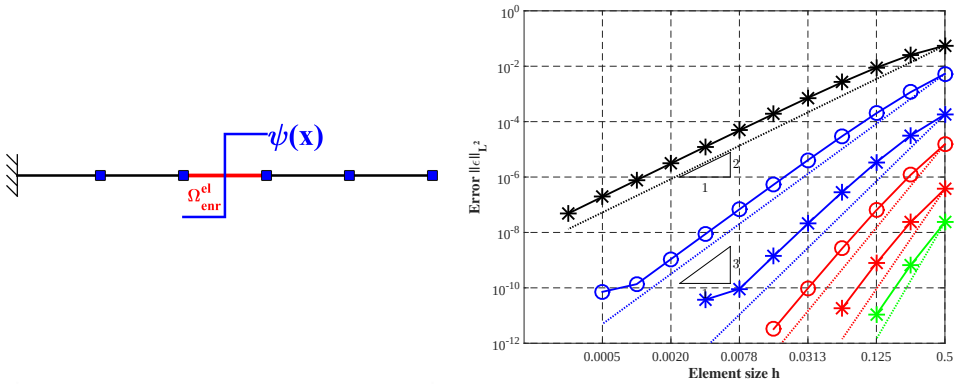
As seen, this approach gives optimal convergence rates for lower-order approximations, but fails to reproduce optimal convergence properties for all $p > 2$. The reason becomes more clear when considering a concrete example: in order to reproduce a weak discontinuity in a quadratic displacement field it is necessary to have 5 unknowns. Particularly, two quadratic parabolas give three unknown parameters, respectively, minus the matching condition which is due to the continuity requirement of the field. However, only three additional unknowns are provided by the XFEM approximation, clearly resulting in sub-optimal rates. Curiously, in case that a discontinuity is located just in the middle of an element, and a symmetry of the displacement field, optimal convergence rates are obtained.

Heaviside enrichment. The Heaviside-enrichment is due to [91] and is commonly used to represent strong discontinuities, as e.g. jumps in the displacement field in case of a crack face, [127] or to represent pressure discontinuities in a bi-material incompressible flow problem [87]. The enrichment function is defined either using the sign- or the Heaviside function. Both functions give identical results, as they span the same approximation space, for a vi-

sualization of the function see Fig. 19(a). The global enrichment may be written as

$$\psi(\boldsymbol{x}) = H(x) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in \Omega^+ \\ \text{-1} & \text{if } \boldsymbol{x} \in \Omega^- \end{cases}.$$

The XFEM approximation is obviously discontinuous along the interface, and hence the two domains Ω^+ and Ω^- are completely decoupled. To enforce the continuity, additional considerations must be made; continuity is then imposed either using Lagrangian multipliers or a consistent penalty parameter, both acting on the weak form. Using modifications in the weak form to enforce some side conditions is due to Nitsche and leads directly to the so-called CutFEM, which is discussed in more detail below. Using Lagrange multipliers to enforce continuity results in additional equations in the system matrix and transforms the standard FE formulation into a mixed formulation.



(a) Heaviside enrichment function (b) Convergence study for the displacements

Figure 19: Bi-material rod using the Heaviside function as enrichment (a) and convergence properties of the considered enrichment in the L^2-norm in (b).

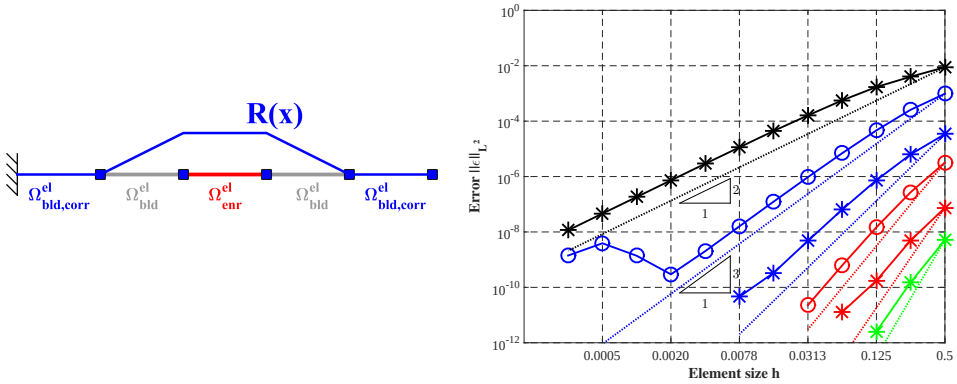
The results in Fig. 19(b) show that optimal convergence rates are obtained. The blending elements pose no problems as the local enrichment function is a product of the partition of unity functions and a linear (unshifted) or even zero function. In both cases, the enrichment function can be balanced by the standard finite element functions, therefore no parasitic terms occur.

The drawback herein is an augmented system of equations and the non-trivial choice of the Lagrange multiplier space. As is well known from mixed finite element methods, the choice of the standard approximation space and the Lagrange multiplier space is restricted by the Babuška-Brezzi conditions. In particular, it is not straightforward to extend this approach to multi-dimensional problems.

Corrected XFEM. The corrected XFEM was firstly introduced in [69], first and foremost as a remedy to the issues in the blending elements. A *ramp* or a *weight* function $R(\boldsymbol{x})$, as depicted in Fig. 20(a), is used to localize the standard abs-enrichment function before it is multiplied

with the PU part. As a result, the global enrichment function is

$$\psi(\boldsymbol{x}) = R(\boldsymbol{x})|\phi(\boldsymbol{x})|.$$



(a) Ramp-function (b) Convergence study for the displacements

Figure 20: Bi-material rod using the an enrichment function multiplied with the ramp function as in (a) and convergence properties of the considered enrichment in the L^2-norm in (b).

It is emphasized that for the corrected XFEM a slightly modified set of nodes is enriched compared to the standard XFEM. The set of nodes J consists of the nodes of the reproducing *and* of the blending elements. Consequently, a complete partition of unity is now present in both the reproducing and the blending element. This approach may be considered as shifting the initial blending elements one level away from the reproducing elements, as depicted in Fig. 20(a). The global enrichment function in the new blending elements is zero, therefore, no parasitic terms occur. In other words, this approach works essentially by localizing the global enrichment function, before it is multiplied with the partition of unity. As a result, the method recovers globally a partition of unity and leads to optimal convergence rates, see the results in Fig. 20(b). The general procedure is comparatively simple to implement and easily extended to other types of enrichment, for details see [69].

We have already discussed different choices of enrichment functions and their relevance to convergence rates. In addition to the enrichment functions, further implementational challenges arise, e.g. the FEM-solver must be able to handle a variable number of degrees of freedom per node.

Another theoretically and practically important point is that the ordinarily continuous entries in the stiffness matrix, the integrands, are actually non-smooth. A typical numerical integration procedure (e.g. the standard Gauß-Legendre quadrature) implies a polynomial approximation and certain regularity properties on the integrand and can, therefore, not be applied. To evaluate non-smooth, possibly even discontinuous integrands, other, specialized integration techniques are required. We will focus on this topic in Section 5.

3.3 Finite cell method

The finite cell method was introduced by the research group of A. Düster and E. Rank in [142]. As with most fictitious domain methods, the initial intention was to ease the burden of mesh generation for complex geometries. The important differences of the FCM compared with other fictitious domain approaches are

(i) The method was developed already with a p-extension of the version in mind. This is observable in the choice of the basis of the approximation space. Instead of the nodal-based Lagrange elements, which are used in this work, hierarchical modal-based basis functions are utilized.

(ii) The Dirichlet boundary conditions are enforced weakly e.g. by utilizing Lagrange multipliers, Nitsche's approach or the penalty method.

(iii) To reach optimal or close-to-optimal convergence rates a higher-order accurate integration scheme is inevitable.

The basic idea of the FCM is to define two complementary domains Ω^+ and Ω^-, as visualized in Fig. 21.

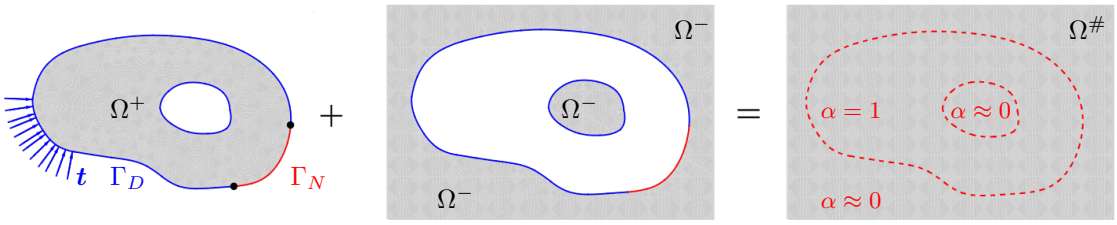


Figure 21: This sketch shows the basic idea of the FCM. The physical domain, here labelled with Ω^+ is completely embedded in $\Omega^\#$. $\Omega^\#$ is defined on the background mesh, which is often a Cartesian grid. The indicator function $\alpha(\boldsymbol{x})$ is introduced to scale the stiffness of the fictitious domain to a very small number.

The subdomain Ω^+ represents the real or physical domain of the problem with the actual occurring boundary Γ_{ext}. The complementary domain Ω^-, also referred as fictitious domain, extends the potential complex boundaries of Ω^+ to the union of the two domains $\Omega^\# := \Omega^+ \cup \Omega^-$. Note that (i) $\Omega^\#$ is defined as a geometrically simple domain and can be meshed conveniently by e.g. a Cartesian grid and (ii) $\Omega^\#$ is completely independent of the current boundary Γ_{ext}. The next component of the method is the definition of an additional scalar field $\alpha(\boldsymbol{x})$. This auxiliary the field is defined over the domain $\Omega^\#$, and varies depending on the spatial position as

$$\alpha(\boldsymbol{x}) := \begin{cases} 1, & \text{if } \boldsymbol{x} \in \Omega^+ \\ \ll 1 & \text{if } \boldsymbol{x} \in \Omega^- \end{cases} \tag{3.13}$$

Usually $\alpha(\boldsymbol{x})$ is taken as a factor to the material stiffness tensor in the stiffness matrix, resulting in the true material stiffness in the physical domain and assigning a fictitious, very small stiffness to Ω^-. Likewise, the body forces $\boldsymbol{f}$ on the right hand side are premultiplied with $\alpha(\boldsymbol{x})$, resulting in very small body forces in the fictitious domain Ω^-. According to [174], the value of $\alpha(\boldsymbol{x})$ can be

taken as zero as long as the polynomial degree of the ansatz is moderately low ($p < 4$). However, for higher polynomial degrees it is necessary to choose a value that is as small as possible but large enough to prevent an ill-conditioning of the system of equations. The reason to choose the factor as small as possible is immediately evident; the soft material is a good approximation of the void region around the solid body and, therefore, there will be only minimal interference with the solution in the physical domain.

Using the coordinate dependent material matrix and considering that there is no explicit representation of the boundary in the FCM, leads to the following weak form for the Poisson equation

$$\begin{aligned} &\text{find } u^h \in H^1(\Omega^{\#}) \quad \text{such that} \\ &-\int_{\Omega^{\#}} \nabla(N_i(\boldsymbol{x})u^h)\,\alpha(\boldsymbol{x})\,\nabla(N_i(\boldsymbol{x})w^h) = \int_{\Omega^{\#}} f\,\alpha(\boldsymbol{x})\,(N_i(\boldsymbol{x})w^h) - \int_{\partial\Omega^{+}} t\,(N_i(\boldsymbol{x})w^h) \qquad (3.14) \\ &\qquad\qquad \forall w^h \in H_0^1(\Omega^{\#}) \end{aligned}$$

Note that, in general, the integration interval is the background mesh of $\Omega^{\#}$, namely, the union of the real and the fictitious domain. As mentioned, predominantly structured meshes are employed and as a consequence the elements are named as *finite cells*. It is important to emphasize that the integration of the weak form is the essential point of the FCM. When integrating the weak form the information regarding the domain boundaries is taken into account by modifying the stiffness using the indicator function $\alpha(\boldsymbol{x})$. Basically, the complexity of mesh generation has been shifted to the higher-order accurate integration of the weak form. This is even more true in case of curved boundaries, where specialized integration procedures (e.g. decomposition methods, octree methods, moment-fitting) have to be applied. Another problem, closely related with the integration is the application of Neumann boundary conditions. The traction forces are applied directly on the true boundary Γ, therefore some explicit description of the boundary must be available. For an overview of numerical integration methods in the FCM-context see [1].

The solution is completely represented by means of shape functions on the background mesh. Another difference compared with the standard FEM is the choice of approximation spaces. As a matter of fact usually Legendre basis functions (hierarchical shape functions) are applied. There are also extensions using B-splines, [149], and Lagrange polynomials with Gauß-Lobatto points, in this case the method is referred as *spectral cell method* (SCM) [55]. The utilization of hierarchical shape functions reduces considerably the condition number, but may lead to difficulties in the geometry description. That is, the functions do not possess Kronecker delta property, therefore a linear system of equations needs to be solved to determine the coefficients for the coordinate mappings. The same is true for the application of Dirichlet boundary conditions.

It is noted that the finite cell method works "out of the box" for problems involving voids (or unfitted Dirichlet boundary conditions). However, in case of general weak discontinuities (material interfaces or elastoplastic materials) it is necessary to extend the approximation space and to introduce enrichment functions, similar to those used in the XFEM. Note, that if for a problem with unfitted interfaces equispaced Lagrange shape functions (i.e., the standard finite element functions) are used, the finite cell method is very similar to an extended finite element method with weakly enforced boundary conditions.

In the meantime a large number of publications on the FCM were released, including a rigorous mathematical analysis on the convergence properties of the method [47], and extensions to various application fields, including topology optimization [141], homogenization [95], large deformations [149] and plasticity [2, 3].

3.4 Cut finite element method

The cut finite element method (*CutFEM*) was introduced in a series of papers by E. Burman and P. Hansbo starting with [28]. There are quite some similarities to the XFEM but also to other fictitious domain methods. Essentially, the method might also be considered an additionaly stabilized XFEM-technique with jump enrichments and weakly enforced interface conditions. In fact, see [94] for a fictitious domain XFEM-approach that is very comparable with the CutFEM. The key points of the CutFEM are as follows

(i) The interface and, for unfitted boundary problems, the Dirichlet boundary conditions are enforced weakly, preferably using Nitsche's method [29]. Alternatively, there are variants using Lagrange multipliers [28].

(ii) In order to reach optimal convergence rates, irrespective of location of the interface within the elements, it is necessary to apply additional stabilization techniques. In the realm of CutFEM usually the ghost penalty approach is applied, see [25] .

Typically, boundary and interface conditions are built into the variational formulation using Nitsche's method. Simple background meshes such as Cartesian grids can be used, similar to XFEM and FCM.

The second point concerns stability issues due to the position of the interface. The interface can be described implicitly, preferably using the level set method. If the intersection is very close to an edge or to a face, integration over these parts may lead to very small or even negligible entries in the system matrix. This results in an ill-conditioned system of equations. There are different possibilities to solve this problem. In this particular method, an additional stiffness term is added to the weak form over the band of the elements cut by the interface. This term is independent on the location of the interface inside of the element and is referred as *ghost penalty*, [25], which is due to the fact that the nodes outside the real domain are sometimes called ghost nodes.

The standard formulation including Nitsche's approach and ghost penalty results in the fol-

lowing continuous weak form

$$\text{find } u \in H^1(\Omega) \quad \text{such that}$$

$$\int_\Omega \nabla u \cdot \nabla w - \underbrace{\int_{\Gamma_D} \nabla u \cdot \boldsymbol{n}_\Gamma w}_{(i)} - \underbrace{\int_{\Gamma_D} \nabla w \cdot \boldsymbol{n}_\Gamma u}_{(ii)} + \underbrace{\gamma_D \int_{\Gamma_D} \frac{1}{h} u\, w}_{(iii)} + \dots$$

$$\underbrace{\gamma_N \int_{\Gamma_N} h \boldsymbol{n}_\Gamma \cdot \nabla u \boldsymbol{n}_\Gamma \cdot \nabla w}_{(iv)} + \underbrace{\gamma_C \sum_{m \in F_G} h\, \nabla_{\boldsymbol{n}} u_m\, \nabla_{\boldsymbol{n}} w_m}_{(v)} = \int_\Omega f\, w - \int_{\partial\Omega} g_D\, w + \dots$$

$$\underbrace{\gamma_D \int_{\Gamma_D} \frac{1}{h} g_D\, v}_{(iii)} + \underbrace{\gamma_N \int_{\Gamma_N} h \boldsymbol{n}_\Gamma \cdot \nabla u \boldsymbol{n}_\Gamma \cdot \nabla v}_{(iv)} \qquad \forall w \in H^1(\Omega)$$

Beside the terms in the standard finite element formulation (i) arises from partial integration and establishes the consistency, (ii) gives symmetry to the formulation, (iii) and (iv) are the penalty terms for the Dirichlet and the Neumann boundary, respectively, ensuring the stability of the method and (v) is the ghost penalty stabilization. The ghost penalty term, that basically penalizes discontinuities in the gradient of the derivative, is independent on the position of the cut within the elements. From a mathematical point of view, this term extends the coercivity to the fictitious domain and eliminates the dependence of the condition number on the boundary location with respect to the mesh. For higher-order approximations the ghost-penalty takes into account higher derivatives, see [114]. γ_D, γ_N and γ_C are positive penalty parameters, for a choice see [29].

The great advantage of the approach is the stability and versatility it offers. The technique is applicable to virtually any geometry, including moving and evolving interfaces. It is even possible to treat PDEs on surfaces that are embedded in $\mathbb{R}^{\text{dim}}, \text{dim} = \{2, 3\}$ in a very similar fashion as PDEs defined in the bulk. The base functions of the background mesh are then evaluated on the intersection with the domain of interest, therefore again no explicit representation of the surface is necessary. In case of embedded surfaces the method is also referred as *traceFEM*. Using basically the same procedure problems involving non-matching and overlapping meshes can be addressed, see [123], also multi-physics coupling was investigated [26]. Curved boundaries pose essentially no problems and also higher-order approaches are possible, see [29, 114]. Another advantage is the existing mathematical literature. There is already a vast number of publications giving the method a theoretical basis including stabilization techniques, error and condition number estimates as well as results regarding inf-sup stability.

3.5 Common issues associated with embedded domain methods

In order to compare the discussed techniques, some important properties of the methods are discussed. Some issues are characteristic to each of the individual methods, e.g. the choice of the enrichment functions and treatment of blending elements in the XFEM, choice of the virtual stiffness parameter α (and also choice of enrichment function in case of weak discontinuities) in the FCM and the choice of stabilization parameters in the CutFEM.

However, many of the most important properties are, in fact, shared between the methods.

These are outlined below.

3.5.1 Imposition of boundary and interface conditions

The imposition of essential boundary and interface conditions is a major drawback of the methods outlined in this section, because the boundaries/interfaces are not explicitly represented by element edges or faces. The conditions are then typically imposed in a weak sense. These topics are still researched, see the overview publications [62, 40].

In general three different approaches, or minor modifications of them, are widely used to impose the boundary or interface conditions in a weak sense:

Lagrange multiplier method The Lagrange multiplier method is well known and widely used for optimization problems under constraints, [122, 77]. The approach to impose Dirichlet boundary conditions was originally defined in [7]. Despite the fact that the method brings some serious disadvantages with it, it is still often used due to the relatively simple implementation.

The method is based on the idea of introducing an additional field of unknowns, the field of Lagrangian multipliers, to enforce some side conditions, such as enforcing the incompressibility condition or linking of multiple subdomains. The standard approach involves minimization of a functional including two or more fields of unknowns. In our case the boundary conditions are interpreted as *side conditions* and a Galerkin scheme is applied. Only the situation of Dirichlet boundary conditions is considered for simplicity; the situation is similar for interface conditions. The side conditions are multiplied with the new set of unknowns (Lagrange multipliers) leading to a saddle-point problem (min-max problem), resulting in

$$\text{find } u \in H^1(\Omega) \quad -\int_\Omega \nabla u \cdot \nabla w - \int_{\partial\Omega} \lambda\, w = \int_\Omega f\, w - \int_{\partial\Omega} g\, w \quad \forall w \in H^1(\Omega) \tag{3.15}$$

$$\text{find } q \in H^1(\Omega) \quad -\textstyle\int_\Omega q \cdot u = \int_\Omega q \cdot w \quad \forall w \in H_0^1(\Omega) \tag{3.16}$$

The first equation is similar to the standard weak form, see Section 2.18, with the traction acting as a Lagrangian multiplier. The only difference is that by construction the test function does not need to be zero on the Dirichlet boundary. The second equation represents the weak form of the Dirichlet boundary conditions. After discretization, Eq. (3.15) results again in a linear system of equations.

We end up with two different fields of unknowns, known as *mixed finite element methods* in the literature. As well known since the 1970s it is necessary to carefully choose the approximation spaces for displacements and Lagrange multipliers carefully in order to fulfill the notorious *inf-sup condition.* Another possibility, which circumvents the inf-sup condition, is adding a stabilization term to the second equation. Different variants of stabilization terms exist, see [180] and the references therein. For more theoretical insight concerning mixed finite element methods, we refer to [21] and [15].

It is important to note that the bi-linear form on the right hand side is defined only on the Dirichlet boundary. However, if the Lagrangian multipliers λ are defined inside the whole

computational domain, this approach is called *distributed Lagrange multipliers*, see [93]. An approach similar to the distributed LMM is also presented in [181]. The obvious disadvantage of the distributed LMM, is that the approach introduces a field of additional unknowns which are usually discretized on the same mesh as the primary unknowns. This results in a significantly larger system than the one with the original variational form.
Defining λ only on the Dirichlet boundary has the advantage that the resulting discrete system is not considerably larger then the original one. For this reason, this approach is more wide-spread than the distributed LMM. In order to apply Eq. (3.15) directly, with λ defined only at the boundary, one needs an $(D-1)$-dimensional interface mesh that represents the boundary. The generation of such a mesh might be quite costly, especially in 3D. This is an important reason why the CutFEM introduces multipliers which are defined only on intersected elements, this is shown in [28]. The LMM method is often combined with the XFEM in the context of interface conditions, see [78, 79].

The clear disadvantages are that the size of the system of equations is considerably increased (by the number of constraint equations). Furthermore, the matrix in general loses the banded structure. And as a third point, zeros on the main diagonal appear. This imposes some additional requirements on the equation solver, see [24].

Nitsche method This method was introduced in [132] for the Poisson equation to handle Dirichlet boundary conditions weakly. The particular advantage of the method is that it does not require additional unknowns or a special interface mesh. However, the standard formulation requires the value of a penalty coefficient, for which the optimal value is still topic of research. It is interesting to note that the approach inspired the development of Discontinuous Galerkin (DG) methods.
The basic idea is to define and minimize a functional J (similar to the energy functional defined in Eq. (2.23)), penalizing both deviations from the solution of the PDE inside the domain and from the boundary conditions. The main step is the determination of J such that the resulting analytical minimization formula remains consistent and also fulfills some additional conditions as e.g. symmetry. In the resulting equation for a given PDE, in addition to the existing integral terms, new boundary integrals result. This leads to a consistent method to impose boundary conditions in a general way and *without additional degrees of freedom*. Due to its generality, Nitsche's method is often employed in generalized finite element formulations [10], and in mesh-based discretizations of interface problems [91, 92].
Starting now with the standard weak form, two terms are added, one term for symmetry

$$-\int_{\partial\Omega} \nabla v \cdot \boldsymbol{n}\,(u - g_D) \tag{3.17}$$

and another term that ensures stability

$$-\int_{\partial\Omega} \frac{\lambda}{h}\,(u - g_D)\,v \tag{3.18}$$

Therefore, at the end, the following weak form is obtained

$$\text{find } u \in H^1(\Omega) \tag{3.19}$$

$$-\int_\Omega \nabla u \cdot \nabla w - \int_{\partial\Omega} \nabla v \cdot \boldsymbol{n}\,(u - g_D) - \int_{\partial\Omega} \frac{\lambda}{h}\,(u - g_D)\,v = \dots$$

$$\int_\Omega f\,w - \int_{\partial\Omega} g\,w \qquad \forall w \in H_0^1(\Omega) \tag{3.20}$$

The implementation of the method is in general problem dependent. In the FCM, the Nitsche method is applied to handle Dirichlet BCs, in the CutFEM it is applied to impose general interface conditions. It is important to emphasize that no additional unknowns are introduced and in addition a symmetric and positive definite stiffness matrix is obtained. The Nitsche approach satisfies variational consistency in the sense that solutions of the weak form can be shown to be solutions of the original boundary value problem.

3.5.2 Linear dependence due to slightly cut elements

For all presented methods, except the CutFEM, one particular issue arises which can become critical and can prevent optimal convergence rates. Dependent on the location of the nodes in the background mesh and the position of the zero level set, a situation as depicted in Fig. 22 may arise. The element is cut very close to a node or to an element edge. In the XFEM this leads to almost linear dependent enrichment functions and involves an almost singular Jacobian matrix which results consequently in an ill-conditioned system matrix.

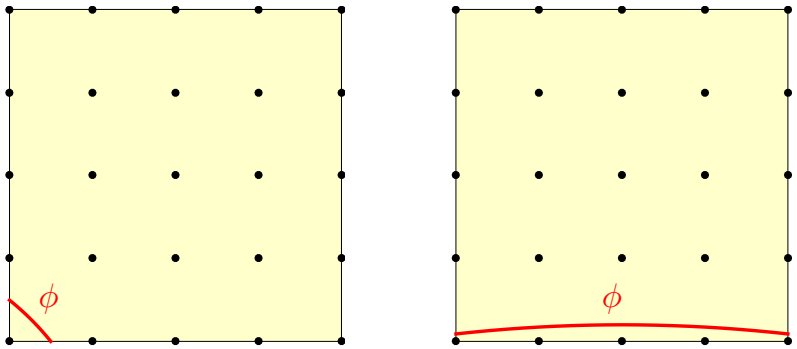


Figure 22: The zero level set, representing some interface passes close to an element node (a) or edge (b). Topological situations as depicted lead for any fictitious domain method to a poor conditioning of the resulting equation system.

3.5.3 Numerical integration in cut elements

Other issues, which are essential for all fictitious domain methods are for one the numerical integration of the weak form. This problem, especially in 3D applications, including curved edges and faces, poses still serious problems. In order to tackle this problem, the next section engages with higher-order accurate reconstructions of zero level sets. In fact, one of the major

contributions of this work is a consistent Integration scheme. As we shall see, the approach is based on a subdivision technique and is applicable for general background meshes and for two and three-dimensional applications. The resulting, discretized level set will then further directly be used in an integration and a remeshing context.

3.6 Other methods

Some other related methods in the context of fictitious domain methods are listed below.

One approach is due to S. Frei [66], who constructs *local* tessellations (triangulations) within the cut elements. This is done to define parameterized finite element functions, which allow for kinks in the solutions. Furthermore, a suited stabilization procedure is developed to bound the condition number. However, the approach has so far proven to work only for low-order elements.

Closely related is the immersed interface finite element method, e.g. see the references [83, 33], where in cut elements basis functions are constructed that satisfy the interface jump conditions.

For another approach in the realm of discontinuous Galerkin (DG) methods see [124]. Based on the work of [91], a method based on the symmetric interior penalty discontinuous Galerkin method (*SIPG*) with a Nitsche-type penalization at the interface is constructed. The method performs optimal for h-refinement and close-to-optimal for p-refinement.

The cut-cell method is a higher-order DG-method which stems from finite volume applications in fluid mechanics, and was introduced in [63, 64]. The approach uses an explicit representation of the interface via splines. The results indicate higher-order accuracy, however, under immense numerical costs for numerical integration and mapping procedures for the nodal points.

An approach in the XFEM-setting was introduced in [176], where an extended finite element approach based on DG-schemes is introduced. Optimal convergence rates are shown for higher-order approximations in L^2 and H^1- norms.

4 Reconstruction of zero level sets

As comprehensively discussed earlier, an explicit representation of the boundary, in our case the zero level set, is highly useful. This applies to fitted as well as to unfitted finite element methods, especially in a higher-order context. The detection and the meshing of the zero level set is crucial for the numerical integration and, as a consequence, for the approximation properties in an unfitted finite element analysis. A fitted finite element analysis requires, per definitionem, a higher-order accurate representation of the boundary to achieve optimal convergence properties.

The aim of this section is to locate and finally approximate the zero level set of the discretized level set function ϕ^h by use of higher-order finite elements of the Lagrange family, called *interface elements* herein. This step, the detection and the subsequent meshing of the zero level set is called *reconstruction* from here on. The topic of reconstructing zero level sets in two and three dimensions for one level set as well as for multiple level sets, including intersections that produce edges and corners, is discussed thoroughly in [73].

4.1 Reconstruction strategy in a nutshell

Below, we will present a new method for the detecting and meshing of the zero level set, as introduced in [73]. The proposed strategy has a local (element-wise) character, that is, the reconstruction of the zero isoline advances element by element, see Fig. 23, **step (i)**. For each physical element that is cut by the zero level set in the global domain $\Omega_{\boldsymbol{x}}$, the zero level set is reconstructed in the *reference element* $\Omega_{\boldsymbol{r}}$ in terms of the reference coordinates $\boldsymbol{r}$ and, finally, mapped to the physical domain using an isoparametric mapping.

The initial situation is a domain bounded by an external interface Γ_{ext} and possibly featuring an internal interface Γ_{int}. Observe that the boundaries do not conform with the spatial discretization, that is, the computational mesh. The goal of this section is to discretize the boundaries using only the information given in the background mesh.

The starting point of the reconstruction procedure, **step (ii)** is to transfer the level set data, given in the nodes of the background mesh of order m_Ω in the physical domain, to the corresponding reference domain. It is important to notice that the entire reconstruction process is executed in the reference element. The mapping of nodal level set values from $\Omega_{\boldsymbol{x}} \rightarrow \Omega_{\boldsymbol{r}}$ is particularly simple for elements of the Lagrange class, as the nodal values $\phi^h(\boldsymbol{x}_n)$ in the physical domain are taken as the nodal values $\phi^h(\boldsymbol{r}_i)$ in the reference element, $\phi^h(\boldsymbol{x}_n) = \phi^h(\boldsymbol{r}_i)$. The coordinate indices n and i denote a specific node in global and the corresponding node in local node numbering, respectively. Based on this information, elements are detected that are cut by the zero level set. Once the cut elements are identified, it must be ensured that the level set data is valid in the sense that a unique subdivision of the element is possible, otherwise some special treatment of these elements is required. For topologically valid cut elements, in **step (iii)** a tailored scalar root finding algorithm is provided, giving any desired number of points on the interface. These points are finally used to approximate the zero level set in the reference background elements by higher-order *interface* elements. The number of element nodes n_Γ is thereby directly related to the specified order m_Γ. Hence, there are n_Γ positions $\boldsymbol{r}_i^\Gamma$, $i = \{1, \dots, n_\Gamma\}$ that have to be found on the zero level set per element, see Fig. 23. Notice that the dimension of the interface element is always one order lower than the dimension of the problem, i.e. a surface for three-dimensional

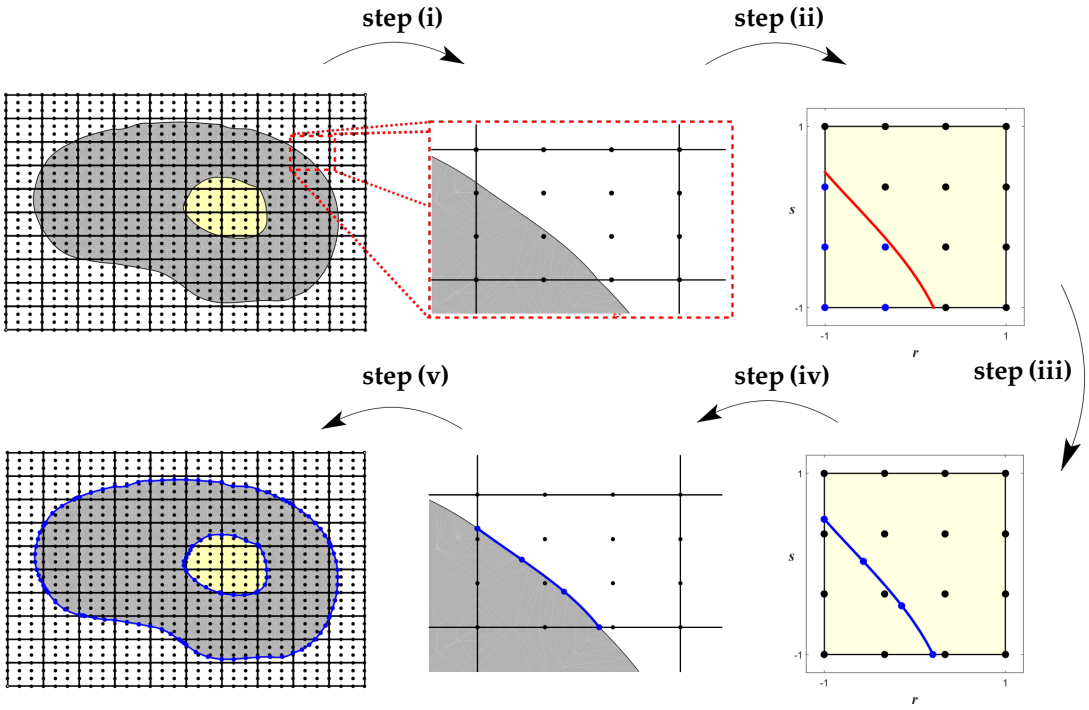


Figure 23: This sketch demonstrates the steps necessary to obtain a valid interface discretization. The procedure advances element by element, as in **step (i)**, and therefore each element can be investigated individually. We will further on investigate different variants to construct the interface element in the current section, as depicted in **step (iii)**. Note that the mapping $\boldsymbol{x}(\boldsymbol{r})$ in **step (iv)** is defined by the background elements of order m_Ω and the order of the background element is *independent* of the order of the reconstructed interface element. In this sketch the order of the background element as well as of the interface element is $m_\Omega = n_\Gamma = 3$.

problems and a line element in case of two-dimensional background meshes and that the interface elements are still defined in the reference domain $\Omega_{\boldsymbol{r}}$. To obtain the corresponding nodal positions in the physical space, the coordinates from the reference element are mapped isoparametrically to the physical domain using Eq. (2.33). The newly constructed interface elements will then serve either to construct a mapping for the integration points, or otherwise supply an ideal base to construct a new conforming discretization. The reconstructed interface mesh fulfills compatibility requirements for embedded meshes, provided that shared faces are treated consistently. Then, the reconstructed surface mesh in 3D is perfectly suitable as a computational mesh for surface PDEs.

The mapping of the interface elements with the order m_Γ, denoted with $\boldsymbol{x}(\boldsymbol{r})$ in **step (iv)**, is defined by the background elements of order m_Ω, see Fig. 23. We always choose elements from the Lagrange class for the *background* mesh as well as for the interface representation. The choice of Lagrange elements for the background mesh is somewhat arbitrary and rather for simplicity as any other type as e.g. B-splines or NURBS could serve the same purpose. However using Lagrange elements enforces automatically a $C^0(\Omega)$ continuity between the elements and also facilitates the computational realization of the method. The choice of the Lagrange class for the *interface* elements on the other hand seems crucial for the reconstruction due to the interpolation

properties of this element type.

Remark 1 The interface elements are in general only an approximation of the zero level set of the discretized level set function ϕ^h. Only the element nodes $\boldsymbol{r}_i^\Gamma$ are "exactly" on the zero level set of ϕ^h (up to a user-defined tolerance of approximately 10^{-12} in the root finding algorithm). Furthermore, also the "exact" zero level set of ϕ^h is only an approximation of the analytically given zero level set of the original level set function $\phi(\boldsymbol{x})$. It is obvious that the accuracy of this double-approximation is directly effecting the accuracy of the proposed integration schemes.

Remark 2 The proposed integration scheme relies on ϕ^h rather than ϕ. Namely, it also applies to situations where the exact level set function $\phi(\boldsymbol{x})$ is not known in closed form but only *nodal* level set values are known or may be determined with arbitrary accuracy. For example, this could be the case in a fictitious domain method, where nodal distances to NURBS surfaces, representing the boundary of the domain, may be computed. Another example are XFEM-simulations of problems with dynamic interfaces described by zero level sets: There, naturally, ϕ^h exists throughout the simulations even if in the first step an analytic ϕ was given. In this sense, using ϕ^h rather than ϕ is on the safe side: When the proposed scheme works for ϕ^h, it would perform at least as good if it were based on ϕ.

Let us summarize: In order to reconstruct valid, implicitly given, geometries by interface elements, the strategy proceeds as follows

(i) detect whether an element is cut by the zero level set, see in Section 4.2.

(ii) ensure that the situation is "topologically valid", see in Section 4.3.

(iii) provide a robust and accurate root finding algorithm for detecting the $\boldsymbol{r}_i^\Gamma$.

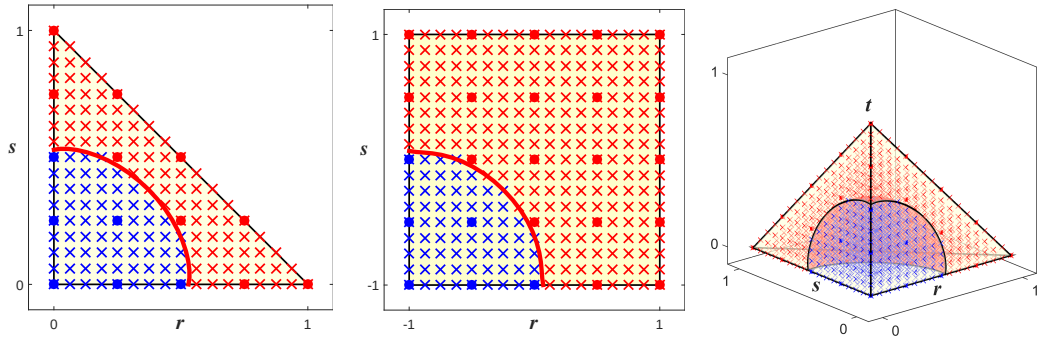
(iv) ensure that the resulting elements are valid in terms of the Jacobian of the implied element mapping.

Because some of these requirements may not be fulfilled for arbitrary $\phi^h(\boldsymbol{r})$ in *all* elements, a reliable error detection in combination with recursive refinement of some elements is a crucial ingredient of the proposed procedures. Upon (automatic) refinement even highly complex situations for zero level sets in a reference element can be resolved successfully. Observe that due to an automatic refinement, hanging nodes appear and therefore the reconstructed interface mesh may not be suitable to directly perform simulations. Here, its only purpose is to define an appropriate mapping for the integration points, either on the interface or in the two sides of the cut element.

4.2 Detection of cut elements

An element is considered cut, when the data at a finite number of points indicates that the approximated zero level set function ϕ^h intersects the element domain. The level set data is given at the nodal points of the physical element $\phi^h(\boldsymbol{x}_i)$ (or indistinguishably in the reference element $\phi^h(\boldsymbol{r}_i)$), but this information alone is not sufficient to determine the topological situation conclusively. Due to the fact that Lagrange shape functions are in general negative in parts of the

reference element, as depicted in Fig. 7, it is possible that an element is cut by the zero level set even if all $\phi^h(\boldsymbol{r}_i) > 0$. Instead, to decide whether the element is cut, a sample grid with an user defined resolution is introduced in the parent element, see Fig. 24. The set of grid points inside the element is called I^{Ω}_{grid}, on the element boundary as I^{Γ}_{grid} and the grid (nodal) positions are denoted as $\boldsymbol{r}_i^{grid}$. The distribution of the grid points is, in principle, arbitrary and can also be adjusted individually for each element if necessary. In general, the resolution of this auxiliary grid can be chosen as a free parameter, however, it should depend on the number of element nodes. For the calculations performed in this work three to five points between element nodes of the background mesh in each dimension were chosen. Exemplary sample grids for different reference background elements are depicted in Fig. 24.



(a) 2D, triangle (b) 2D, quadrilateral (c) 3D, tetrahedron

Figure 24: Sample grids for different reference background elements of order $m_\Omega = 3$. The circles are element nodes and red/blue crosses are grid points in I_{grid}.

Using the element functions and the nodal values $\phi^h(\boldsymbol{r}_i), i = \{1, ..n_\Omega\}$, the level set function is interpolated at all grid points j by

$$\phi_j^{\text{grid}} = \sum_{i=1}^{n_\Omega} N_i^\Omega(\boldsymbol{r}_j^{\text{grid}}) \cdot \phi_i \qquad \forall j \in I_{\text{grid}},$$

which is evaluated efficiently in terms of a matrix-vector product $\boldsymbol{\phi}^{\text{grid}} = \mathbf{A} \cdot \boldsymbol{\phi}$. The matrix $A_{ij} = N_i^\Omega(\boldsymbol{r}_j^{\text{grid}})$ is constant for a particular reference background element of order m_Ω with a specified resolution of the sample grid. As a consequence, the matrix entries A_{ij} can be determined conveniently in a pre-processing step. The set of grid points assigned to an element is defined as I_{grid}. The check for cut elements is carried out by an inspection of the sign of the level set values at the grid points. The considered element is cut, if

$$\min_{i \in I_{\text{grid}}} (\phi_i) \cdot \max_{j \in I_{\text{grid}}} (\phi_j) \leq 0. \tag{4.1}$$

It is recalled, that it is *not* sufficient to examine only the level set data ϕ_i at the element nodes $\boldsymbol{r}_i^\Omega$ to judge whether an element is cut. We note that also the approach based on a sample grid is not completely failsafe: For example, depending on the zero level set a case might occur where a very small region Ω^+ *inbetween* grid points is not detected as all grid points feature a negative sign. Nevertheless, the sample grid approach performs much better than only looking at the

element nodes and the (easily specified) grid density scales with the smallest features that can be resolved, i.e. the size of regions and zero level sets that possibly remain unnoticed.

It is important to note that in contrast to [73], we allow zero values of the level set function at the element nodes, $\phi_i = 0$. This approach introduces additional element topologies and therefore increases the implementational complexity. If this is not desired a small random perturbation of the element level set data in the range of 10^{-14} might be applied, which will not noticeable influence the resulting integration. Alternatively, it is possible to change the geometry of the background mesh by moving the element nodes away from the zero level set ϕ^h. There are different ways to perform the nodal shifting, this is discussed further in Section 6.

4.3 Valid level set data

Arbitrary level set data in higher-order elements may result in very complex patterns of the zero level set and the implied region where $\phi(\boldsymbol{x})$ is positive or negative, respectively. Thus, it is crucial to impose restrictions on the complexity of the level set data. Herein, we call the level set data in a cut element *valid* according to Eq. (4.1) if the following applies:

(i) The element boundary, that is, the element edge, is cut exactly *two* times in 2D. It is possible to generalize this assumption to multiple cuts for some special topologies, as for example three or four cuts on the element boundary, however, this results in a rather tedious implementation. In three dimensions the situation gets slightly more complicated. The zero isoline is then a closed polyline on the element boundary, and cuts always more than one face. For each face the conditions for 2D elements apply, specifically, each cut face is separately subdivided in two sub-domains. In terms of cut edges, we allow the tetrahedron to be cut three or four times at the element edges, which implies two different topological situations. The case of hexahedra is not considered, as hexahedra are for our purposes always subdivided into tetrahedra.

(ii) The zero level set divides the element domain $\Omega^h_{\boldsymbol{r}}$ into *two* sub-domains. This applies for all element types and all dimensions.

(iii) If all grid boundary nodes have identical signs, that is, if no edge in 2D or no face in 3D is cut, the element is completely uncut.

The above-mentioned conditions are now examined more closely. The focus is on a discrete version of the zero level, meaning that the inspection is based on the signs of the discretized zero level set on the sample grid.

Fig. 25(a)-(c) shows valid level set data subdividing miscellaneous reference elements in 2D into two sub-domains. For fairly typical geometries and level set functions describing rather simple objects with low curvature, the discussed conditions are sufficient. But for more complex or/and evolving geometries level set patterns cases may occur that are invalid according to the definition above. Some examples of zero level sets where our conditions are not met are shown in Fig. 26(a)-(c). When the level set data is invalid, there are two options to proceed, which depends on the intended use of the reconstructed interface elements. We distinguish the following cases:

(i) The reconstruction scheme is used to obtain higher-order accurate integration points on the zero level set or the cut elements. Thereby, simple recursive refinements finally yield valid

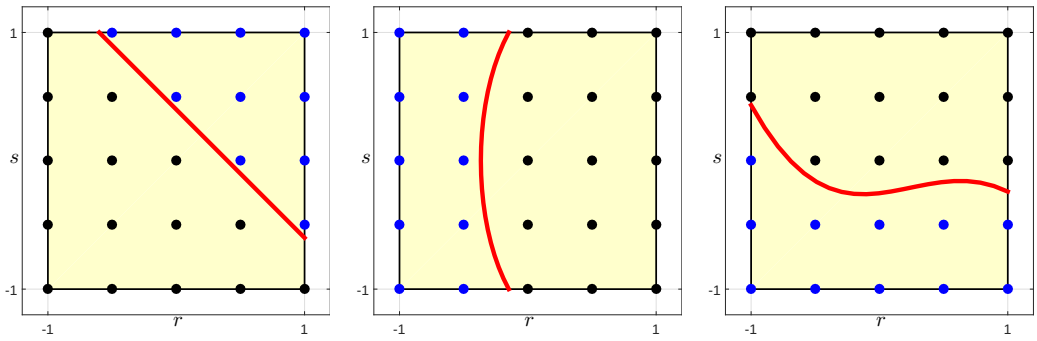


Figure 25: (a) to (c) show zero-isolines in a reference element belonging to valid level set data. Note that decent curvatures are allowed, as shown in (c).

level set data. The invalid coarse higher-order element is replaced by finer higher-order elements with the same order, whereupon the level set data at the element nodes is determined by interpolation from the coarser element. The recursive treatment is completely based on the level set data ϕ^h in the coarsest (the original) reference element, so that the explicit function $\phi(\boldsymbol{x})$ is never needed. It is important to emphasize that this approach results in a mesh with hanging nodes. These nodes will not pose any problems though, as the interface elements will not appear explicitly in the system of equations, but are used for integration purposes only.

(ii) The reconstruction scheme is used to construct a mapping that will finally result in a conforming fitted mesh. Contrary to the integration scheme a recursive refinement of isolated elements is not allowed, as the emanating hanging nodes demand special treatment in the finite element formulation. Instead we assume that either a local refinement, which influences only the direct neighbors, or alternatively a global (or adaptive local) refinement will resolve the situation and we end up with valid (conforming) elements only.

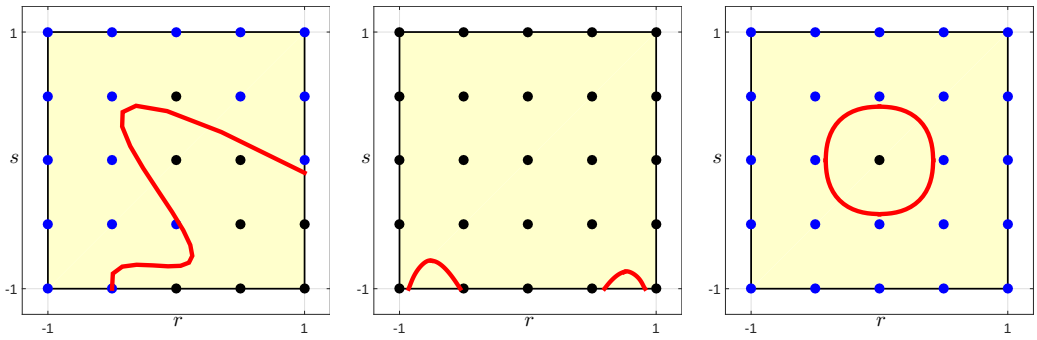


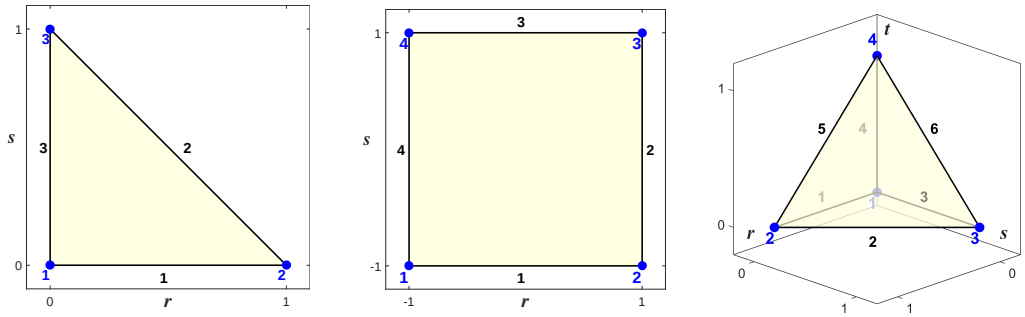
Figure 26: zero level sets in reference elements are invalid according to our criteria: in (a) the curvature is too strong, in (b) the zero level function produces more than two sub-domains, and in (c) the sub-domain can not be detected based on the boundary values (and in addition the curvature is too strong),

Even valid level set data has to meet some additional restrictions regarding the curvature of the zero level set: If changes in the orientation are too strong (e.g. Fig. 26(a)), the reconstruction may fail. Hence, the error treatment must be able to detect failed reconstructions and start a recursive refinement even if the level set data is valid in the previous sense. A situation violating condition (ii) is shown in Fig. 26(b). Here the zero level set function changes more than two times over the element edges, producing more than two separate domains. Inclusions as seen in Fig. 26(c) violate condition (iii), hence, the level set data is not valid. We note that a (locally) refined background mesh always produces valid level set data that meets the conditions from above.

4.4 Element topologies

From now on, it is assumed that the element under consideration is cut and features valid level set data according to Section 4.3. The topological situation in the elements is restricted to a small number of uniquely identifiable cases, that may be distinguished based on the sign of the level set data *at the set of boundary nodes* I^{Γ}_{grid} of the reference background elements of order m_{Ω}.

Regarding the nomenclature of the following topology classifications, some abbreviations are introduced. For the subdivision we consider triangular, quadrilateral and tetrahedral elements. A generalization to prism and hexahedral elements is possible, however, for our purposes not essential. The first letter describes the type of element. This can be either $\mathbf{T}$ for **t**riangular, $\mathbf{Q}$ for **q**uadrilateral elements or $\mathbf{P}$ for tetrahedral elements (**p**olyhedron with four faces) followed by a combination of subscripts and/or a superscripts. The index of the subscripts gives the information which *edges* are cut, whereas the superscripts describes which *nodes* are cut. So $\mathbf{Q}_{IJ}$ denotes a quadrilateral, with a zero level set which cuts edge $I \leq 4$ and edge $J \leq 4$. Another example is $\mathbf{T}^J_I$, describing a triangular element where one edge $I \leq 3$ and one node $J \leq 3$ are cut. The same notation is also valid for the tetrahedron, particularly, $\mathbf{P}_{IJK}$ is a tetrahedron with a zero level set which cuts edges $I \leq 6$, edge $J \leq 6$ and edge $K \leq 6$. The number of the subscripts and superscripts for 2D elements is always two, whereas in 3D the tetrahedron can have three or four indices, depending on the topological situation. The edge and node numbering are evident from Fig. 27. Note, that even in case of higher-order elements only the vertex nodes are relevant for the subdivision scheme.



(a) Linear triangular element (b) Bi-linear quadrilateral element (c) Tri-linear tetrahedral element

Figure 27: Edge and node numbering for linear (a), bi-linear (b) and tri-linear (c) Lagrange elements.

The introduced notation has the advantage that it indicates directly the number of possible topologies. As mentioned in Section 4, a valid cut 2D element has exactly two intersections of the zero level set along the whole element boundary. As a result, all possible cut situations for triangles are described by $\mathbf{T}_I^J$, $\mathbf{T}_{IJ}$ and $\mathbf{T}^{IJ}$, with $I \leq 3$ and $J \leq 3$. The same is valid for quadrilaterals, $\mathbf{Q}_I^J$, $\mathbf{Q}_{IJ}$ and $\mathbf{Q}^{IJ}$, with $I \leq 4$ and $J \leq 4$. It is worth recalling that a symmetry in the indices holds, namely, $\mathbf{T}_{IJ} = \mathbf{T}_{JI}$, which changes only the orientation of the normal vector but not the topological situation itself. As a consequence, the permutations of I and J for subscripts and superscripts cover all possible topological situations. The only invalid situation is *one* node that is cut twice by the level set. In the proposed notation that would result in $\mathbf{T}^{II}$ and $\mathbf{Q}^{JJ}$with $I \leq 3$ and $J \leq 4$. The generalization to three dimensions is reasonably straightforward, with the difference that the number of edges in 3D is not equal to the number of nodes, therefore different indices for sub- and superscripts are necessary. Example topologies are then e.g. $\mathbf{P}_{IJK}$, $\mathbf{P}_{IJKL}$, $\mathbf{P}_I^{MN}$ and $\mathbf{P}^{MNQ}$, with $I, J, K, L \leq 6$ for subscripts and $M, N, Q \leq 4$ for superscripts. Again, a node that is cut twice is not allowed, e.g. $\mathbf{P}^{MMQ}$, and an edge can be cut at most twice, e.g. $\mathbf{P}_{IIK}$. Also the symmetry is clearly not valid in all indices, $\mathbf{P}_{IJKL} \neq \mathbf{P}_{IKJL}$, now all permutations are valid as long as the sequence of the indices remains the same, e.g. $\mathbf{P}_{IJKL} = \mathbf{P}_{JKLI} = \mathbf{P}_{KLIJ} = \mathbf{P}_{LIJK}$. This can conveniently be described using the Levi-Civita symbol, which is for three indices, e.g. $\mathbf{P}_{IJK}$ defined as

$$\epsilon_{IJK} = \begin{cases} +1, & \text{if } (I,J,K) \text{ is } (1,2,3),(2,3,1),(3,1,2) \\ -1, & \text{if } (I,J,K) \text{ is } (3,2,1),(1,3,2),(2,3,1) \\ = 0 & \text{if } (I=J) \text{ or } (I=K) \text{ or } (J=K) \end{cases} . \tag{4.2}$$

If $\epsilon_{IJK} = 1$, the topology is valid. With a little effort this approach can be generalized to four indices.

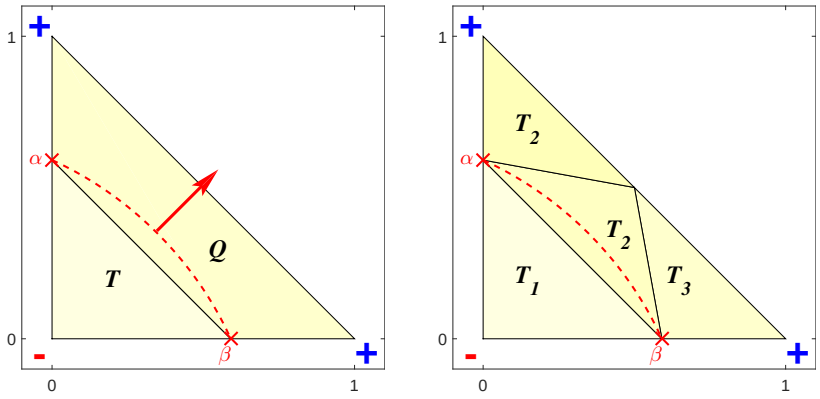
4.5 Subdivision strategy for triangles

A configuration is defined as "local", if the element is subdivided without influencing neighbour elements, which is the standard procedure if only the integration is of interest. For triangular elements there are two *local* configurations, that affect only the considered cut element. Another three *non-local* configurations are present, influencing the subdivision of *one* neighbour element. This particular strategy is designed for remeshing. The non-local scheme works successively which means that the (non-local) subdivision scheme produces a local configuration. Once this is done, the local subdivision algorithm yields the desired element patches, where some element nodes coincide with the reconstructed interface element.

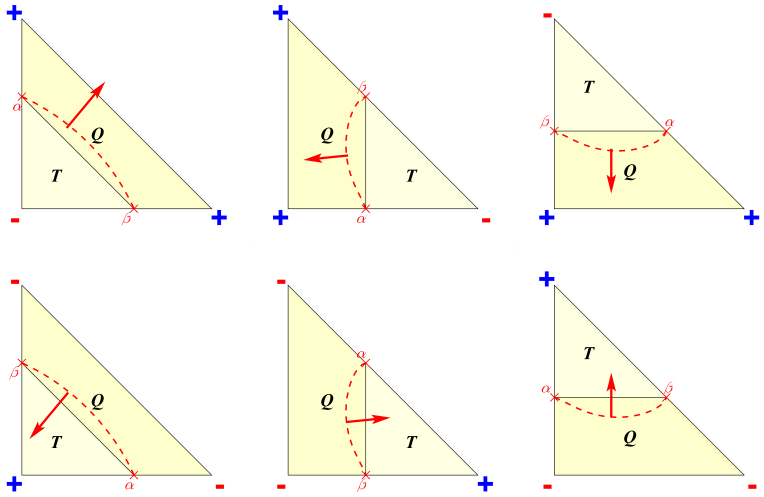
4.5.1 Local subdivisions

The subdivisions, which do not affect the neighbour elements or do not need any information from the neighbour elements to produce a unique subdivision, are referred to as *local subdivisions*. We start with inspecting the sign changes of the level set function at the three element edges, in fact, we inspect the level set function in the sample grid on the element boundary. First, we can identify the uncut case, namely, the boundary values indicate that the zero level set is not within

the considered element. Note, that the actual sign (positive or negative) of the boundary values is completely irrelevant, we denote this situation further on as $\mathbf{T}_0$. For the cut cases, we assume that exactly two boundaries are cut, one time each. Hence, one may distinguish $3^1 = 3$ different topological situations, depending on the combinations of the cut element boundaries. In total it amounts to 6 different topological cases, one exemplary topology is shown in Fig. 28(a). Due to the two cut edges, the initially triangular reference element is cut into one triangle element T_1 and one quadrilateral Q_1. The topological situation in the figure is labeled as $\mathbf{T}_{13}$ according to the introduced notation, however, notice that all situations fulfilling $\mathbf{T}_{IJ}$, $I \leq 3, J \leq 3$ and $I \neq J$ are also included in this topological situation, see Fig. 28(c). For some applications, especially for integration purposes, it can be useful to further sub-divide the quadrilateral Q_1 into three triangles, see Fig. 28(b). It is important to note that some of the edges of the sub-elements coincide with the (approximated and curved) isoline in this element; this will further be discussed below.



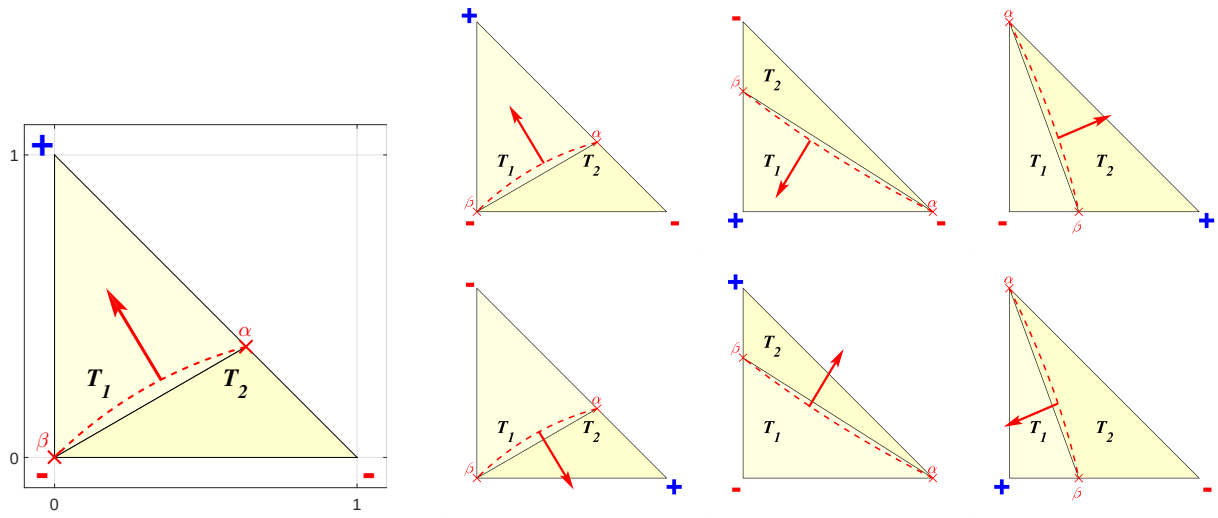
(a) $\mathbf{T}_{13}$ (b) $\hat{\mathbf{T}}_{13}$



(c) Cut scenarios for the topology with two cut edges.

Figure 28: Local subdivisions for triangles, here as an example topology $\mathbf{T}_{13}$, including the zero level set ϕ^h when two different edges are cut in (a). The zero-isoline (of valid level set data) always cuts the reference triangle element into one triangle T_1 and one quadrilateral Q_1, both featuring a curved side. (b) For integration purposes, it is sometime useful to further sub-divide the quadrilateral into 3 triangles. (c) shows all possible cut situations for combination of signs of ϕ_i at the three element edges.

The configurations in Fig. 28(c) and the introduced subdivision scheme apply to the case that both points of the zero level set are within the element edges, but not directly on the vertex. In case that one edge and the opposite vertex node are cut, as shown in Fig. 29(b), we introduce the second local subdivision type. The particular topology in Fig. 29(a) is labeled as $\mathbf{T}_2^1$, but again, the course of action covers in total the combinations $\{\mathbf{T}_2^1, \mathbf{T}_3^2, \mathbf{T}_1^3\}$.



(a) $\mathbf{T}_2^1$ (b) $\mathbf{T}_2^1$

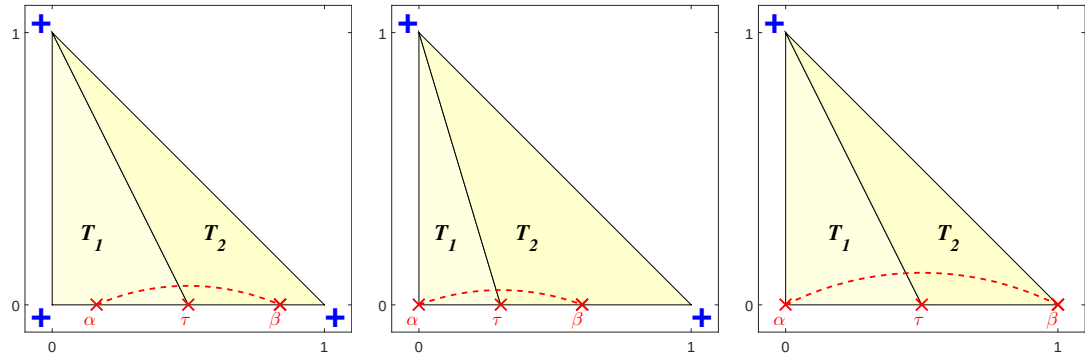
Figure 29: Second type of local subdivisions for triangles, in (a) as an example topology $\mathbf{T}_2^1$, when one vertex and the opposite edge is cut by ϕ^h. (b) shows all possible cut situations for different signs of ϕ_i at the three corner nodes.

A normal vector $\boldsymbol{n}$ pointing from the negative region Ω^- towards the positive region Ω^+ is associated with the isoline. This yields a natural starting point α and end point β of the isoline in the reference element which, furthermore, implies a natural orientation/numbering of the one-dimensional higher-order element that is reconstructed on the isoline. Note, that the normal vector does not influence the node numbering of sub-elements as shown in Fig. 28(c) and Fig. 29(b). For the integration of implicit volumes, see Section 5, this particular numbering is important for the mapping of coordinates from the interface to the reference elements, which is implied by the special sub-elements with one higher-order side.

4.5.2 Non-local subdivisions

Some topologies are considered that need a specialized treatment as they influence the neighbour elements. These particular situations occur in connection with a glancing intersection of the level set with one element edge. If only the integration is considered, the problem can be resolved by a local refinement of the background mesh. Alternatively a global refinement is be conceivable. However, from a computational point of view, it is still preferable to proceed as follows. We go through all elements of the mesh and find the intersections of the level set function with the element boundaries. For edges that are cut twice, the adjacent elements are subdivided as in Fig. 30(a). The element is subdivided in the middle of the interval defined by the two roots on the edge, $\boldsymbol{r}_M = \frac{1}{2}(\boldsymbol{r}_A + \boldsymbol{r}_B)$. Collectively, the topological combinations $\{\mathbf{T}_{11}, \mathbf{T}_{22}, \mathbf{T}_{33}\}$ are covered by this subdivision.

The two other cases are depicted in Fig. 30(b) where a node is cut together with a neighbouring edge, described by $\{\mathbf{T}_1^1, \mathbf{T}_1^2, \mathbf{T}_2^2, \mathbf{T}_2^3, \mathbf{T}_3^3, \mathbf{T}_3^1\}$, and Fig. 30(c) where two nodes are cut, described by $\{\mathbf{T}^{11}, \mathbf{T}^{22}, \mathbf{T}^{33}\}$. The subdivision strategy is similar as before: Both cases can be subdivided



(a) $\mathbf{T}_{11}$ (b) $\mathbf{T}_1^1$ (c) $\mathbf{T}^{12}$

Figure 30: Non-local subdivisions of a triangle as a consequence of (a) one edge cut twice, (b) one cut node with a neighbouring cut edge and (c) two cut nodes.

into two triangles, with the point $\boldsymbol{r}_M$ located in the middle of the intersections of the zero level sets with the element boundary.

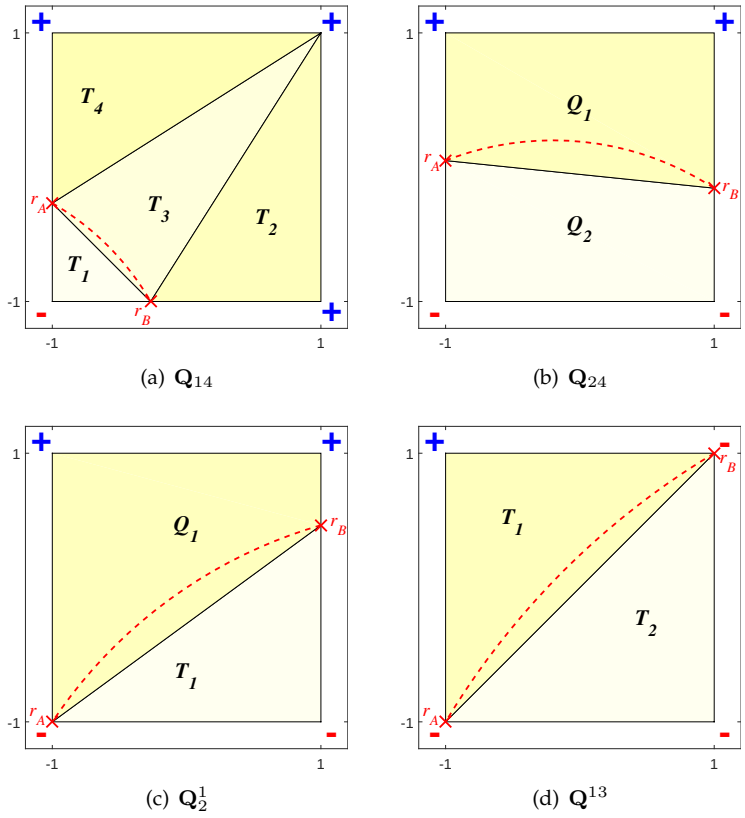
Once the non-local subdivisions are executed, all topological cases can be treated by the local subdivision strategy described before.

4.6 Subdivision strategy for quadrilaterals

Starting with a mesh composed by quadrilateral elements, one approach would be to convert the cut elements into triangles and apply the scheme for triangles from above. However, this would obviously lead to (slightly) different zero level sets compared to those implied by the interpolation functions in a quadrilateral element. Besides, significantly more elements arise and diminish the efficiency of the scheme. Therefore, we rather develop schemes that work without prior conversion. A total of four *local* subdivisions are introduced, depending on the cut situation. As before, it is also necessary to consider cases where the neighbour elements are influenced, which are again specified as *non-local* subdivisions.

4.6.1 Local subdivisions

In quadrilateral elements there are four clearly distinguishable topological situations. All situations featuring two cut *adjacent* edges are topologically equivalent, see Fig. 31(a) for $\mathbf{Q}_{14}$. The subdivision for this topological case produces four triangular elements. All other topological cases, $\{\mathbf{Q}_{12}, \mathbf{Q}_{23}, \mathbf{Q}_{34}\}$, are equally covered with the proposed subdivision. Another case is when two *opposite* edges are cut, as depicted in Fig. 31(b) for $\mathbf{Q}_{24}$. This subdivision creates two quadrilaterals. Again, the rotated situation $\mathbf{Q}_{13}$, is likewise covered by the subdivision. The first topology considering nodal cuts is depicted in Fig. 31(c). Therein, a node and a non-adjacent edge are cut and subdivide the original quadrilateral into a quadrilateral and a triangle. The covered situations are $\{\mathbf{Q}_2^1, \mathbf{Q}_3^1, \mathbf{Q}_3^2, \mathbf{Q}_4^2, \mathbf{Q}_4^3, \mathbf{Q}_1^3, \mathbf{Q}_1^4, \mathbf{Q}_2^4\}$. The last local subdivision is depicted in Fig. 31(c). This topology features two cut opposite nodes, $\{\mathbf{Q}^{13}, \mathbf{Q}^{24}\}$, leading to two triangles. The same comments on the normal vector and the node numbering of the sub-elements from above apply.



(a) $\mathbf{Q}_{14}$ (b) $\mathbf{Q}_{24}$

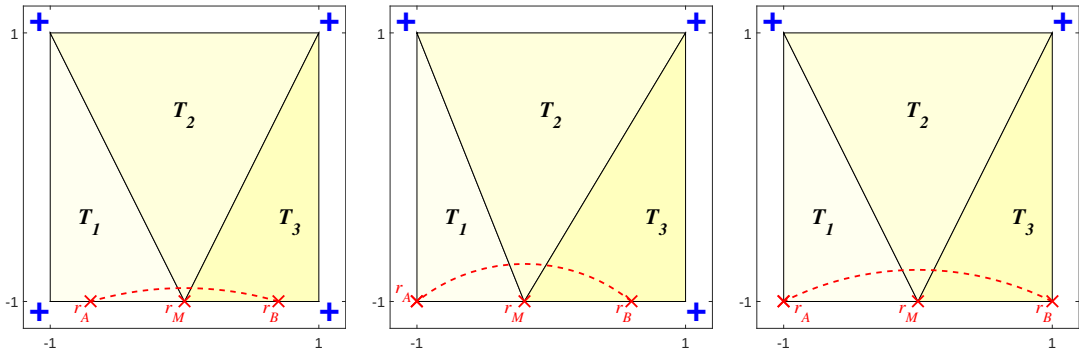
(c) $\mathbf{Q}_2^1$ (d) $\mathbf{Q}^{13}$

Figure 31: All variants of local subdivision schemes for a quadrilateral element: (a) features two neighbouring cut edges, (b) two opposite cut edges, (c) a cut vertex and a non-neighbour cut edge, and (d) two cut opposite vertices.

Again, all possible variations of sign changes are covered by simple mappings and rotations as demonstrated for the triangular topologies.

4.6.2 Non-local subdivisions

To deal with glancing intersections, as depicted in Fig. 32(a), it is necessary to subdivide the elements adjacent to the considered edge. In total, as for the triangular elements, three different topological cases arise. All situations have in common that one edge is cut twice. The first case, see Fig. 32(a), includes the topologies $\{\mathbf{Q}_{11}, \mathbf{Q}_{22}, \mathbf{Q}_{33}\}$. The subdivision of this topological case is executed by three triangles, whereby the triangles share the point $\boldsymbol{r}_M = \frac{1}{2}(\boldsymbol{r}_A + \boldsymbol{r}_B)$. All other topological cases are similar to the former one when placing one root, as in Fig. 32(b), or both roots on the vertex nodes, see Fig. 32(c). The covered situations are then $\{\mathbf{Q}_1^1, \mathbf{Q}_1^2, \mathbf{Q}_2^2, \mathbf{Q}_2^3, \mathbf{Q}_3^3, \mathbf{Q}_3^4, \mathbf{Q}_4^4, \mathbf{Q}_4^1\}$ for one cut vertex and $\{\mathbf{Q}^{11}, \mathbf{Q}^{22}, \mathbf{Q}^{33}\}$ for two cut vertices.



(a) $\mathbf{Q}_{11}$ (b) $\mathbf{Q}_1^1$ (c) $\mathbf{Q}^{12}$

Figure 32: Non-local subdivisions of a quadrilateral with (a) one edge cut twice, (b) a cut node next to a cut edge, (c) two cut neighbouring nodes.

4.7 Topologies in cut tetrahedral elements

In three dimensions, a very large number of cases would arise if level set values are zero at the corner nodes. Therefore, we restrict ourselves to the case, where $\phi_i \neq 0$ is enforced, e.g. by adding slight perturbations to the level set values. Therefore, for tetrahedral reference elements we end up with four local subdivisions, two of them with cut edges only.

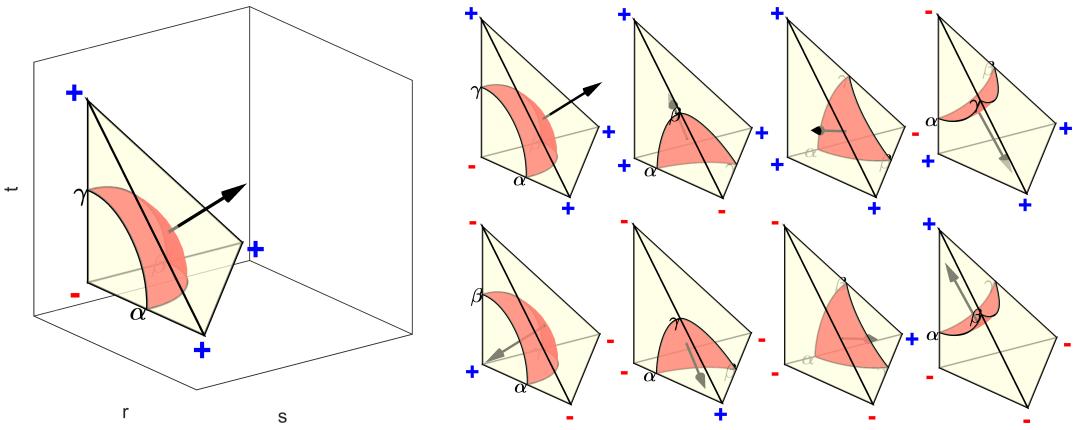
4.7.1 Local subdivisions

In order to check for cut elements only the element edges are inspected. Similar to the 2D cases, a constant sign on *all* element edges, irrespective whether positive or negative, indicates an uncut element. Two topologically different cases result for the case of cut element edges only: For topology 1, see Fig. 33(a), four edges of the element are cut. The tetrahedral reference element is decomposed into one tetrahedron T_1 and one prism P_1 which share a curved triangular face with corner points (α, β, γ) coinciding with the reconstruction of the zero isosurface. The 8 related cases for this topology are shown in Fig. 33(b). Topology 2 features two nodes with the same sign than the other two, see Fig. 34(a); the 6 related cases are shown in Fig. 34(b). In this topological case, the tetrahedron falls into two prisms P_1 and P_2 which share a curved quadrilateral face $(\alpha, \beta, \gamma, \delta)$ at the zero-isosurface.

Again, a normal vector $\boldsymbol{n}$ on the zero-isosurface points from Ω^- towards Ω^+. This yields a natural orientation for the higher-order interface element which determines the sequence of the corner nodes of the reconstructed triangle (α, β, γ) for case 1 or quadrilateral $(\alpha, \beta, \gamma, \delta)$ for case 2. Fig. 33(b) and Fig. 34(b) explicitly show the normal vectors and the related signs of the corner nodes for all cases. Note the particular node numbering of the sub-elements T_i and P_i as shown in Fig. 33(a) and Fig. 34(a) which is later important for the volume integration in Section 5.

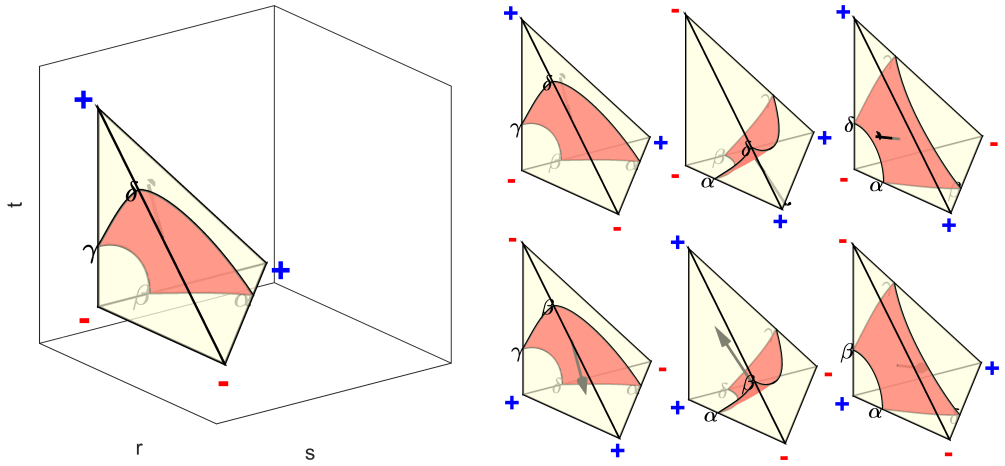
4.8 Topologies in cut hexahedral elements

In hexahedra, a very large number of topological cases arises even for valid level set data. The number of possible topologies can be computed in a similar fashion as for tetrahedra by inspect-



(a) Topology 1. (b) Cut scenarios.

Figure 33: (a) If one of the four corner nodes has a different sign than the other three, the zero-isosurface cuts the reference tetrahedral element into one tetrahedron T_1 and one prism P_1. (b) shows all possible cut situations that lead to topologically equivalent situations.



(a) Topology 2. (b) Cut scenarios.

Figure 34: (a) If two of the four corner nodes have a different sign than the other two, the zero-isosurface cuts the reference tetrahedral element into two prisms P_1, P_2. (b) shows all possible cut situations that lead to topologically equivalent situations.

ing sign changes on the element edges. Concrete examples of resulting topological cases are described in the context of marching cubes [119] and are beyond the scope of this work. One may completely avoid the cumbersome and implementationally lengthy treatment of possible cut situations in hexahedra by decomposing each hexahedron into tetrahedra and applying the reconstruction as for tetrahedra as described above. Using Lagrange elements enables a direct transfer from higher-order hexahedral reference elements to higher-order tetrahedra without any projection.

4.9 Notes on implementational issues

In this section, depending on the location of the zero isoline, we dealt with a number of different topological cases for each element type. In order to simplify the implementation, it is advisable to implement the topological cases based on one *master topology*. We take as example the triangular elements. The master topology based on the boundary information is chosen as in Fig. 28(a) and all other topological situations, that is, the other topological cases in Fig. 28(b), result from simple transformations. The transformations are restricted either to isoparametric mappings (in case of rotational invariance) or/and to a reversal of all signs on the element boundary to adjust the normal vector.

4.10 Reconstruction of zero level set by Newton-Raphson procedure

We proceed, assuming that an element is cut based on valid level set data on the element boundary, and features a topology according to the defined cases in Section 4.3. The task is to approximate the zero level set in the reference element with an interface element, in which the order m_Γ of the interface element is user specified. A number of nodes $\boldsymbol{r}_i$ are hereby assigned to any interface element, appropriate to the order m_Γ. The numbering must be such that a consistent normal vector $\boldsymbol{n}$ pointing from Ω^- to Ω^+ is obtained. This is achieved when the corner nodes of the higher-order interface elements match the nodes $\alpha, \beta, \gamma, \delta$ as in Figs. 33 and 34.

For accuracy reasons, it is crucial that the element nodes $\boldsymbol{r}_i$ must be on the approximated zero level set ϕ^h, up to the computational precision. This is equivalent of finding the root of a scalar non-linear function, and naturally an iterative procedure is needed. The following requirements have to be fulfilled by the iteration method:

(i) Given a start position inside the element, the point *must not* leave the reference element during the iteration procedure.

(ii) The iteration must maintain the topology that is intended by the definition of the starting points.

(iii) The iteration must be efficient and robust.

The first item suggests that search paths can be provided with any starting point; the choice will, however, influence the interpolation properties of the final interface element. For each topology of the cut scenario, different starting points and associated search paths may be provided, which guarantee that the zero level set is intersected. The second item proposes that the search paths in terms of starting points and direction should be as close as possible to the zero level set to

avoid invalid topologies or too many recursive refinements. The last item suggests that a scheme should converge quadratically, such as the Newton–Raphson family of methods or at least close to this for any other choice.

It is noted that there are methods such as gradient ascent/descent methods [137], which do not require search paths but move "freely" towards a point on the zero level set. These methods were also tested; however, they cannot guarantee that the point stays inside the element and the convergence rate is only linear. Furthermore, the results indicate that the Newton-Raphson method performs very good in terms of stability and accuracy, therefore we recommend their use, in particular with tailored starting values and search paths.

4.10.1 Newton–Raphson scheme with prescribed search directions

The task is to find the root of a scalar function, i.e. the zero level set, of $\phi^h(\boldsymbol{r})$ along a path which is specified by a parametrized vector function $\boldsymbol{r}(q)$, $q \in \mathbb{R}$. Applying the standard Newton-Raphson approach to this type of problem and observing that $\phi^h = \phi^h(\boldsymbol{r}(q^i))$, gives

$$q^{i+1} = q^i - \frac{\phi^h}{\nabla_q \phi^h} \tag{4.3}$$

with

$$\nabla_q \phi^h = \nabla_{\boldsymbol{r}} \phi^h \cdot \nabla_q \boldsymbol{r} = \frac{\partial \phi^h}{\partial r} \cdot \frac{\partial r}{\partial q} + \frac{\partial \phi^h}{\partial s} \cdot \frac{\partial s}{\partial q} + \frac{\partial \phi^h}{\partial t} \cdot \frac{\partial t}{\partial q}. \tag{4.4}$$

Once some error measure of Eq. (4.3) falls under a user specified threshold, we set $\boldsymbol{r}_i^\Gamma = \boldsymbol{r}(q^k)$ for one of the element nodes. Here, we want to restrict the situation to finding the root along a *straight* line going through the point $\boldsymbol{P}$ with direction $\boldsymbol{N}$, hence

$$\boldsymbol{r}(q) = \boldsymbol{P} + q \cdot \boldsymbol{N}. \tag{4.5}$$

The search path is defined by the tuple consisting of the two vectors $\{\boldsymbol{P}, \boldsymbol{N}\}$, where $\boldsymbol{P}$ is any point on the straight path and $\boldsymbol{N}$ is the search direction. Due to the linearity of Eq. (4.5), the iterative procedure can be expressed directly in $\boldsymbol{r}$ as

$$\boldsymbol{r}^{i+1} = \boldsymbol{r}^i - \frac{\phi^h\left(\boldsymbol{r}^i\right)}{\nabla_{\boldsymbol{r}} \phi^h \cdot \boldsymbol{N}} \cdot \boldsymbol{N} \tag{4.6}$$

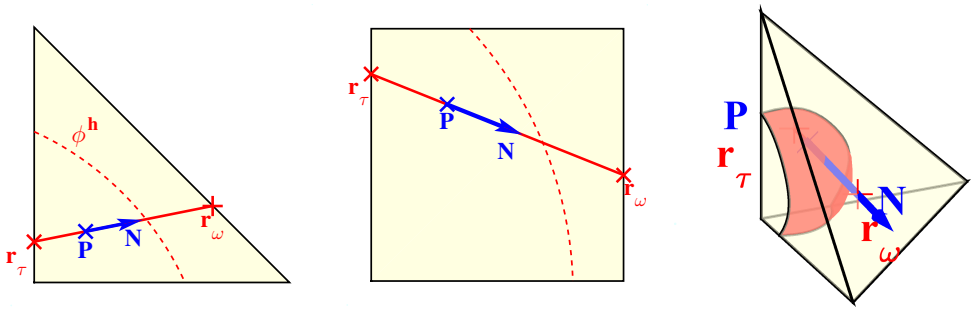
where $\boldsymbol{N} = [N_r, N_s, N_t]$. The derivatives of ϕ^h are easily obtained from Eq. (3.2) after differentiation of the shape functions with respect to $\boldsymbol{r}$. Again, we set $\boldsymbol{r}_i^\Gamma = \boldsymbol{r}^k$ after convergence of Eq. (4.6). It remains to define the straight search paths $\{\boldsymbol{P}, \boldsymbol{N}\}$ and the choice of starting points $\boldsymbol{r}^0$ for the iteration.

The process starts with finding roots on the element edge, the number depends on the dimension (always two in 2D, three or four in 3D) and the emerging topology. Then, a number of points $\boldsymbol{P}_i$, where i depends on the desired element order, are distributed on the *linear* approximation of the zero level set, which is the connection of the points found on the edges and gives a line in 2D and a plane in 3D. These points serve as the starting guesses for the iterative root-finding inside the element.

4.10.2 Starting points for the Newton-Raphson iteration

Assume that the limitations on the level set data from Section 4.3 are met and that a given search path $\{\boldsymbol{P}, \boldsymbol{N}\}$ is valid in that it intersects *once* with the zero level set inside the considered element. Then, every start guess $\boldsymbol{r}^0 \in \{\boldsymbol{P}, \boldsymbol{N}\}$ converges to the same root, i.e. the sought-after intersection point with the zero level set.

In the implementation it is important to specify the set of admissible bounds on the search path. As mentioned before, the iteration must not leave the element domain for any given search path $\{\boldsymbol{P}, \boldsymbol{N}\}$. Taking into account that the search path is restricted to remain on a straight line, the valid range is then given by the two intersection points α and β on the element edge and the element face for 2D and 3D, respectively, see Fig. 35. One may easily provide a valid start guess, if possible, close to the zero level set ϕ^h. If, for rare cases, the start guess is outside of the convergence radius of the Newton-Raphson method, that is, convergence is not achieved, another start guess may be chosen within the valid range. The default setting for consecutive start guesses (until a converging start guess is found) is to try (1) $\boldsymbol{r}_0 = {}^1\!/_2 \cdot (\boldsymbol{r}_A + \boldsymbol{r}_B)$, (2) $\boldsymbol{r}_0 = \boldsymbol{r}_A$, (3) $\boldsymbol{r}_0 = \boldsymbol{r}_B$, then several linearly spaced points between $\boldsymbol{r}_A$ and $\boldsymbol{r}_B$.



(a) Triangle (b) Quadrilateral (c) Tetrahedron

Figure 35: The search path $\{\boldsymbol{P}, \boldsymbol{N}\}$ defines a valid line interval in the element between the two intersection points $\boldsymbol{r}_\tau$ and $\boldsymbol{r}_\omega$ on the element boundary and the element face, respectively. The starting point $\boldsymbol{r}^0$ and the final root must be on this line interval.

Provided that the search path intersects the zero level set within the element, it is thereby almost guaranteed to find a converging start guess leading to the desired root. In the numerical results, failed cases, where the iteration Eq. (4.6) did not converge to a valid root, could always be traced back to invalid search paths, i.e. those where no valid intersection point inside the element existed. It is concluded that the definition of suitable search paths is a crucial matter for the stability in finding the $\boldsymbol{r}_i^\Gamma$. Furthermore, an error treatment should still be implemented which may initiate a (recursive) refinement of the element until valid $\boldsymbol{r}_i^\Gamma$ are obtained.

4.10.3 Search paths

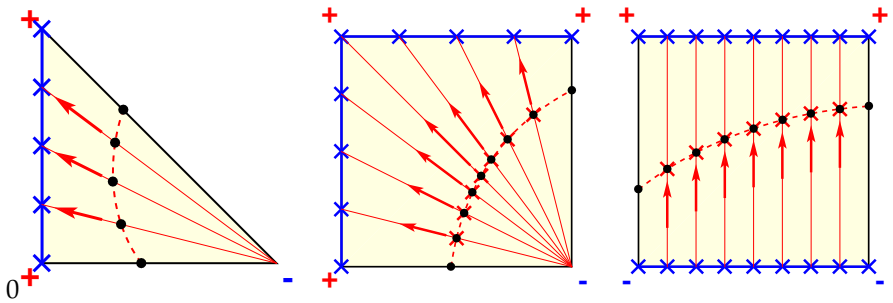
The focus is now on providing the $2-$tupel $\{\boldsymbol{P}_i, \boldsymbol{N}_i\}$, which defines the search path. After convergence of Eq. (4.6), they give the sought roots $\boldsymbol{r}_i^\Gamma$ of the level set function. These points $\boldsymbol{r}_i^\Gamma$ define a higher-order interface element which is an approximation of the exact zero level set of ϕ^h. It is later seen in the numerical results that the definition of $\boldsymbol{P}_i$ and $\boldsymbol{N}_i$ largely influences

the accuracy of the reconstructed zero level set and consequently the resulting integration. They are also critical for the robustness and the efficiency of the scheme as search paths which do not intersect with the zero level set within the element (excluding the element boundary) are not able to provide a valid root and a recursive refinement is prompted.

4.10.4 Search paths in two dimensions

In two dimensions, four variants of search paths are investigated.

In **variant (i)**, the search paths depend only on the topological classification, see Section 4.4, but *not* on the actual location of the zero level set within the element. If for triangular elements the topology includes two cut edges, auxiliary nodes $\boldsymbol{A}_i$ are evenly distributed along the uncut element edges, see the blue crosses in Fig. 36(a). The search directions $\boldsymbol{N}_i$ then point towards the one node with a different sign. The same is valid for quadrilateral elements and topology 1, see Fig. 36(b). For elements where two neighboring nodes have the same sign, but opposite from the other two (topology 2), the search directions $\boldsymbol{N}_i$ align with the $\boldsymbol{r}$-coordinates, depending whether horizontal or vertical edges are cut, see Fig. 36(c).



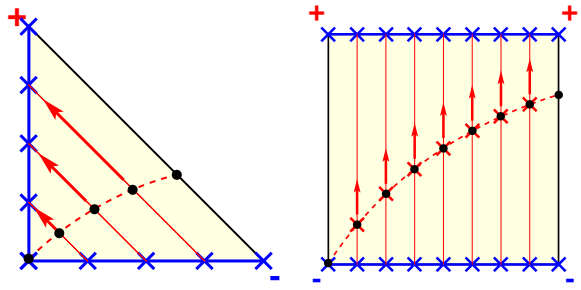
(a) Triangle (b) Quad, topology 1 (c) Quad, topology 2

Figure 36: Search paths for **variant (i)** and local topologies with cut edges only for 2D-elements.

For the other topologies, the ones including cut nodes, the auxiliary nodes $\boldsymbol{A}_i$ are distributed along both non-cut edges, see Fig. 37(a). For the quadrilateral element with cut nodes, the strategy is very similar to the triangular elements. Two edges are chosen in a way that (i) the edge itself must not be cut and (ii) the edge opposite to the chosen edge must also not be cut. For the example shown in Fig. 37(b) this leaves edges 1 and 3 to distribute the points $\boldsymbol{A}_i$.

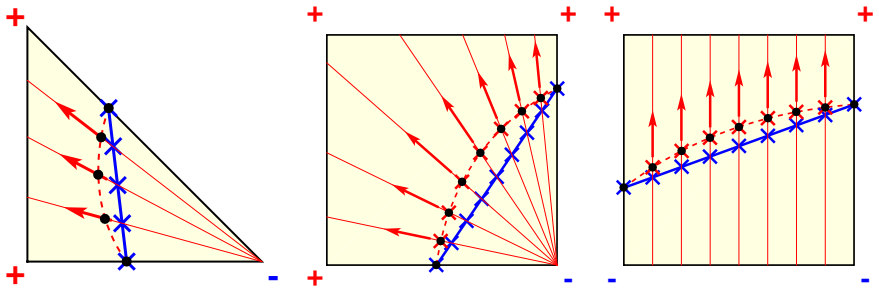
All other variants depend on a *linear reconstruction* of the zero isoline. In a first step, the roots are determined on the (two) cut element edges, these two nodes already represent the end nodes of the interface element. Their linear connection yields a rough approximation of the actual zero isoline. When n_Γ element nodes are sought on the isoline in one element, search paths are now required for the remaining $n_\Gamma - 2$ roots within the element. The points $\boldsymbol{A}_i$ are always obtained by distributing $n_\Gamma - 2$ equally spaced points on the linear approximation of the isoline. Variants (ii) to (iv) differ merely in the search directions $\boldsymbol{N}_i$.

In **variant (ii)**, for the topology of two cut edges, the search directions point towards the one uncut edge, see Fig. 38(a) and (b). For quadrilateral elements with two cut opposite edges (topology 2), the $\boldsymbol{N}_i$ are parallel to the $\boldsymbol{r}$-coordinates, depending whether horizontal or vertical edges are cut, see Fig. 38(c).



(a) Triangle (b) Quad, topology 1

Figure 37: Search paths for variant (i) and topologies including cut nodes for 2D-elements.



(a) Triangle (b) Quad, topology 1 (c) Quad, topology 2

Figure 38: Search paths for variant (ii) in 2D-elements.

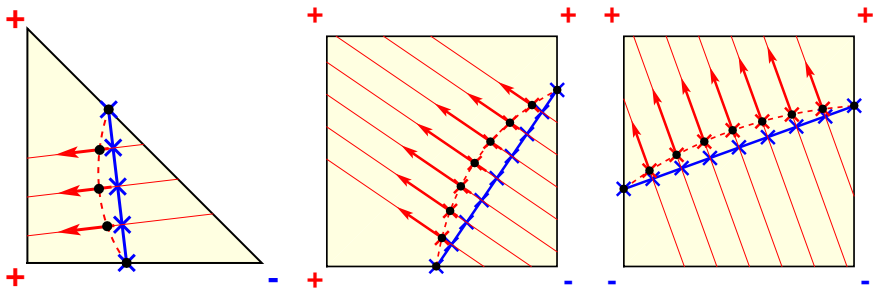
In **variant (iii)**, the $\boldsymbol{N}_i$ are equal for all points $\boldsymbol{A}_i$, specifically, the search directions are parallel. They are orthogonal to the straight line being the linear approximation of the zero isoline in an element. See Fig. 39 for a visualization of the search paths according to variant (iii).

In **variant (iv)**, the search directions $\boldsymbol{N}_i$ are based on the *gradient* of the discretized level set function at $\boldsymbol{A}_i$, i.e. $\boldsymbol{N}_i = \nabla\phi^h(\boldsymbol{A}_i)$, see Fig. 40.

4.10.5 Search paths in three dimensions

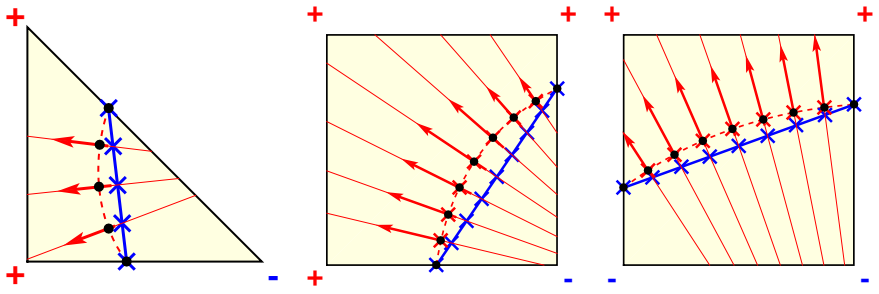
In order to reconstruct the zero isosurface in 3D tetrahedral elements, five variants are investigated herein. Essentially, the variants are the three-dimensional analogies to the two-dimensional variants suggested above, with some additional considerations regarding the mapping of the interface element.

In **variant (i)**, the search paths only depend on the topological situation, see Section 4.4, but *not* on the actual level set data within the element. If three linked non-planar edges are cut (or if one node has a different sign than the others (topology 1)), nodes $\boldsymbol{A}_i$ are projected from a reference triangle of the desired order m_Γ onto the face with no cut edges, see the blue crosses in Fig. 41(a). The search directions $\boldsymbol{N}_i$ then point towards the one node with a different sign. When in total four edges are cut (or alternatively two nodes share one sign which is opposite from the other two), the reconstructed interface element is a higher-order quadrilateral (topology 2). From the 6 edges of the tetrahedral element, two are not cut by the zero isosurface, i.e. their two end points



(a) Triangle (b) Quad, topology 1 (c) Quad, topology 2

Figure 39: Search paths for variant (iii) in 2D-elements.



(a) Triangle (b) Quad, topology 1 (c) Quad, topology 2

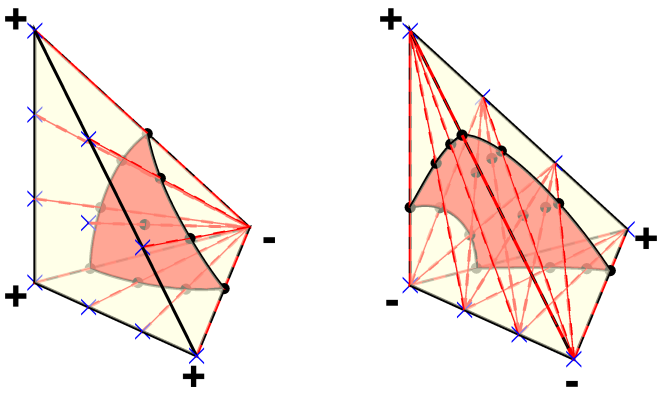
Figure 40: Search paths for variant (iv) in 2D-elements.

have the same sign each. Then, $(m_\Gamma + 1)$ points are distributed with equal spacing along each of these two edges. The search paths of the quadrilateral element with $n_\Gamma = (m_\Gamma + 1) \times (m_\Gamma + 1)$ nodes are determined by combining the edge nodes in a tensor-product fashion, see Fig. 41(b).

In **variant (ii)**, the search paths are determined by means of a low-order reconstruction of the zero isosurface. For topology 1, the roots of the level set function are determined on the three cut edges. Thereby, a linear triangle is defined onto which a (flat) higher-order triangle with the desired order is projected whose element nodes are the points $\boldsymbol{A}_i$. The search directions $\boldsymbol{N}_i$ then point towards the one node where the three cut edges intersect, see Fig. 42(a). For topology 2, a bi-linear quadrilateral is defined by the roots on the four cut edges. A higher-order quadrilateral element is projected onto this element. Then, each search path is the straight line going through the point on the bi-linear quadrilateral and intersecting with the two uncut edges, see Fig. 42(b).

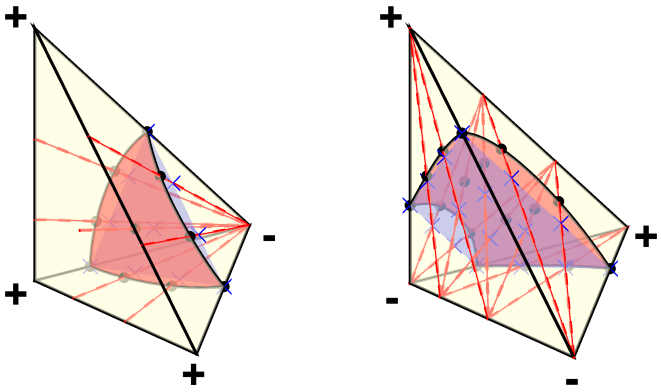
In analogy to the two-dimensional situation, there are two straightforward extensions each differing only in the search direction $\boldsymbol{N}_i$: One is where the search directions are the normal vectors at $\boldsymbol{A}_i$ on the low-order reconstruction of the interface. The other uses the gradient of the approximated level set function, i.e. $\boldsymbol{N}_i = \nabla\phi^h(\boldsymbol{A}_i)$. However, in the numerical studies it was found that these variants are not robust when they are based on the *linear* reconstruction. Frequently, the zero isosurface is not intersected by the search path, leading to fail cases (and consequently initiating a recursive refinement).

Therefore, three other variants are proposed. The idea is to split the root finding for (i) the edge (outer) nodes of the higher-order interface element that are on the faces of the tetrahedron



(a) Topology 1 (b) Topology 2

Figure 41: Search paths for variant (i) in 3D-tetrahedra.



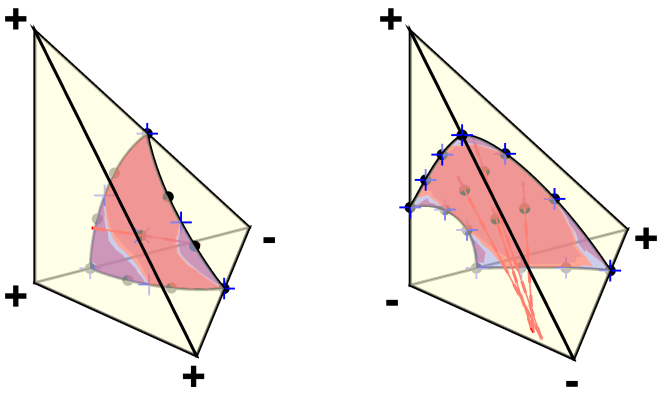
(a) Topology 1 (b) Topology 2

Figure 42: Search paths for variant (ii) in 3D-tetrahedra.

from (ii) those that are within the interface element (the inner nodes). For the nodes on the faces, the starting point is similar to the two-dimensional situation as discussed in Section 4.10.4: In fact, the task is to identify the zero-iso*line* by $(m_\Gamma + 1)$ points on each of the faces for which conveniently any of the discussed variants of Section 4.10.4 can be employed. For topology 1, there are 3 higher-order isolines because three faces are cut by the zero-isosurface. These three isolines provide a (curved) boundary of the higher-order triangle element to be reconstructed at the zero-isosurface. For topology 2, there are 4 isolines on the 4 cut faces providing a boundary for the higher-order quadrilateral element approximating the zero-isosurface.

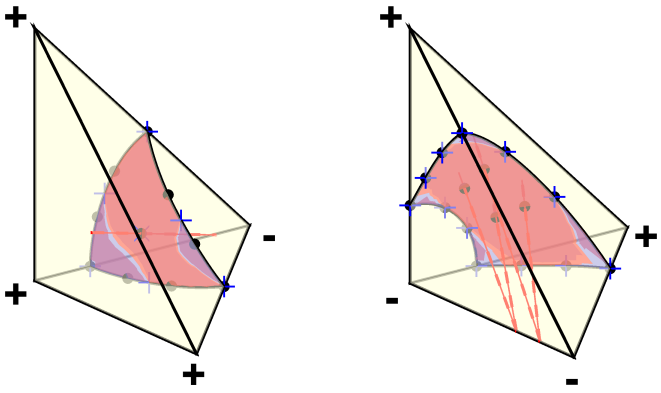
It now remains to discuss the search paths for the inner nodes of the interface element. For topology 1, a *quadratic triangle* element is reconstructed (not having any inner nodes) only based on the two-dimensional methods of the previous section. A higher-order element of the desired order m_Γ is then mapped onto this quadratic element in order to achieve the points $\boldsymbol{P}_i$. For topology 2, a quadrilateral *cubic Serendipity element* (again without any inner nodes) is reconstructed by the two-dimensional approaches applied to the respective faces. A higher-order quadrilateral element (of the Lagrange class) is then projected onto this cubic element to obtain $\boldsymbol{A}_i$.

Variant (iii) uses these points with search directions $\boldsymbol{N}_i$ that coincide with the normal vec-



(a) Topology 1 (b) Topology 2

Figure 43: Search paths for variant (iii) in 3D-tetrahedra. Only the search paths for inner nodes are shown. They are based on a quadratic triangle element for topology 1 and a cubic Serendipity element for topology 2, see the blue plus symbols in (a) and (b), respectively.



(a) Topology 1 (b) Topology 2

Figure 44: Search paths for variant (iv) in 3D-tetrahedra. Only the search paths for inner nodes are shown. They are based on a quadratic triangle element for topology 1 and a cubic Serendipity element for topology 2, see the blue plus symbols in (a) and (b), respectively.

tors at $\boldsymbol{P}_i$ on the preliminary reconstruction of the interface (by a quadratic triangle or a cubic Serendipity element), see Fig. 43. In **variant (iv)**, the search directions are the gradients of the level set function, $\boldsymbol{N}_i = \nabla\phi^h(\boldsymbol{P}_i)$. As seen in Fig. 44, the two variants can hardly be distinguished by visual inspection: The nodes on the faces are identical and the implied preliminary reconstruction by a quadratic triangle or a cubic Serendipity element gives already a very good approximation of the interface. Therefore, also the search directions of these two variants are almost identical. However, as will be shown later, both variants result in sub-optimal convergence rates for integration and approximations problems. Therefore another variant is introduced.

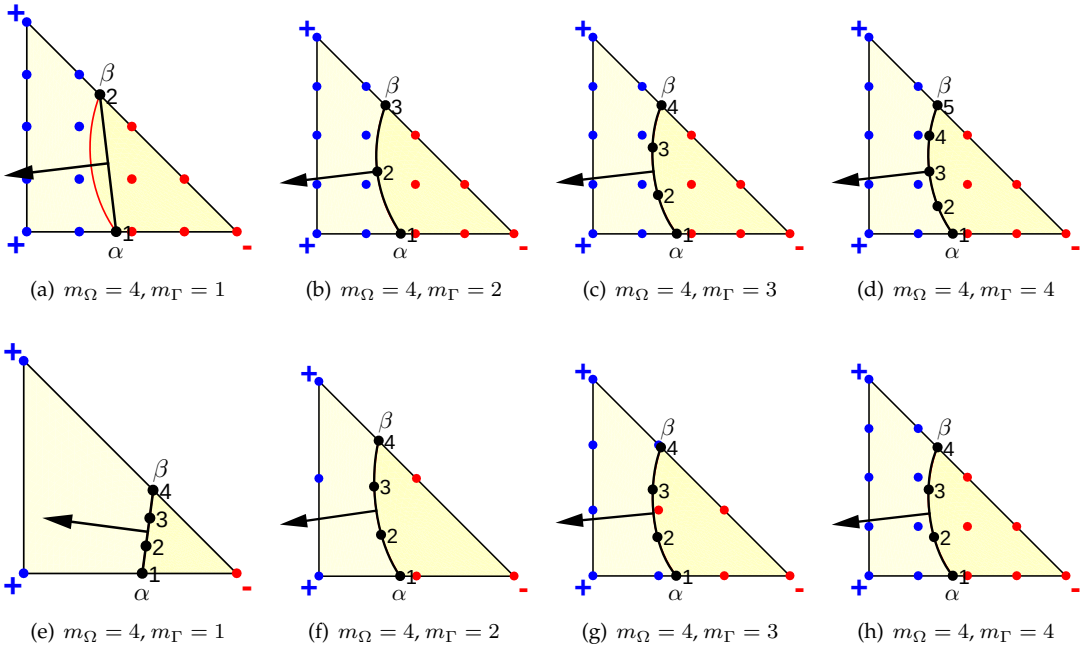
Variant (v) uses the same outer nodes as the two variants before, but the inner nodes are obtained based on a transfinite mapping as introduced in [157]. This mapping defines a surface (a triangle or a quadrilateral) or a volume based on parametrized element boundaries and faces. The details of the mapping for two- and three dimensions and various element types are given in Section 5.

4.11 Examples for reconstructed zero level sets

Some examples of the introduced methodology are given next. Reconstructed zero level sets in two and three dimensions for different type of reference elements and element orders are shown.

4.11.1 zero level sets in two-dimensional reference elements

Fig. 45(a)-(d) shows a reference triangle element of order $m_\Omega = 4$. The zero-isoline is approximated with curved one-dimensional elements of increasing orders m_Γ. Variant (iv) of the 2D reconstruction has been used here, see Section 4.10.4. The other variants achieve *visually* very similar 1D-elements, however, with considerable different properties in terms of accuracy and convergence in the numerical studies. Fig. 45(e)-(h) shows approximated isolines with a fixed order $m_\Gamma = 3$, in triangular reference elements of increasing orders m_Ω. Note that the normal vector implied by the order of the element nodes must point from Ω^- to Ω^+.



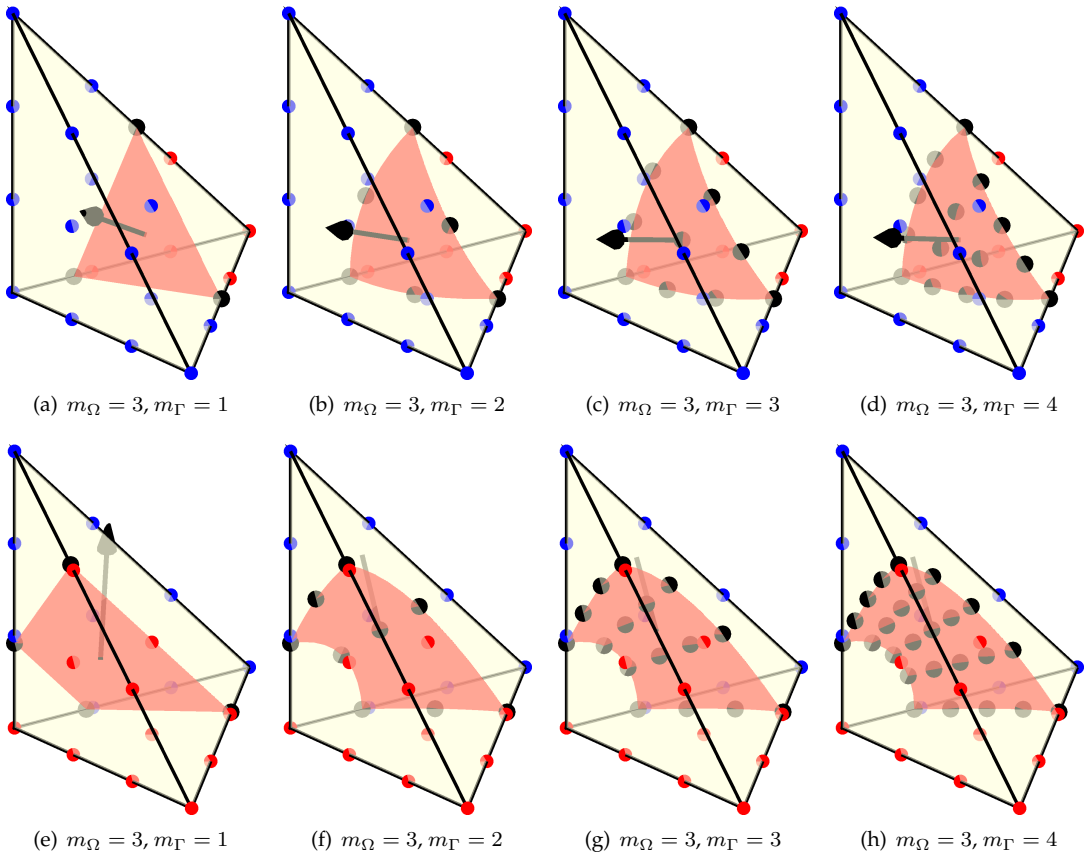
(a) $m_\Omega = 4, m_\Gamma = 1$ (b) $m_\Omega = 4, m_\Gamma = 2$ (c) $m_\Omega = 4, m_\Gamma = 3$ (d) $m_\Omega = 4, m_\Gamma = 4$

(e) $m_\Omega = 4, m_\Gamma = 1$ (f) $m_\Omega = 4, m_\Gamma = 2$ (g) $m_\Omega = 4, m_\Gamma = 3$ (h) $m_\Omega = 4, m_\Gamma = 4$

Figure 45: Example for a reconstructed zero-isoline with 1D-elements of different orders m_Γ in a triangular reference element of order $m_\Omega = 4$ in (a)-(d). In (e)-(h) examples for a reconstructed zero-isoline with a 1D-element of order $m_\Gamma = 3$ in triangular reference elements of different orders m_Ω are shown.

4.11.2 zero level sets in three-dimensional reference elements

The reconstructions of zero isosurfaces in a tetrahedral reference element are given in Fig. 46(a)-(h). In Fig. 46(a)-(d), the order of the tetrahedral element is kept constant, $m_\Omega = 3$, and the order of the reconstructed interface element m_Γ is varied. Fig. 46(a)-(d) shows a reconstructed zero-isosurface in terms of a higher-order triangle interface element which is relevant for topology 1. An example for topology 2, where the zero-isosurface is approximated by a higher-order quadrilateral interface element is seen in Fig. 46(e)-(h) .



(a) $m_\Omega = 3, m_\Gamma = 1$ (b) $m_\Omega = 3, m_\Gamma = 2$ (c) $m_\Omega = 3, m_\Gamma = 3$ (d) $m_\Omega = 3, m_\Gamma = 4$

(e) $m_\Omega = 3, m_\Gamma = 1$ (f) $m_\Omega = 3, m_\Gamma = 2$ (g) $m_\Omega = 3, m_\Gamma = 3$ (h) $m_\Omega = 3, m_\Gamma = 4$

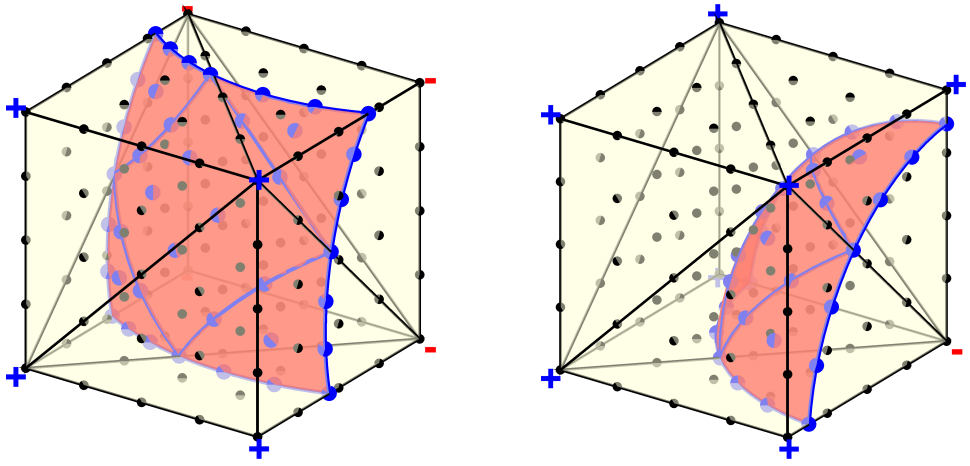
Figure 46: Example for a reconstructed zero-isosurface for topology 1 with interface elements of different orders m_Γ in a tetrahedral reference element of order $m_\Omega = 3$ in (a)-(d). In (e)-(f) examples for a reconstructed zero-isosurface for topology 2 with interface elements of different orders m_Γ in a tetrahedral reference element of order $m_\Omega = 3$ are shown.

Reconstructions of zero-isosurfaces in a hexahedral reference element are shown in Fig. 47. Each hexahedron is decomposed into 5 tetrahedra and the zero-isosurface is, in a first step, approximated by the algorithms for tetrahedra mentioned above. The nodes of the resulting interface elements serve as accurate starting guesses for the root search of the zero-isosurface of the *hexahedron*.

In Fig. 48, the issue of the implied normal vectors for the reconstructed interface elements is elaborated. Note that the zero-isosurface does not change when a level set function is multiplied by (-1), however, the normal vector $\boldsymbol{n}$ points in the opposite direction. This is realized in terms of a changed node numbering of the reconstructed element, compare Figs. 48(a) and (b) on the one hand and (c) and (d) on the other.

4.11.3 zero level sets by recursive refinement

The above examples of zero level sets in the reference elements were based on valid level set data in the sense that the element edges define the topology of the cut situation in the element, see also Section 4.3. Then, typically no (recursive) refinement is needed. For the approximation of a rather complicated zero level set in a mesh, the refinement of reference elements is a common



(a) $m_\Omega = 3, m_\Gamma = 2$ (b) $m_\Omega = 3, m_\Gamma = 2$

Figure 47: Examples for different reconstructed zero isosurfaces in a hexahedral reference element of order $m_\Omega = 4$ with interface elements of order $m_\Gamma = 3$. Also the decomposition into the 5 tetrahedra is shown.

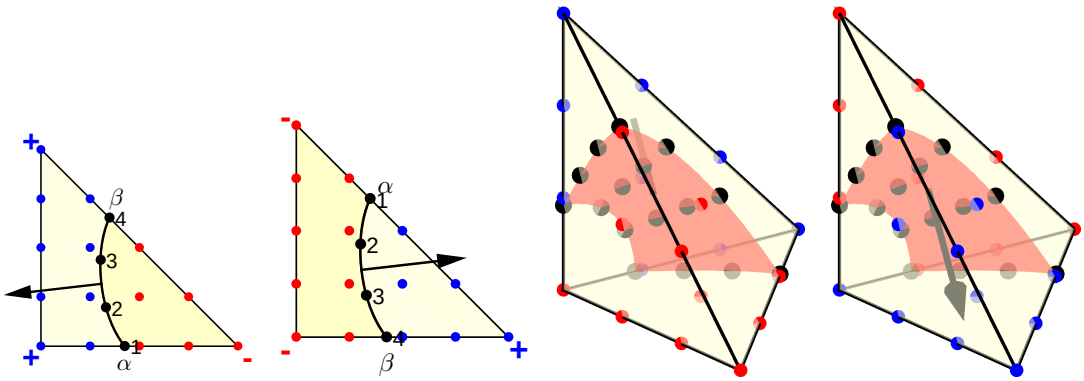
issue. However, it is emphasized that the recursive refinement typically applies only for a few elements in a mesh. Examples of the recursive treatment, in cases where the reconstruction in the original reference element was not successful, are shown in Fig. 49.
It is seen that highly complex zero level sets are reconstructed successfully. There is no need that the resulting zero level set is connected. It is interesting to note that Runge's phenomenon may be observed for zero level sets with significant curvature, see Fig. 49(c). Therefore, it is still recommended to avoid level set data in an element that leads to such complex zero level sets. One should use suitable element sizes in the mesh and if possible adjust the size of the background element if the zero level set is complex as shown here. It is later seen in the numerical results that Runge's phenomenon does not hinder optimal convergence rates of the reconstruction. That is, the error which is potentially induced by this phenomenon is diminished upon mesh refinement and does not dominate the results.

4.11.4 zero level sets in meshes

Recall that the reconstruction of the zero level set in the reference elements is only a first step. The task is to approximate the zero level set of $\phi(\boldsymbol{x})$ in the global domain Ω. This is achieved by an isoparametric map from points on the reconstructed interface element of order m_Γ (in coordinates $\boldsymbol{r}$) inside a reference background element of order m_Ω to the physical element (in coordinates $\boldsymbol{x}$). Some examples are shown in Fig. 50 for a two-dimensional domain and in Fig. 51 for a three-dimensional domain. The level set function

$$\phi(\boldsymbol{x}) = \sqrt{x^2 + y^2 + z^2} - r \tag{4.7}$$

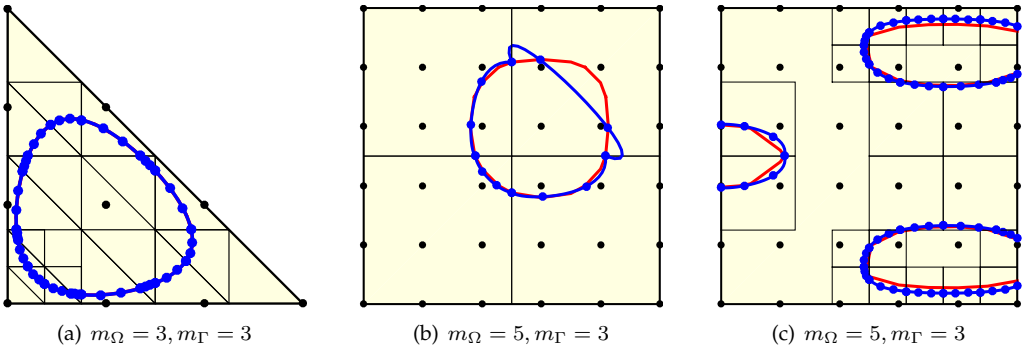
is used in these examples. It is evaluated at the nodes in Ω to obtain $\phi_i = \phi(\boldsymbol{x}_i)$ and thereby defines $\phi^h(\boldsymbol{x})$. Note that there is no need to use structured background meshes as e.g. in Fig. 50. The quality of the reconstructions is investigated thoroughly in Section. 7.



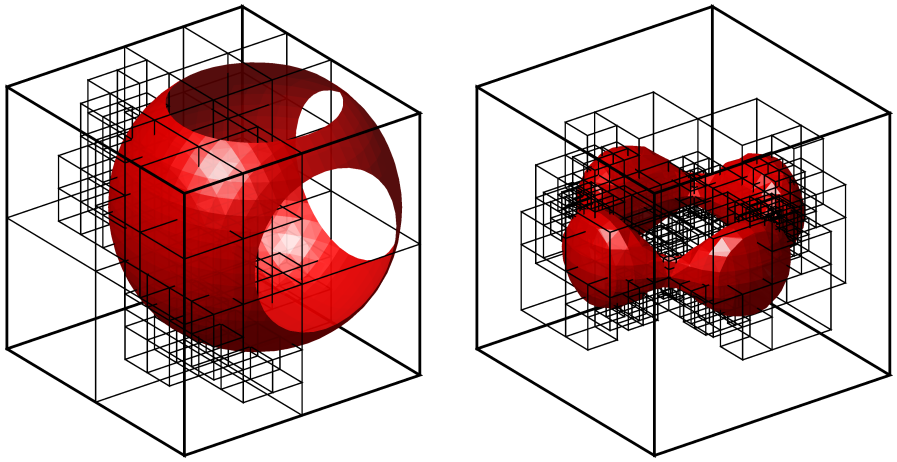
(a) ϕ^h_{tri} (b) $-\phi^h_{\text{tri}}$ (c) ϕ^h_{tetr} (d) $-\phi^h_{\text{tetr}}$

Figure 48: The normal vector always points from Ω^- to Ω^+. The consequence is that when the level set data is multiplied by (-1), this implies a change in the numbering of the element nodes.

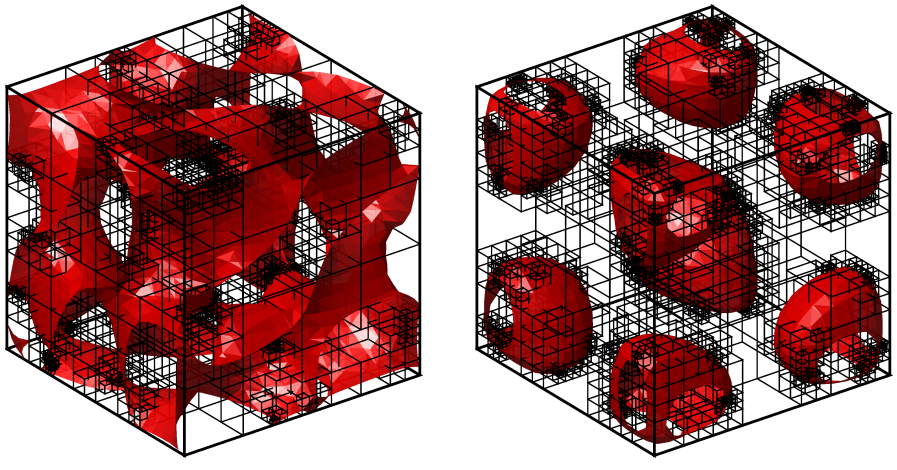
If the reconstructed mesh is used for simulation purposes, a recursive scheme is prohibited as it introduces hanging nodes. This results in a non-conforming mesh which can only be used employing additional techniques, this approach is not investigated further in this work. Instead, we use the non-local subdivision method. Thereby, a step before the subdivision is necessary, removing non-valid situations with glancing intersections, see Fig. 52. Even though this approach works well for the glancing intersections, regions with higher curvatures can not be resolved as efficient as with a recursive subdivision. To resolve local features, either a global refinement or an adaptive subdivision according to some geometric criteria is necessary.



(a) $m_\Omega = 3, m_\Gamma = 3$ (b) $m_\Omega = 5, m_\Gamma = 3$ (c) $m_\Omega = 5, m_\Gamma = 3$

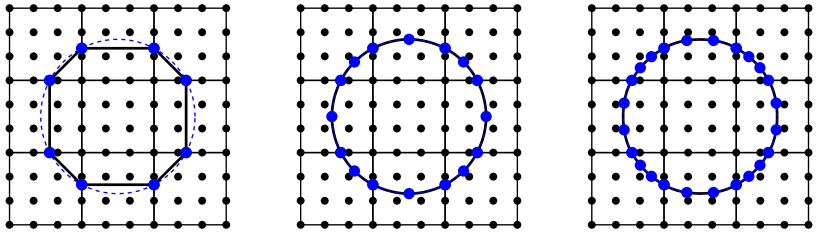


(d) $m_\Omega = 4, m_\Gamma = 2$ (e) $m_\Omega = 4, m_\Gamma = 2$



(f) $m_\Omega = 4, m_\Gamma = 2$ (g) $m_\Omega = 4, m_\Gamma = 2$

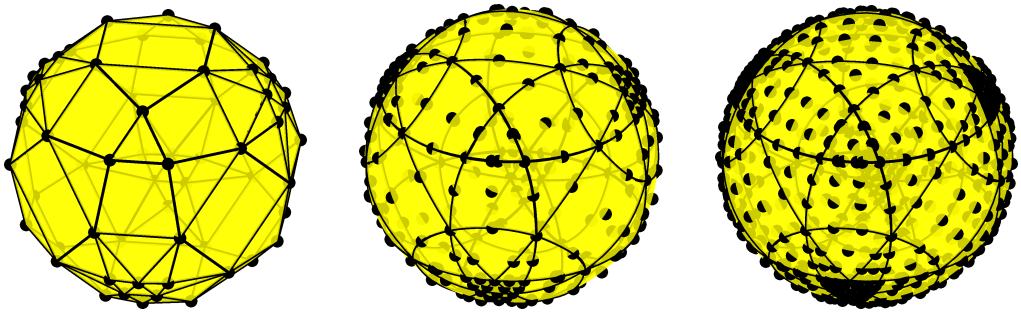
Figure 49: For complex zero level sets, a recursive refinement is needed. For significant curvature in the reference element, Runge's phenomenon may be observed as in (c) (where, the red line is the exact zero-isoline of ϕ^h and the reconstruction is blue). Connected and unconnected zero level set functions with strong curvatures for three dimensional problems are shown in (d)-(g).



(a) $m_\Omega = 3, m_\Gamma = 1$ (b) $m_\Omega = 3, m_\Gamma = 2$ (c) $m_\Omega = 3, m_\Gamma = 3$

(d) $m_\Omega = 1, m_\Gamma = 3$ (e) $m_\Omega = 2, m_\Gamma = 3$ (f) $m_\Omega = 3, m_\Gamma = 3$

Figure 50: The isoline is reconstructed on meshes of different orders m_Ω by curved 1D-elements of different orders m_Γ.



(a) $m_\Gamma = 1$ (b) $m_\Gamma = 2$ (c) $m_\Gamma = 3$

Figure 51: The zero isosurface is reconstructed on a tetrahedral mesh with order $m_\Omega = 3$ (not shown) by interface elements of different orders m_Γ. (a) $m_\Gamma = 1$; (b) $m_\Gamma = 2$ and $m_\Gamma = 3$;

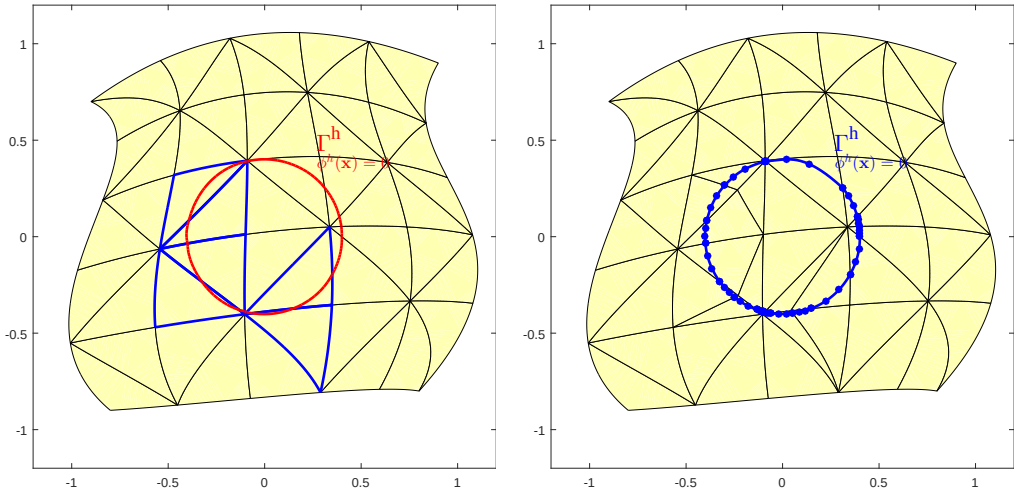


Figure 52: Triangular background meshes with $m_\Omega = 3$ for a circular zero level set ϕ^h. The task is to approximate the level set function using the introduced subdivision technique. However, some of the elements feature an invalid topological situation due to glancing intersections, they are highlighted in (a). After a non-local subdivision, which is a refinement with minimal disturbance of elements in direct proximity, all elements exhibit valid level set data .

5 Higher-order integration techniques in/on implicitly defined domains

Up to this point we assumed that all emerging integrals that appear in the computation of the stiffness matrix and the load vector are evaluated analytically. In fact, the individual terms are products of shape function derivatives and shape functions, which are both by default polynomials, see Section 2. Even so, for general integrands over complex shaped domains analytical integration is pointless and most of the time even impossible. Instead of analytical integration the standard procedure for approximate methods is to use numerical integration techniques, namely, the integral expression is replaced by a finite sum of integrand evaluations in certain points. Especially the multidimensional numerical integration of smooth functions over simple domains has been studied intensively, see [163, 49, 109]. A vast number of procedures exist in the literature, among them the class of rules based on integrand interpolation. This means that the integrand is replaced by a polynomial and instead of approximately integrating the initially given function, the substitute function is then integrated exactly. In that sense the integration and the interpolation error commute.

Methods in this category are, amongst others, the *Newton-Cotes formulas.* They are characterized by an equidistant distribution of integration points. The location of the integration points, specifically, the points where the integrand is evaluated, is fixed and the weights are optimized in order to integrate optimally the highest possible polynomial. Special cases of Newton-Cotes formulas are the midpoint, trapezoidal and Simpsons rule. It can be shown that the Newton–Cotes formulas have an order of accuracy of $p = n$ for n odd and $p = n + 1$ for n even. The extra accuracy for n even, is, as in Simpsons rule, due to symmetry. However, the indisputable drawback of the method is that for high orders ($p > 8$) negative weights may appear, which results finally in a less robust scheme. Another popular class is the *Clenshaw-Curtis quadrature*; this quadrature rule uses a coordinate transformation and evaluates the integrand in the roots of the Chebyshev polynomials. The integration points for this method cluster near the endpoints of the interval and show an increased stability for higher polynomial orders. Furthermore, it can be shown that the integration coefficients are always positive, see [172, 173].

The most important and widely applied class of numerical integration rules is, especially due to its efficiency, the *Gauß-Legendre quadrature*. It possesses the very attractive property to integrate polynomial functions *exactly*, up to a degree of

$$p = 2n - 1, \tag{5.1}$$

by using n evaluations of the integrand. Consequently, the integration scheme is referred as n-point Gaussian quadrature and is given in the one-dimensional case as

$$I = \int_{-1}^{1} f(r)\,\mathrm{d}r \approx \sum_{i=1}^{n} w_{n,i} f(r_{n,i})$$

with $I : \mathbb{R} \to \mathbb{R}$ as the value of the integral, $f(r)$ as a continuous function, $r_{n,i} \in (-1, 1), i = 1, 2 \dots n$, are the integration points in the specified interval, and $w_{n,i} \in \mathbb{R}$ are the integration

weights. Similar to the Clenshaw-Curtis quadrature, the integration weights for the Gauß-Legendre rule are strictly positive and at least in 1D have to satisfy

$$\sum_{i=1}^{n} w_{n,i} = 2,$$

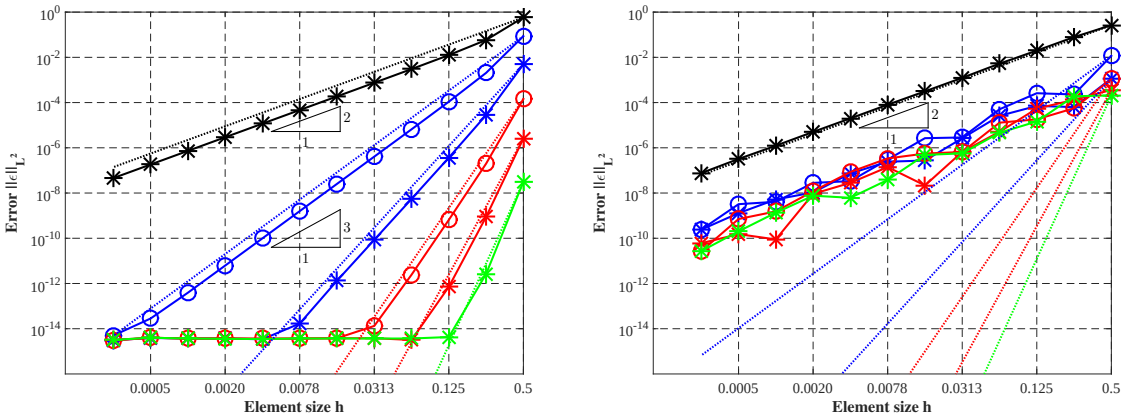
which is the length of the integration interval. The position of the integration points and the weights are optimized in order to maximize the exactly integrated polynomial degree p. The location of the sampling points, $r_{n.i}$, is given by the roots of the Legendre polynomials. Alternatively, in case of an ill-conditioning, the roots can be obtained as the solution of an Eigenvalue-problem, see [51]. The integration weights $w_{n,i}$, can be determined either solving a linear system of equations or even in an analytical form by integrating Lagrange polynomials. Observe that the integration domain is identical with the interval of the reference element; in the event that the integrand is defined on any other interval, a coordinate transformation, that is, a mapping for the location of the points from the reference to the real integration domain is necessary, as explained in Section 2.4. If the integration domain is multidimensional, e.g. a quadrilateral or hexahedral domain, the integration points are given as a tensor product of the one-dimensional abscissas. In case of triangular or tetrahedral elements, either the quadrilateral or the hexahedral elements are collapsed and transformed to the desired element. However, the results are not optimal in this case as the invariance requirement for general integrands is not fulfilled. There exist a number of custom integration rules for simplexes, see [46, 90].

Remark 1 One serious drawback of this class of integration formulas is, that the sampling points of two integration rules with different orders have in general no common abscissas. Thus, when p-adaptive schemes are applied and the polynomial order is increased, all integration points have to be exchanged. In such cases, the problem can be overcome by using a special class of integration methods, devised by Kronrod. The abscissas of the low-order integration are taken and a $(2n + 1)$-formula is fabricated. In doing so, the new nested abscissas provide the highest possible accuracy.

As mentioned before, Gaussian quadrature belongs to the class of interpolatory integration formulas. This means that for non-polynomial functions f, not the function itself but rather the interpolant $\Pi f =$ is integrated. Consequently, an integration error arises, which corresponds directly with the interpolation error of the polynomial implied by the integration order. Given a smooth function $f(x) = \sin(x); f \in C^{\infty}(\Omega)$, for simplicity reasons the function for this example is defined on the reference domain $x \in (-1, 1)$. As an analytical solution is available, a convergence diagram for the integration error is shown in Fig. 53(a). Given the results, it is obvious that the error decreases in an optimal sense, that is, the characteristical fan-shaped diagram is obtained.

If, on the contrary, the integrand is *non*-smooth, e.g. two polynomials of order $p = 10$ and an enforced global continuity, that is $f \in C^0(\Omega)$, the integration error degrades the convergence rates to a constant m, see Fig. 53(a). The constant depends on the regularity of the function f.

Remark 2 Using an interpolatory integration scheme to integrate the entries in the system matrix and in the load vector is indeed an advantage. Remember that the entries are composed as products of shape functions or shape function derivatives, which are in fact polynomial



(a) (b)

Figure 53: Convergence study for Gauß-Legendre integration for a C^∞-smooth function in (a) and for a C^0-continuous function in (b), in an error norm which corresponds to the L^2-norm, respectively.

functions. Even so, notice that this statement is valid in the reference elements only as long as the mapping of the elements from reference to real domain is affine. Once the elements are mapped to the real domain, with possibly curved boundaries, the mapping becomes non-polynomial and cannot be integrated exactly, independent of the chosen integration order.

Remark 3 Problems with seriously degraded convergence rates for numerical integration do not only occur in case of non-smooth solutions, but also in case of high gradients as in spike-shaped functions. Other difficult functions include strongly oscillatory integrands (which leads to an ill-conditioning) as well as weakly singular or even strongly singular functions as frequently arising in boundary element methods. We note that most of the mentioned problems need some kind of special treatment; either classical adaptive refinement or some tailored approaches.

The fact that inaccurate integration of a local non-smooth function degrades the global convergence properties has important consequences for the numerical schemes. Two different types of approximations were introduced up to his point; one was the choice of approximate (polynomial) spaces to describe the solution, that is $u^h \in \mathcal{S}^h$ instead of $u \in \mathcal{S}$; the second point is that within the weak form not the approximated function u^h is integrated but some polynomial approximation of it. Hence, if a non-smooth boundary value problem is approximated even utilizing a scheme that includes the proper approximation spaces which account for the discontinuities as e.g. the XFEM, FCM or CutFEM, if the integration is not executed carefully, the mentioned higher-order approximation schemes would yield rather unsatisfactory accurate results.

5.1 State of the art for discontinuous integrands

After the early stages of focusing on the methodological parts of unfitted finite element methods, nowadays the focus is turned to the numerical integration for discontinuous integrands, as

arising in the weak form of the presented methods in Section 3. This can be easily shown by the increasing number of publications over the last few years focusing only on integration schemes, for an overview see [169, 73] and the references therein. To overcome the delicate problem of computing discontinuous integrands, over the time various approaches have been developed. The methods can roughly be categorized into two major approaches:

5.1.1 Methods that use subdivision techniques

The idea is to include the discontinuity by element subdivision along the interface. The domain is divided into integration cells and each continuous portion of the function is separately integrated using standard Gauß-Legendre quadrature. The partitioning is carried out only for elements that are cut by the zero level set and is done purely for integration purposes, thus, no further degrees of freedom are introduced. The subdivision type can be very diverse, e.g. the earliest approach in [127, 165, 18] was to subdivide the element into linear sub-cells. Of course, the resulting scheme has rather modest accuracy and is not suited for higher-order approaches and curved interfaces. Other alternatives include a uniformly refined grid in cut elements or integration rules higher then actually required by the ansatz order p, see [162, 142]. Another technique is an adaptive space-tree decomposition, namely, cut elements are recursively subdivided into low-order sub-cells. In two dimensions the approach is then referred to as *quad-tree*, in three dimensions as *octree decomposition*, see e.g. [54, 1]. It is noted that even at the finest level the resolution of the zero level set is still constant, that is, the level set is described by rectangular sub-cells. As a consequence, a large number of refinements and therefore a very high number of integration points may be necessary to reach a user-defined abort criterion. Newly developed methods as in [34] use curved sub-cells that conform with the interface. Another recently introduced approach that produces sub-cells conforming with the zero level set is the so-called *smart-octree* technique. It combines quad tree/octree refinements with local reparametrization and node moving, see [110]. Most of the mentioned approaches need an explicit representation of the interface, which is not readily obtained, especially for curved interfaces in three dimensions.

5.1.2 Methods that use modified integration rules.

These approaches circumvent partitioning of the cut elements and use diverse kinds of regularizations. A large variety of methods exist, with no common basis except that element subdivision is avoided. For example, in [175] an approach is employed where non-polynomial and non-smooth functions are replaced by *equivalent polynomials*. In doing so, the polynomial is constructed such that the integral of the non-smooth function and the integral of the new equivalent polynomial coincide *exactly*. These polynomials can be integrated readily using regular Gaussian quadrature schemes. The approach has a rather limited generality; it works only for straight interfaces and only for triangular and tetrahedral elements, but not for quadrilaterals and hexahedra. We mention a closely related approach, the *variable weight quadrature*, as introduced in [97], which also suffers from the same drawbacks as the equivalent polynomial method. Another, very recently introduced approach in [128, 164, 129, 105], the so-called moment-fitting, is based on an optimization procedure. The general idea is to use standard Gauß-points and to re-compute their weights by moment-fitting equations, that is, solving a linear system of equations; a drawback

of the approach is that a suitable divergence-free function basis must be provided. In [113], a parametric mapping of the underlying background mesh is constructed in a way that after the mapping a planar discretization conforms to curved interfaces, even in case of sharp features. However, the procedure has the serious drawback that the level set function *must* be given in an analytical form. It is emphasized that none of the so far introduced methods is able to deal with multiple level sets and eventually occurring geometrically complicated intersections. At least no optimal results are known to the author. Compared with the element decomposition techniques, it seems that the modified integration approaches are often less intuitive and offer not the same generality. Another approaches in this category are the *Schwarz-Christoffel conformal mapping* introduced in [130, 131] and *Lasserre's method* as introduced in [112]. Using the divergence theorem volume integrals are reduced to the domain boundary. Recently the method was extended to include non-convex polygons and curved regions, see [36].

5.2 Higher-order accurate integration of discontinuous functions using a conformal decomposition technique

At this point we describe an alternative, higher-order accurate, integration method as introduced in [73], which can be classified in the realm of methods using subdivision techniques. We assume that a general higher-order cut background element $\Omega^h_{\boldsymbol{r}}$ is given and the zero level set is successfully approximated by the reconstructed interface element $\Gamma^h_{\boldsymbol{r}}$, see Section 4. In this section we are interested only in integration points, nevertheless, the aim is to find a map for *any point* from the higher-order reference element into the integration sub-cells in physical space, which is not trivial as will be seen below. It is noted that the resulting decomposition features triangular and quadrilateral elements in 2D as well as tetrahedral and prism elements in 3D.

There are several mappings and transformations involved and Fig. 54 shows the complete process in one diagram.
Emanating from the physical element $\Omega^h_{\boldsymbol{x}}$, in **step 1** the level set values are transferred to the parent domain $\Omega^h_{\boldsymbol{r}}$, where the reconstruction of the zero level set is realized, this procedure was described in detail in Section 4. In **step 2** the subdivision of $\Omega^h_{\boldsymbol{r}}$ takes place. The element is subdivided into a few predefined topologies depending on the cut situation,. The only missing link for a working computational mesh is a node distribution within the element patch that preserves optimal accuracy for higher-order elements. This is done by means of a customized mapping procedure, sketched in **step 3** of Fig. 54, which is why the approach for triangular and quadrilateral elements differentiates significantly. Once the nodal coordinates are determined, a conforming higher-order element patch is obtained. At last, in **step 4** the data is projected back to the physical domain via an isoparametric mapping.

5.3 Coordinate mappings

The task is to construct a map for *any point* from a higher-order reference element to the previously defined low-order sub-cells with potentially one higher-order side. The mapping is constructed in a way that the following properties are fulfilled: (i) the higher-order element resulting from the coordinate transformation yields optimal interpolation properties and (ii) one element edge or face aligns with the (curved) higher-order interface element $\Gamma^h_{\boldsymbol{r}}$.

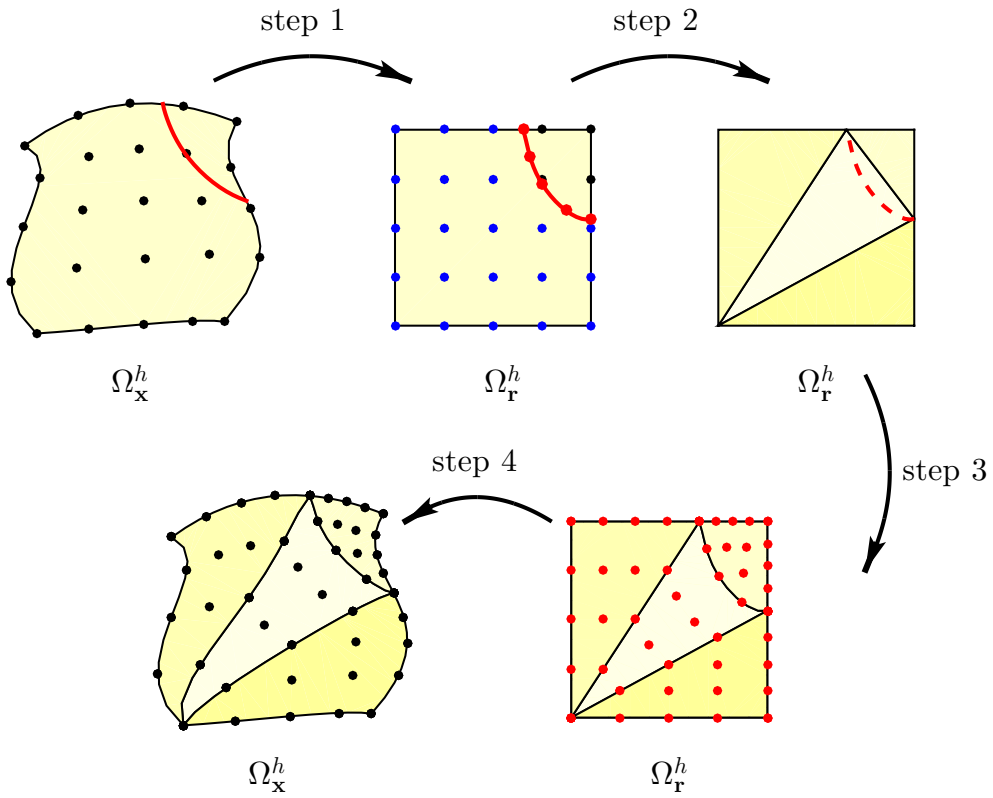


Figure 54: Overview of the complete decomposition and mapping strategy. Notice that the element in the first and the last step is defined in the physical space and all steps in-between are defined in the reference element.

As seen from the two-dimensional example in Fig. 55, several element types and corresponding coordinate systems are involved: Central to the whole procedure are the low-order sub-cells (with one curved higher-order side) described in the coordinate system $\boldsymbol{r}$, as defined above. They define the outer boundary of the higher-order elements resulting from the automatic conformal decomposition. A key point is that the mapping for inner points in the sub-cells is not unique. This means that inner the nodes of the resulting higher-order elements may be placed differently, depending on the applied map. This fact has important implications on the convergence properties of the whole mesh. The higher-order interface element representing the discretized zero level set ϕ^h is initially defined in the reference coordinate system u (and accordingly $\boldsymbol{u}$ in 3D) and is mapped to the reference element $\Omega_{\boldsymbol{r}}$ by $\boldsymbol{r}(u) : \mathbb{R} \rightarrow \mathbb{R}^2$ and $\boldsymbol{r}(\boldsymbol{u}) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ for two-dimensional and three-dimensional problems, respectively. Finally, there are higher-order reference sub-elements defined in the coordinate system $\boldsymbol{a}$. These sub-elements are used to define the initial location of the nodes and the mapping to $\Omega_{\boldsymbol{r}}$ is, depending on the problem dimension, denoted as $\boldsymbol{r}(\boldsymbol{a}) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ and $\boldsymbol{r}(\boldsymbol{a}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$.

In the following three different categories of mappings will be introduced. The concept of the "proper" choice of mapping is of paramount importance for the interpolation and consequently the approximation properties.

It is useful to introduce the following definitions related to nodal sets and corresponding shape functions of the higher-order sub-elements in the reference coordinate system $\boldsymbol{a}$. All de-

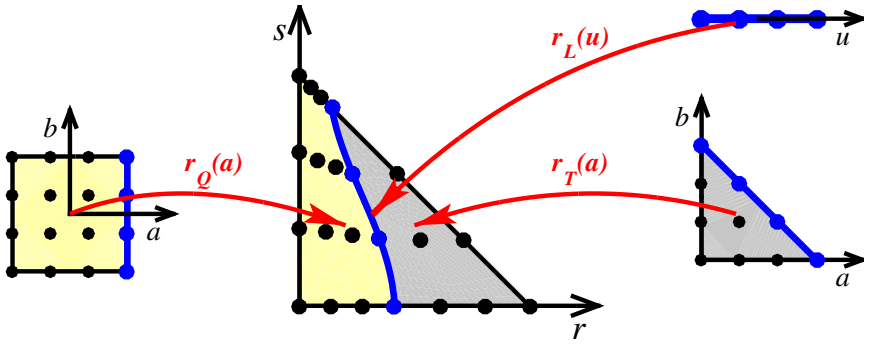


Figure 55: A higher-order triangular background element is cut by the reconstructed interface element $\Gamma_{\boldsymbol{r}}^h$ and subdivided into low-order sub-cells defining topology $\mathbf{T}_{12}$. The mapping from the reconstructed element to the coordinates $\boldsymbol{r}$ is given by $\boldsymbol{r}_L(u)$. A set of higher-order reference elements in the coordinate system $\boldsymbol{a}$ is introduced. The coordinate transformations $\boldsymbol{r}_T(\boldsymbol{a})$ and $\boldsymbol{r}_Q(\boldsymbol{a})$ map the higher-order reference sub-elements to the coordinates $\boldsymbol{r}$ such that one special side (in our case always edge number two) matches exactly $\Gamma_{\boldsymbol{r}}^h$.

scriptions given here are for elements in two dimensions but the principal approach in three dimensions is the same. The set of the corner nodes for each element is defined by I_{lin}^{Ω}. These are three nodes for the triangular and four for quadrilateral elements. From these nodes, either one node (see Fig. 31(a) topology $\mathbf{Q}_{14}$, elements $\mathbf{T}_2$ and $\mathbf{T}_4$) or two nodes (see Fig. 31(a) for topology $\mathbf{Q}_{14}$, elements $\mathbf{T}_1$ and $\mathbf{T}_3$) belonging to $I_{\text{side}}^{\Omega} \subset I_{\text{lin}}^{\Omega}$ are on the linear reconstruction when mapped to $\boldsymbol{r}$. For simplicity, we choose a node numbering in the sub-elements such that consistently edge number two of triangular or quadrilateral elements in 2D or face number four for tetrahedral elements in 3D coincide exactly with the interface element. The set of nodes on edge number two is called I_{ho}^{Γ}, and the corner nodes are $I_{\text{lin}}^{\Gamma} \subset I_{\text{ho}}^{\Gamma}$. Related to these nodes are the nodal coordinates $\boldsymbol{a}_i$ before the mapping and $\boldsymbol{r}_i$ after the mapping to the cut reference background element. Observe that the nodes in I_{side}^{Ω} and I_{lin}^{Γ} must superpose in coordinate systems $\boldsymbol{a}$ and $\boldsymbol{r}$. We associate the following set of shape functions with the above-mentioned nodes: $N_{\text{lin},i}^{\Omega}(\boldsymbol{a})$ for $i \in I_{\text{lin}}^{\Omega}$ are the low-order shape functions of the reference sub-element. $N_{\text{ho},i}^{\Gamma}(u)$ for $i \in I_{\text{ho}}^{\Gamma}$ are the higher-order shape functions on the special side of the sub-element. $N_{\text{lin},i}^{\Gamma}(u)$ for $i \in I_{\text{lin}}^{\Gamma}$ are low-order shape functions on the special side which coincide with the trace of the low-order shape functions $N_{\text{lin},i}^{\Omega}(\boldsymbol{a})$ for $i \in I_{\text{side}}^{\Omega}$ on the second edge. The only change for three dimensional problems is that the set I_{ho}^{Γ} now belongs to face number four and the local parametrization of the interface is vector-valued, that is $N_{\text{ho},i}^{\Gamma}(\boldsymbol{u})$.

The ambition is to construct a map for all points from $\boldsymbol{a}$ to $\boldsymbol{r}$ based on the known parametrization of the element boundary. It is essential, that $\Gamma_{\boldsymbol{r}}^h$ is included in some of the elements of the element patch as a curved boundary. The mapping transforms the low-order sub-cells ultimately into higher-order elements with one curved edge. At the end we adapt a transfinite interpolation technique as described e.g. in [85, 86, 169, 157]. The scheme yields the blending function method for elements with a tensor-product like structure (quadrilaterals, hexahedra) [85, 86] and a slightly modified technique for simplex shaped elements (triangles, tetrahedra). We start with the mapping for the elements having one node in I_{side}^{Ω} (or alternatively the assumption that $\Gamma_{\boldsymbol{r}}^h$

is straight). Thereby, the map from $\boldsymbol{a}$ to $\boldsymbol{r}$ is defined completely by the affine mapping $\boldsymbol{r}_m^v(\boldsymbol{a}_n)$ specified only by the element vertices. The map reduces in two and three dimensions to the affine isoparametric mapping (bi-linear and tri-linear elements in case of quadrilaterals and tetrahedra, respectively) as

$$\boldsymbol{r}_m(\boldsymbol{a}_n) = \boldsymbol{r}_m^v(\boldsymbol{a}_n) := \sum_{i \in I_{\text{lin}}^\Omega} N_{\text{lin},i}^\Omega(\boldsymbol{a}_n) \cdot \boldsymbol{r}_{\text{lin},i}^\Omega, \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^{\text{dim}}. \tag{5.2}$$

The coordinates $\boldsymbol{a}_n$ and $\boldsymbol{r}_m$ are the nodal coordinates and $\boldsymbol{r}_{\text{lin},i}^\Omega$ are the corner nodes of a sub-element in the cut reference background element in the corresponding coordinate systems. To include the more elaborate case of a *curved* higher-order interface element, an additional term to the standard isoparametric mapping must be included. The two dimensional map is defined as

$$\boldsymbol{r}_m(\boldsymbol{a}_n) = \boldsymbol{r}_m^v(\boldsymbol{a}_n) + \boldsymbol{r}_m^e(\boldsymbol{a}_n), \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^2. \tag{5.3}$$

The term $\boldsymbol{r}_m^e$ takes into account curved edges on an elsewhere standard reference element. In three dimensions not only the edges but also the faces can be curved, therefore, an additional term is needed as

$$\boldsymbol{r}_m(\boldsymbol{a}_n) = \boldsymbol{r}_m^v(\boldsymbol{a}_n) + \boldsymbol{r}_m^e(\boldsymbol{a}_n) + \boldsymbol{r}_m^f(\boldsymbol{a}_n). \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^3. \tag{5.4}$$

Here $\boldsymbol{r}_m^f(\boldsymbol{a}_n)$ is defined as the contribution of curved faces that *vanishes* on the element edges (sometimes referred to as bubble function). Eqs. (5.2), (5.3) and (5.4) are valid for *any* point inside $\Omega_{\boldsymbol{a}}$. Note, that Eq. (5.3) *must* include Eq. (5.2) as a special case of an element with only straight edges. The additional vector functions $\boldsymbol{r}_m^e(\boldsymbol{a}_n)$ and $\boldsymbol{r}_m^f(\boldsymbol{a}_n)$ must be defined such that they vanish for elements with only straight edges. It is interesting to note that in case that Eqs. (5.3, 5.4) are rewritten as

$$\boldsymbol{r}_m(\boldsymbol{a}_n) = \boldsymbol{r}_m^v(\boldsymbol{a}_n) + \boldsymbol{M}(\boldsymbol{a}_n), \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^2. \tag{5.5}$$

with $\boldsymbol{M}(\boldsymbol{a}) := \boldsymbol{r}_m^e(\boldsymbol{a}_n)$ and $\boldsymbol{M}(\boldsymbol{a}) := \boldsymbol{r}_m^e(\boldsymbol{a}_n) + \boldsymbol{r}_m^f(\boldsymbol{a}_n)$ in 2D and 3D, respectively, the definition of the geometric mapping formally matches *enriched* approximations such as those used in the XFEM [16, 127, 72, 10, 11].

For the definition of the edge and face contributions $\boldsymbol{M}(\boldsymbol{a})$, we impose three important constraints:

1. The mapping must be local, namely, the mapping $\boldsymbol{M}(\boldsymbol{a})$ is constructed using only points in the considered element.
2. The corner nodes of the higher-order interface element I_{lin}^Γ match the corner nodes of the special side of the low-order element I_{side}^Ω in the coordinate system $\boldsymbol{r}$.
3. Those edges of the low-order elements that are not on the special side shall remain straight after the mapping.

This leads to the requirement that

$$\boldsymbol{M}(\boldsymbol{a}_i) = \boldsymbol{0} \quad \forall i \in I_{\text{lin}}^\Omega, \tag{5.6}$$

as the vertex contribution is defined in $\boldsymbol{r}_m^v\left(\boldsymbol{a}_n\right)$. We find that for the sake of clarity it is useful to split $\boldsymbol{M}\left(\boldsymbol{a}\right)$ into the product

$$\boldsymbol{M}\left(\boldsymbol{a}\right) = \psi\left(\boldsymbol{a}\right) \cdot \boldsymbol{f}\left(u\left(\boldsymbol{a}\right)\right)$$

with the definition of $\boldsymbol{f}\left(u(\boldsymbol{a})\right)$ as

$$\boldsymbol{f}\left(u(\boldsymbol{a})\right) = \sum_{i \in I_{\mathrm{ho}}^{\Gamma}} N_{\mathrm{ho},i}^{\Gamma}(u(\boldsymbol{a})) \cdot \boldsymbol{r}_i^{\Gamma} - \sum_{i \in I_{\mathrm{lin}}^{\Gamma}} N_{\mathrm{lin},i}^{\Gamma}(u(\boldsymbol{a})) \cdot \boldsymbol{r}_i^{\Gamma} \tag{5.7}$$

The coordinates $\boldsymbol{r}_i^{\Gamma}$ are the element nodes of the reconstructed interface elements in the coordinate system $\boldsymbol{r}$. It is worth noting, that the function $\boldsymbol{f}\left(u(\boldsymbol{a})\right)$ corresponds geometrically to subtracting the linear connection of the nodes in $I_{\mathrm{lin}}^{\Gamma}$ from the curved higher-order side. The definition of ψ depends on the element type and, in simple words, the function describes the process how the curved edge is blended into an element with only straight edges. It is obvious that ψ is not unique and there are many conceivable alternatives to it.

As theoretically devised by [38] for 2D simplex elements and extended by [115] and [19] for higher dimensions, certain smoothness properties have to be fulfilled by the desired coordinate transformation. It can be stated that the geometrical mapping, that is defined by the shape functions in the reference domain, should be polynomial in the real space. Further explanations to other types of transfinite mappings are found in [157, 171].

There exist other ways to construct a mapping, for example, solving a Laplace equation as done in [63, 64] in context of the cut-cell method. But the equation has to be solved for every cut element, which is numerically far more demanding than to construct an analytical mapping as done in the following.

5.3.1 Mapping for triangular elements

We proceed here from **step ii** in Fig. 54, i.e. the interface element is located and meshed according to Section 4.2, and the topology of the cut element is determined as described in Section 4.3. Similar to the two former steps, the complete mapping construction process takes place in the reference element. The final goal of this section is to construct a mathematical function which is valid for almost all points within the element, with the possible exception of some isolated singular points. The function maps any point from the linear reference element to the element with one curved and parametrized edge, based on boundary information only. It is important to realize that the result of the mapping is still in reference coordinates. The presented procedures are not restricted to the particular situation of one curved edge, e.g. in case of three curved edges the chosen mapping function is applied sequentially to all edges. We make use of this when considering the mapping of tetrahedra.

The higher-order edge, for two-dimensional elements defined as edge number two, coincides after the mapping with the curved reconstructed zero isoline. The higher-order side is for our purposes parametrized by an one-dimensional straight curve with the local coordinates $u_T(r,s)$: $\{r,s\} \in [-1,1]$, see Fig. 56(a).

There is a large variety of possibilities to map an arbitrary point $\boldsymbol{a}^{\star}$ with the coordinates (a,b) onto the boundary. One method is to map the point perpendicular to edge number two, see

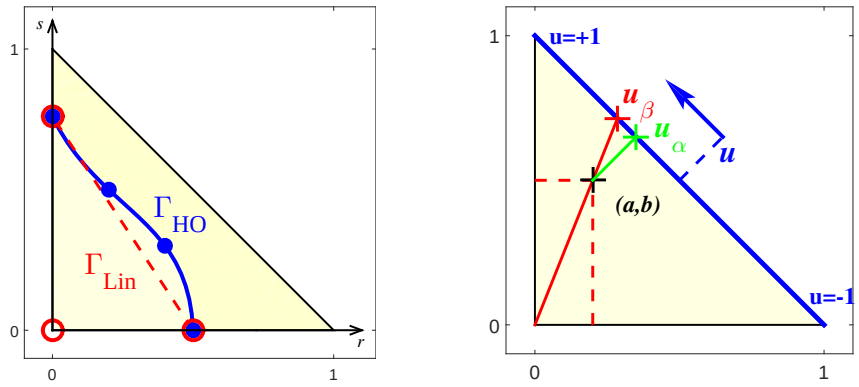


Figure 56: Cut triangular reference element with the reconstructed higher-order line element Γ_{HO} and the indicated topology $\boldsymbol{T}_{13}$ in (a). Overview of used coordinate systems on the parameterized curved edge in (b). The hypotenuse is parametrized by a 1D element with domain ranging from $-1 \leq u_T \leq 1$. The two used projections $u_{T,\alpha}$ and $u_{T,\beta}$ are shown exemplary using an arbitrary point with the coordinates (a, b).

Fig. 56(b), this is given as

$$u_{T,\alpha}(\boldsymbol{a}) = b - a \qquad u_{T,\alpha}(\boldsymbol{a}) = N^{\Omega}_{\text{lin},3}(\boldsymbol{a}) - N^{\Omega}_{\text{lin},2}(\boldsymbol{a})\,. \tag{5.8}$$

Other possibilities include a projection connecting the coordinate origin, thus node 1, with the desired point within of the element and extend the line to the point of intersection with the boundary, see Fig. 56(b). The mathematical description is then

$$u_{T,\beta}(\boldsymbol{a}) = \frac{b-a}{a+b} \qquad u_{T,\beta}(\boldsymbol{a}) = \frac{N^{\Omega}_{\text{lin},3}(\boldsymbol{a}) - N^{\Omega}_{\text{lin},2}(\boldsymbol{a})}{N^{\Omega}_{\text{lin},3}(\boldsymbol{a}) + N^{\Omega}_{\text{lin},2}(\boldsymbol{a})} \tag{5.9}$$

Notice that for the mapping u_β the origin of the coordinate system $\boldsymbol{r}$ represents a singular point, as for $a = b = 0$ the denominator becomes zero. This means that the point $(0, 0)$ cannot be mapped uniquely onto the (second) edge, however, all other points are well-defined.

Blending function mapping

The basic idea of this mapping is to naïvely extend the blending function approach, initially developed for quadrilaterals in [85, 86], to triangles, see [14]. The mapping from element interior to the element boundary is done using u_β and afterwards a linear weighting function (or ramp function, from here on denoted by $R(\boldsymbol{a})$) is used to blend the given function values on the boundary to the element interior. The weight function ψ is given by [73] as

$$\psi(\boldsymbol{a}) = a + b \qquad \psi(\boldsymbol{a}) = R_2(\boldsymbol{a}) = \sum_{i=2}^{3} N^{\Omega}_{\text{lin},i}(\boldsymbol{a})\,, \tag{5.10}$$

with $R_2(\boldsymbol{a})$ as ramp function. This means $R_2(\boldsymbol{a})$ has unity value on edge two and vanishes towards node number one, see Fig. 58(a). The boundary values are given by the function $\boldsymbol{f}\,(u(\boldsymbol{a}))$

from Eq. 5.7, describing the deviation of the curved boundary edge from the straight-sided edge. The final mapping is then defined as

$$\boldsymbol{r}_m\left(\boldsymbol{a}_n\right) = \sum_{i=2}^{3} N_{\mathrm{lin},i}^{\Omega}\left(\boldsymbol{a}_n\right) \cdot \boldsymbol{r}_{\mathrm{lin},i}^{\Omega} + \sum_{i=2}^{3} N_{\mathrm{lin},i}^{\Omega}\left(\boldsymbol{a}\right) \cdot \boldsymbol{f}(u_T(\boldsymbol{a})), \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^2.$$

which is valid for all points with exception of nodal points on $I_{\mathrm{side}}^{\Omega}$, where a division by zero arises. For these points, however, the mapping of the vertices is defined anyway by the standard isoparametric mapping and therefore ψ can be set to zero there.

Transfinite mapping

The blending function mapping is a particular case of a more general type of mappings, the so-called class of transfinite interpolations. This concept was for the first time introduced in [42], and later on extended among others in [85, 86]. The mappings were used to construct interpolatory surfaces which would coincide with some prescribed curves. For further references and a more general treatment of transfinite mappings, see [157].

This particular type of function was already used in [136, 74] for remeshing and integration purposes, respectively, and will be the preferred function for our intended purpose. We point out the good interpolation properties of the map as well as its extendability to multiple dimensions. The mapping from the element domain onto the element boundary is done utilizing the normal projection u_α. Note that the orientation of the second edge, as defined in Fig. 56(b) has to be taken into account. As a result, the weighting function $\psi(\boldsymbol{a})$ is then given in continuous and discrete form, respectively, as

$$\psi(\boldsymbol{a}) = \frac{r \cdot s}{u_T(\boldsymbol{a}) \cdot (1 - u(\boldsymbol{a}))} \qquad \psi(\boldsymbol{a}) = \frac{\prod\limits_{i \in I_{\mathrm{lin}}^{\Omega}} N_{\mathrm{lin},i}^{\Omega}\left(\boldsymbol{a}\right)}{\prod\limits_{i \in I_{\mathrm{lin}}^{\Gamma}} N_{\mathrm{lin},i}^{\Gamma}\left(u_T\left(\boldsymbol{a}\right)\right)}. \tag{5.11}$$

Recall that this mapping is valid for all points, except of the points located on the straight boundaries where a division by zero arises. For these points, however, the mapping of the vertices is defined anyway by the standard isoparametric mapping. The final mapping is given as follows

$$\boldsymbol{r}_m\left(\boldsymbol{a}_n\right) = \sum_{i \in I_{\mathrm{lin}}^{\Omega}} N_{\mathrm{lin},i}^{\Omega}\left(\boldsymbol{a}_n\right) \cdot \boldsymbol{r}_{\mathrm{lin},i}^{\Omega} + \frac{\prod\limits_{i=2}^{3} N_{\mathrm{lin},i}^{\Omega}\left(\boldsymbol{a}\right)}{\prod\limits_{i=2}^{3} N_{\mathrm{lin},i}^{\Gamma}\left(u\left(\boldsymbol{a}\right)\right)} \cdot \boldsymbol{f}(u_T(\boldsymbol{a})), \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^2.$$

complemented by contributions of the vertex nodes and the curved edge.

Lenoir-mapping

The Lenoir-mapping was initially suggested in [115], who generalized the work of [38, 150], which was at first valid only for quadratic and cubic triangular elements, to any element order and any dimension. The function is defined recursively, thus, the construction of an element with order m approximating a curved boundary relies upon the construction of the $m-1$ elements, starting with a linear mapping up to the element of order $m-1$.

This mapping type is slightly more complicated as the formerly introduced mappings. Due to its recursive nature it can not be written in the convenient form of Eq. (5.7). Instead, Eq. (5.3) is rewritten for an element of order k as

$$\boldsymbol{r}_m^k(\boldsymbol{a}_n) = \left(1 - \sum_{i \in I_{\text{lin}}^{\Omega}} N_{\text{lin},i}^{\Omega}(\boldsymbol{a}_n)\right)^k + \left[\sum_{i \in I_{\text{ho}}^{\Gamma}} N_{\text{ho},i}^{\Gamma,k}(u_T(\boldsymbol{a})) - \sum_{i \in I_{\text{ho}}^{\Gamma}} N_{\text{ho},i}^{\Gamma,k-1}(u_T(\boldsymbol{a}))\right] + . \quad (5.12)$$

$$+\boldsymbol{r}_m^{k-1}(\boldsymbol{a}_n), \quad \forall k > 1 \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^2. \quad (5.13)$$

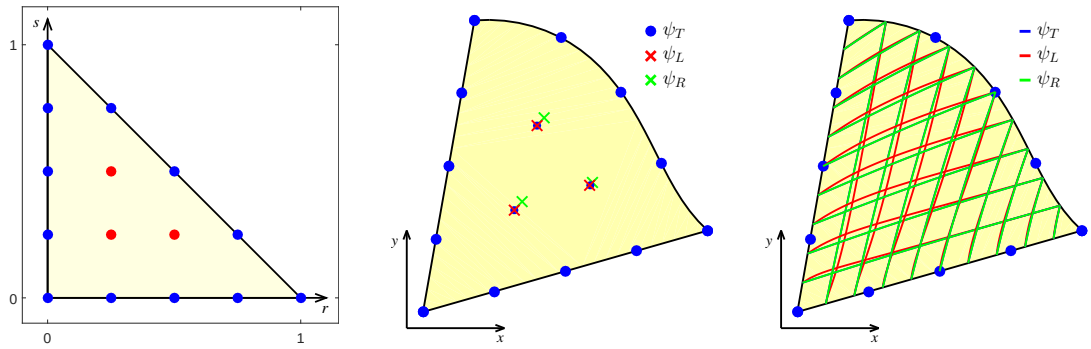
Similar to the transfinite map, this function is valid for all points with exception of the vertex nodes within the element. It can be proven that all expressions in Eq. (5.12) are polynomials of degree $< k$, which is useful as a basic regularity property in [38]. Note that this particular mapping is valid exclusively for an isoparametric description of the boundary, which was not the case for the former two. To extend this function to general parametrizations of the boundary minor modifications of Eq. (5.12) are necessary.

Comparison of different mapping strategies

For a comparison of the three presented mappings, for an element of order $m = 4$ and three points in element interior, in fact the three nodal points, see Fig. 57. In Fig. 57(a) the points of interest are shown in the initial, undeformed reference element; in Fig. 57(b) the three points are shown in the deformed element in physical space and in Fig. 57(c) grid lines for all mappings are visualized. Apparently, the results show that the transfinite mapping and the Lenoir map give essentially identical results, whereas the blending function mapping differs slightly visually, but with important disadvantages in the resulting convergence properties (as shown below). We emphasize that the transfinite mapping as well as the Lenoir mapping result in the best interpolation properties, but in terms of efficiency the Lenoir map is worse than the transfinite mapping. The blending function mapping on the contrary results in significantly worse approximation properties, which can be traced back to the smoothness of the map. For a comparison of the regularity of the transfinite map and the blending function mapping, shown on the example of the Jacobian of the mapping, see Fig. 58(c) and (d), in particular notice the singularity in Fig. 58(c).

Remark 1 In case of straight edges the enrichment function $\boldsymbol{M}(\boldsymbol{a}) = \boldsymbol{0}$ for all mapping types and consequently a affine map is obtained.

Remark 2 It is possible to construct other types of mappings, as shown in e.g. [82, 74]. In addition to the already introduced mapping, an intersection mapping is shown. Based on the projection of the desired point on the element edges, the intersection of the very same points in the curved element is found and taken as the new point. Thereby, there is no explicit mapping, the procedure is rather based on geometrical conception. The clear drawback of the method is the lack of extendability to the three dimensional case. However, it performs very well in the two-dimensional case of triangular elements.



(a) (b) (c)

Figure 57: (a) Triangular reference element for order $m = 4$, the blue nodes are given as boundary data in physical space, the red points are mapped to the physical space in (b) for different choices of ψ. (c) shows a mapping of grid-lines for all introduced maps.

5.3.2 Mapping for quadrilateral elements

For quadrilateral elements, the optimal mapping is found utilizing the so-called blending function method. It was initially introduced by [86] to represent transfinite, that is, general non-polynomial edge parametrizations as circle arcs and other conical sections *exactly*. But the method can also be used for isoparametric edge approximations to include given boundary information, in fact, the standard isoparametric mapping is just a special case of transfinite mappings. The initial situation is shown in Fig. 59(a), with the higher-order interface element subdividing an reference element into an element patch of two elements sharing one curved edge.

A scalar vertex or weight function is defined by

$$\psi(\boldsymbol{a}) = b \qquad \psi(\boldsymbol{a}) = R_2(\boldsymbol{a}) = \sum_{i=2}^{3} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a}), \tag{5.14}$$

the ramp function is unity on the curved edge (edge number two), vanishes on the opposite edge (edge number four) and it varies linearly in-between, as shown in Fig. 60(d). The edge contribution $\boldsymbol{r}^e(\boldsymbol{a})$ for quadrilaterals is defined as the linearly blended map

$$\boldsymbol{r}^e(\boldsymbol{a}) = R(\boldsymbol{a}) \cdot \boldsymbol{f}(u_Q(\boldsymbol{a})).$$

For quadrilateral elements the parametrization of edge two is given by $u_Q(\boldsymbol{a}) = b$. It is seen that $\boldsymbol{f}(u_{Q,i}) = \boldsymbol{0}$ for $i \in I^{\Gamma}_{\mathrm{lin}}$ and, hence, also for the nodes in $I^{\Omega}_{\mathrm{side}}$. Furthermore, $R(\boldsymbol{a}_i) = 0$ for $i \in I^{\Omega}_{\mathrm{lin}} \setminus I^{\Omega}_{\mathrm{side}}$, i.e. for all remaining nodes in $I^{\Omega}_{\mathrm{lin}}$. Again, if edge two is straight, the edge contribution $\boldsymbol{r}^e(\boldsymbol{a}) = \boldsymbol{0}$ and therefore , and a standard low-order isoparametric mapping is obtained.
Given some arbitrary parametrization of edge number two, we see in Fig. 60(c) on the deformation of the grid-lines that the map appears to be quite regular, which is confirmed when inspect-

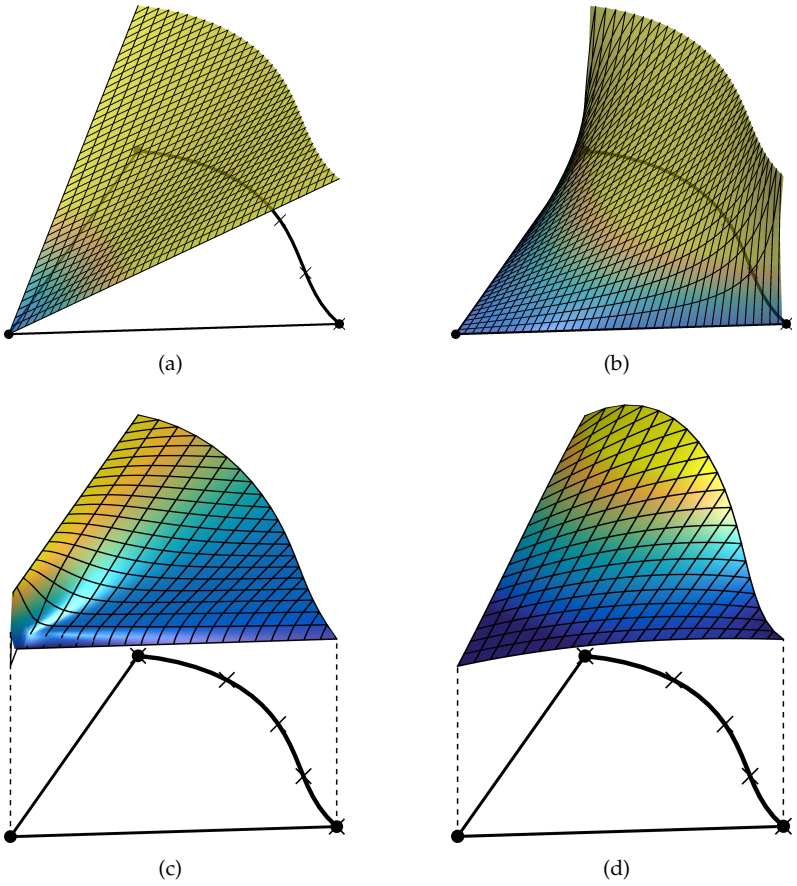


Figure 58: (a) Ramp function as used for the blending function mapping and (b) the function that defines the transfinite mapping, both of them on a triangular element in physical space for $m = 4$. In (c) and (d) the Jacobian of the mappings is shown, notice in particular the singular point for the blending function mapping in (c).

ing the Jacobian of the mapping in Fig. 60(e). Rewriting the whole equation reads as

$$\boldsymbol{r}_m\left(\boldsymbol{a}_n\right) = \sum_{i \in I^{\Omega}_{\text{lin}}} N^{\Omega}_{\text{lin},i}\left(\boldsymbol{a}_n\right) \cdot \boldsymbol{r}^{\Omega}_{\text{lin},i} + \frac{\prod_{i=2}^{3} N^{\Omega}_{\text{lin},i}\left(\boldsymbol{a}\right)}{\prod_{i=2}^{3} N^{\Gamma}_{\text{lin},i}\left(u_Q\left(\boldsymbol{a}\right)\right)} \cdot \boldsymbol{f}(u_Q(\boldsymbol{a})), \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^2.$$

The structure of the mapping equation is essentially the same as the one for triangular elements, only the mapping of internal points from interior onto the boundary is different.

5.3.3 Mapping for tetrahedra

The mapping procedure for tetrahedra follows the line of the approach applied to the triangular elements. An additional challenge is that for cut elements, in general, not only the edges, but also the faces are curved. We emphasize that in this work the edges as well as the faces are parametrized by isoparametric elements, which is not mandatory for the general approach. Actually, the procedure works also for any type of edge and face parametrization as e.g. splines.

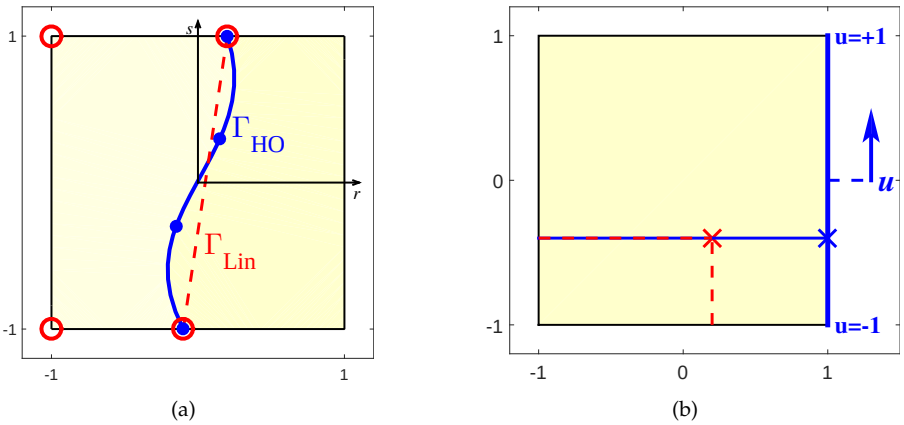


Figure 59: Cut quadrilateral reference element with the reconstructed higher-order line element Γ_{HO} with the indicated topology $\boldsymbol{Q}_{13}$ in (a). In (b) the used local coordinate systems for the parameterized curved edge number two is shown. The edge is parametrized by an 1D element with the element domain ranging from $-1 \leq u \leq 1$.

An implementational issue of elements in three dimensions concerns the parametrization of the faces. The faces, restricted to the element edges, *must* match the parametrization of the curved edges. This includes also the compatibility of the orientation of the element edge and the element face restricted on the edge. The extension of the parametrized curved face into the element domain, the blending, is done using the same three techniques as for the triangle.
Before describing the individual methods, it is necessary to re-define the projection of interior points onto the element boundary. As before, multiple variants can be devised for this task. In analogy to the triangles the first method is to project the point perpendicular to face four, see Fig. 61(b). But if the transfinite maps are applied to the tetrahedra some additional considerations need to be made. In fact, in addition to the map from the interior to the (curved) faces it is necessary to define a map from the interior to the (curved) *edges*. The other possibility, also in analogy to the triangles, is to connect the coordinate origin, thus node number one, with the desired point within of the element and extend the line to the point of intersection with the boundary, this is depicted in Fig. 61(b). The first map (for the transfinite class), from the element interior to the element edge between the nodes i and j is given in continuous and discrete form, respectively, as

$$u_{P,\alpha^e} = \frac{\lambda_j - \lambda_i + 1}{2} \qquad u_{P,\alpha^e} = \frac{N^{\Omega}_{\mathrm{lin},j}(\boldsymbol{a}) - N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a}) + 1}{2}, \tag{5.15}$$

with λ_i being the affine coordinates of the reference tetrahedron. The second map (again for the transfinite class), from the element interior to the curved element face defined by the nodes with the indices (i, j, k), is defined as

$$u_{P,\alpha^f} = \lambda_j - \frac{\lambda_i}{3}; \quad v_{P,\alpha^f} = \lambda_k - \frac{\lambda_i}{3} \qquad u_{\alpha^f} = N^{\Omega}_{\mathrm{lin},j}(\boldsymbol{a}) - \frac{N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a})}{3}; \quad v_{P,\alpha^f} = N^{\Omega}_{\mathrm{lin},k}(\boldsymbol{a}) - \frac{N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a})}{3}. \tag{5.16}$$

The second type of projection, used for the blending function mapping, is given as

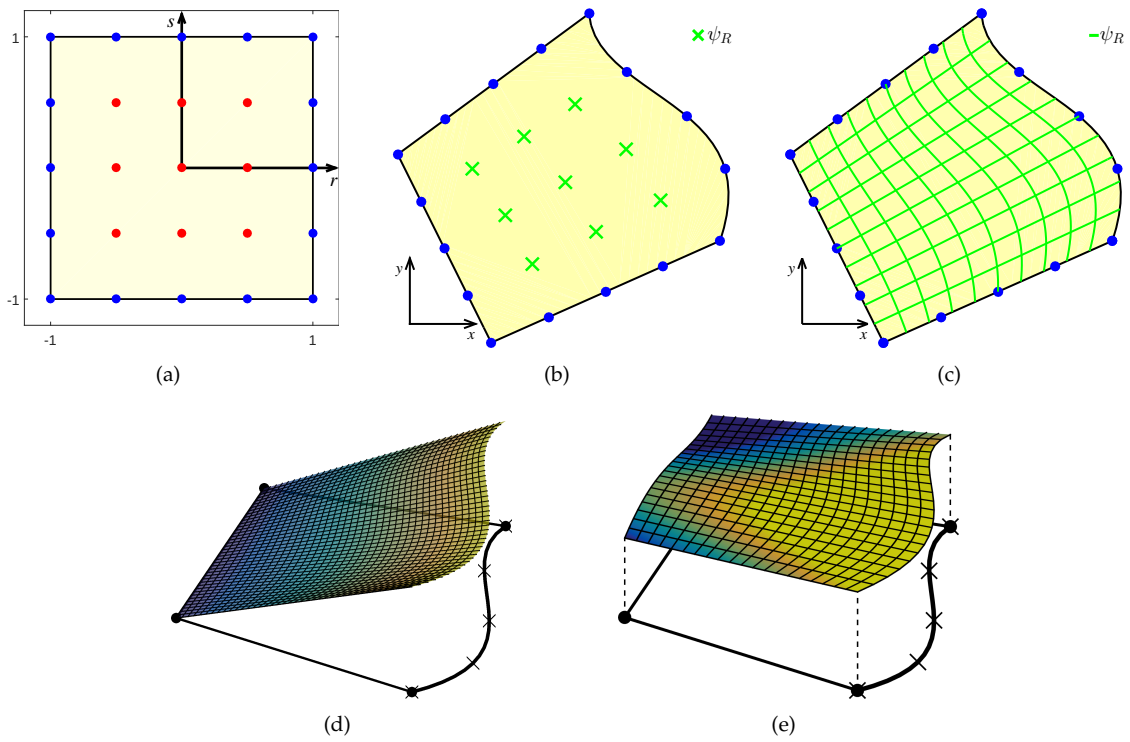


Figure 60: (a) Quadrilateral reference element of order $m = 4$ with boundary (blue) and internal (red) nodes. (b) The deformed element in physical space given only the boundary data, including the curved edge. The internal points are defined by the blending function mapping. (c) The same element and the deformed grid-lines using the blending function map. (d) and (e) The weight function ψ and the Jacobian of the weight function, respectively.

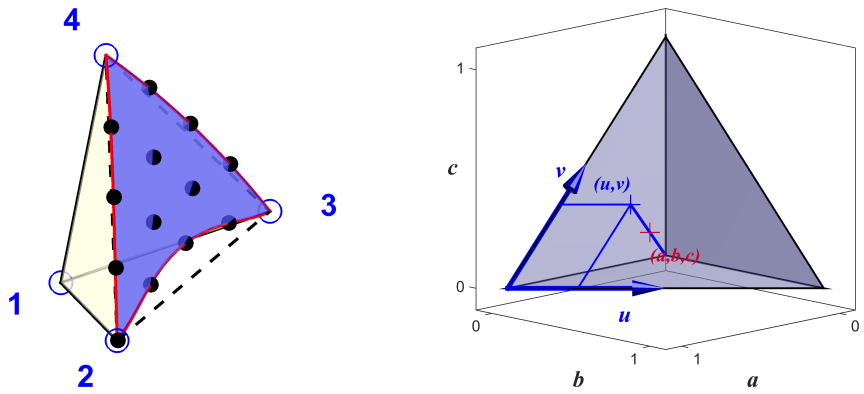
$$u_{P,\beta} = \frac{b}{a+b+c} \quad v_{P,\beta} = \frac{c}{a+b+c} \qquad u_{P,\beta} = \frac{N^{\Omega}_{\text{lin},2}(\boldsymbol{a})}{\sum_{i=2}^{4} N^{\Omega}_{\text{lin},i}}; \quad v_{P,\beta} = \frac{N^{\Omega}_{\text{lin},3}(\boldsymbol{a})}{\sum_{i=2}^{4} N^{\Omega}_{\text{lin},i}} \tag{5.17}$$

Note, that for the mapping $u_{P,\alpha}$ the origin of the coordinate system, which is located in node number one, represents again a singular point. Es evident from Fig. 61(b) in case of tetrahedra, in general, three curved edges arise, this amounts to four curved faces. The procedure it not restricted to this particular situation, in case that all edges are curved, the mapping function is applied sequentially to all edges and faces.

Blending function mapping

The blending function mapping for tetrahedra is the straightforward generalization of the preceding concepts for triangular elements and, therefore, consequently inherits also the regularity characteristics from the triangles. The weighing function ψ can be written as

$$\psi(\boldsymbol{a}) = a + b + c \qquad \psi(\boldsymbol{a}) = R_2(\boldsymbol{a}) = \sum_{i \in I^{\Omega}_{\text{lin}}} N^{\Omega}_{\text{lin},i}(\boldsymbol{a}), \tag{5.18}$$



(a) (b)

Figure 61: Tetrahedral element with the reconstructed higher-order line face, in our convention face number four Γ_{HO}. (b) The used local coordinate systems mapping points from element interior onto the parameterized curved face are shown. The face is parametrized by a standard triangular element with the element domain ranging from $0 \leq r, s, r+s \leq 1$.

notice that the index on the ramp function refers now to the face number. Given the deviation from the linear element, denoted by $\boldsymbol{f}(\boldsymbol{u}(\boldsymbol{a}))$, the final mapping is given as

$$\boldsymbol{r}_m\left(\boldsymbol{a}_n\right) = \sum_{i \in I^{\Omega}_{\text{lin}}} N^{\Omega}_{\text{lin},i}\left(\boldsymbol{a}_n\right) \cdot \boldsymbol{r}^{\Omega}_{\text{lin},i} + \sum_{i=2}^{4} N^{\Omega}_{\text{lin},i}\left(\boldsymbol{a}\right) \cdot \boldsymbol{f}(u(\boldsymbol{a})), \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^3.$$

The mapping is valid for all points with the exception of the origin, where a division by zero arises. For this point the weight function is set to zero and the mapping of the vertex is defined by the standard isoparametric mapping.

Transfinite mapping

The presentation here follows the argumentation as given in [157]. Recall that the total mapping is composed of three separate parts, the vertex, edge and face contributions. The weight function ψ^e for the edge contribution, in continuous and discrete form, respectively, is then

$$\psi^e(\boldsymbol{a}) = \frac{r \cdot s}{u_{T,\alpha}(\boldsymbol{a}) \cdot (1 - u_{T,\alpha}(\boldsymbol{a}))} \qquad \psi^e(\boldsymbol{a}) = \frac{\prod\limits_{i \in I^{\Omega}_{\text{lin}}} N^{\Omega}_{\text{lin},i}\left(\boldsymbol{a}\right)}{\prod\limits_{i \in I^{\Gamma}_{\text{lin}}} N^{\Gamma}_{\text{lin},i}\left(u_{T,\alpha}\left(\boldsymbol{a}\right)\right)}. \tag{5.19}$$

It is important to notice that this edge contribution is for one edge only, therefore Eq. (5.19) must be applied to all three curved edges. As for the triangles, in the vertices of the curved face the weight function ψ^e is set to zero. The face contribution to the transfinite interpolant is defined in continuous and discrete form as

$$\begin{aligned} \psi^f(\boldsymbol{a}) &= \frac{r \cdot s \cdot t}{u_{P,\alpha}(\boldsymbol{a}) \cdot v_{P,\alpha}(\boldsymbol{a}) \cdot (1 - u_{P,\alpha}(\boldsymbol{a}) - v_{P,\alpha}(\boldsymbol{a}))} \\ \psi^f(\boldsymbol{a}) &= \frac{\prod_{i=2}^{4} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a})}{u_{P,\alpha}(\boldsymbol{a}) \cdot v_{P,\alpha}(\boldsymbol{a}) \cdot (1 - u_{P,\alpha}(\boldsymbol{a}) - v_{P,\alpha}(\boldsymbol{a}))} \end{aligned} \tag{5.20}$$

The mapping is valid for all points, except of points located directly on the edges of the curved face, where a division by zero arises. For these points, however, the weight function is, as usually, set to zero and the mapping is defined solely by the remaining part of the mapping. The final mapping is then

$$\begin{aligned} \boldsymbol{r}_m(\boldsymbol{a}_n) = & \sum_{i \in I^{\Omega}_{\mathrm{lin}}} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a}_n) \cdot \boldsymbol{r}^{\Omega}_{\mathrm{lin},i} + \sum_{e=3}^{5} \frac{\prod_{i \in I^{\Omega}_{\mathrm{lin}}} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a})}{\prod_{i \in I^{\Gamma}_{\mathrm{lin}}} N^{\Gamma}_{\mathrm{lin},i}(u_{T,\alpha}(\boldsymbol{a}))} \cdot \boldsymbol{f}(u(\boldsymbol{a})) + \ldots \\ & \frac{\prod_{i=2}^{4} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a})}{u_{P,\alpha}(\boldsymbol{a}) \cdot v_{P,\alpha}(\boldsymbol{a}) \cdot (1 - u_{P,\alpha}(\boldsymbol{a}) - v_{P,\alpha}(\boldsymbol{a}))} \boldsymbol{f}(u(\boldsymbol{a})), \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^3. \end{aligned}$$

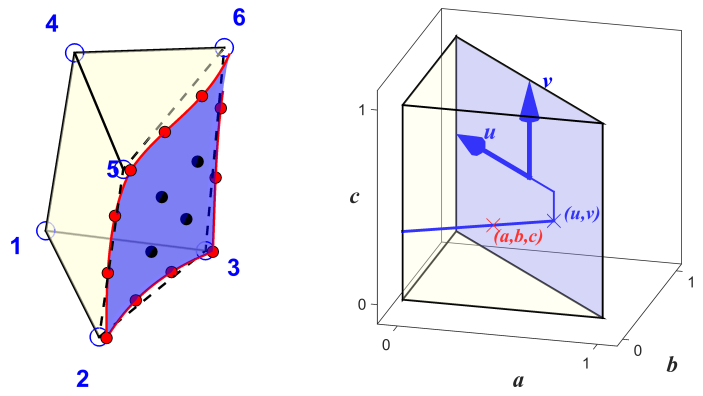
separated by contributions of the vertex nodes and the curved edges and faces. Notice that the last part, taking into account the bubble function on a curved face, is defined only on edge number 4, all other faces are curved, but the bubble part is zero.

5.3.4 Mapping for prisms

Prismatic elements are not used as elements for the background mesh, but they appear for certain topologies in cut tetrahedral elements. The peculiarity when considering prism elements is that the curved face might be either a triangular or a quadrilateral element. As before, it is important that the orientation of the parametrized surfaces fits the definition of curved element faces. If the curved surface is a quadrilateral element, as shown in Fig. 62(a), for any point in the element interior the cross-section parallel to the plane defined by the $a - b$ axis is a triangle, see Fig. 62(b). The mapping onto the quadrilateral face can be executed considering only the triangular cross-section, that is, the mapping defined before for triangular elements applies. For simplicity reasons and due to the fact that it results in better interpolation properties, we consider only the map normal to the second edge, which was referred to as $u_{T,\alpha}(\boldsymbol{a})$.
The final mapping, containing all contributions is given as

$$\boldsymbol{r}_m(\boldsymbol{a}_n) = \sum_{i \in I^{\Omega}_{\mathrm{lin}}} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a}_n) \cdot \boldsymbol{r}^{\Omega}_{\mathrm{lin},i} + \sum_{e=3}^{6} \frac{\prod_{i=2}^{3} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a})}{\prod_{i=2}^{3} N^{\Gamma}_{\mathrm{lin},i}(u_T(\boldsymbol{a}))} \cdot \boldsymbol{f}(u(\boldsymbol{a})) + \frac{\prod_{i=2}^{3} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a})}{\prod_{i=2}^{3} N^{\Gamma}_{\mathrm{lin},i}(u_T(\boldsymbol{a}))} \cdot \boldsymbol{f}(u(\boldsymbol{a})), \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^3.$$

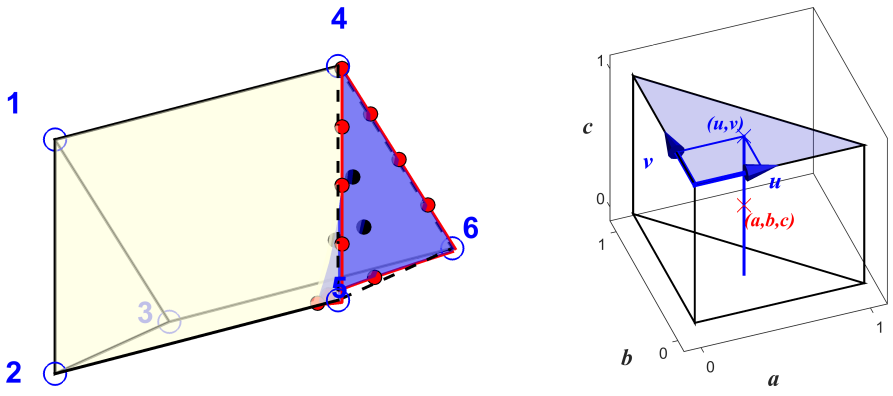
In case that the curved surface is a triangular element, see Fig. 63(a), any vertical cross-section is a quadrilateral. Then, without any difficulty, the resulting bubble can be blended in the element



(a) (b)

Figure 62: The search path $\{\boldsymbol{P}, \boldsymbol{N}\}$ defines a valid line interval in the element between the two intersection points $\boldsymbol{r}_\tau$ and $\boldsymbol{r}_\omega$ on the element boundary and element edge, respectively. The starting point $\boldsymbol{r}^0$ and the final root must be on this line interval.

interior using a simple ramp function, as used for quadrilateral elements. The ramp function is defined as unity on the curved surface and vanishes on the opposite triangular face.



(a) (b)

Figure 63: The search path $\{\boldsymbol{P}, \boldsymbol{N}\}$ defines a valid line interval in the element between the two intersection points $\boldsymbol{r}_\tau$ and $\boldsymbol{r}_\omega$ on the element boundary and element edge, respectively. The starting point $\boldsymbol{r}^0$ and the final root must be on this line interval.

Finally, the definitive mapping for this type of prism elements is given as

$$\boldsymbol{r}_m(\boldsymbol{a}_n) = \sum_{i=2}^{3} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a}_n)\cdot\boldsymbol{r}^{\Omega}_{\mathrm{lin},i} + \frac{\prod_{i=2}^{3} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a})}{\prod_{i=2}^{3} N^{\Gamma}_{\mathrm{lin},i}(u_T(\boldsymbol{a}))}\cdot\boldsymbol{f}(u(\boldsymbol{a})) + \frac{\prod_{i=2}^{3} N^{\Omega}_{\mathrm{lin},i}(\boldsymbol{a})}{\prod_{i=2}^{3} N^{\Gamma}_{\mathrm{lin},i}(u_T(\boldsymbol{a}))}\cdot\boldsymbol{f}(u(\boldsymbol{a})), \quad \boldsymbol{a}_n, \boldsymbol{r}_m \in \mathbb{R}^3.$$

5.3.5 Mapping for hexahedra

The case of curved hexahedra is not considered in this work. Similar to the reconstruction task, there is always the possibility to decompose the hexahedra into tetrahedra and apply the tetra-

hedral mapping to the emerging sub-elements. However, at least conceptually no difficulties should occur as the blending function mapping (as applied to quadrilaterals in 2D) should be transferable to the case of hexahedra without causing problems, see [157].

5.4 Integration rules in cut elements

By means of the introduced mappings, in particular the transfinite mapping, any point $\boldsymbol{a}_i$ in a reference sub-element can be mapped to the corresponding point $\boldsymbol{r}_i$ in the cut reference background element. For the purposes of this section, these points are integration points that are used to evaluate the integrands in the weak form. For all described sub-elements (triangles and quadrilaterals for dim $= 2$ or tetrahedra and prisms for dim $= 3$), the position of integration points and their corresponding integration weights, i.e. tuples $(\boldsymbol{a}_i, w_i^{\boldsymbol{a}}) \in \mathbb{R}^{\dim}$, are needed. As mentioned, the location of the integration points $\boldsymbol{r}_i$ in the cut reference elements are given by the introduced mapping procedures, the integration coefficients, however, are defined as

$$w_i^{\boldsymbol{r}} = \det\left(\mathbf{J}_{\boldsymbol{r}}\left(\boldsymbol{a}_i\right)\right) \cdot w_i^{\boldsymbol{a}}.$$

It is seen that the Jacobian of the mapping $\boldsymbol{r}\left(\boldsymbol{a}\right)$ is needed in order to determine the integration weights in the cut element. When the integration points $(\boldsymbol{r}_i, w_i^{\boldsymbol{r}})$ are further mapped to the global coordinates $\boldsymbol{x}\left(\boldsymbol{r}\right)$, using an isoparametric mapping implied by the shape functions in the cut higher-order element the weights are again modified once more as

$$w_i^{\boldsymbol{x}} = \det\left(\mathbf{J}_{\boldsymbol{x}}\left(\boldsymbol{r}_i\right)\right) \cdot w_i^{\boldsymbol{r}},$$

to obtain the final points $(\boldsymbol{x}_i, w_i^{\boldsymbol{x}})$; $\mathbf{J}_{\boldsymbol{x}}$ is the Jacobi matrix of the isoparametric mapping to the physical space.

It is now clear how the integration weights are treated and we turn our focus to the positions of integration points in the reference sub-elements in $\boldsymbol{a}$. In order to choose suitable integration rules, we rely on numerical studies rather than on tedious mathematical investigations.

6 Conformal higher-order decomposition finite element method

Once the reconstruction of the zero level curve by interface elements (see Section 4) is completed, the next logical step is to construct mappings valid for any point on the two sides of the cut elements. (Section 5). We accentuate that the important point is the regularity and consistency of these mappings. If the mapped points are integration points, the resulting "mesh" is capable to *integrate* non-smooth functions with the discontinuity defined by the zero level set. The discontinuity may be weak or strong and the "mesh" is only a loose distribution of integration points with no requirements on continuity along element edges or faces. This "integration mesh", which subdivides the domain along the discontinuity can then be processed further in any fictitious domain method as e.g. CutFEM or FCM. This is in particular important for CutFEM-applications, because a boundary term appears in the weak form in addition to the standard domain integrals, see Section 3. The automatically generated interface mesh then yields the integration points for the boundary integration.

The new idea is to use the interface reconstruction as well as the mappings as basis building blocks to construct a new finite element method of higher accuracy which is a hybrid between fictitious domain methods and classical finite element methods. The proposed method was introduced in [136] in a 2D-context with some extensions in [74]. Other approaches which are conceptually related to the proposed method were presented in [117, 133, 107]. In the second publication the name *conformal decomposition finite element method* (CDFEM) was coined which is also used herein. The authors classified the CDFEM therein as a special type of XFEM, in particular a so-called nodal enrichment was applied. The fundamental idea was that for linear elements only additional nodes on the interface appear, and they were interpreted as degrees of freedom stemming from an XFEM-like enrichment. However, the authors treated linear elements only and even if for low-order elements the method might be interpreted in this way, for higher-order elements this point of view is clearly not helpful any more. Other, low-order yet conceptually similar approaches are discussed in [102, 4].

6.1 Outline of the method

Let us give an overview of the proposed methodology. The underlying idea is to discretize the zero level set by interface elements which subsequently act as element edges or element faces and employ the special mappings from Section 5 to map *element nodes* from the reference element to the physical domain. Then, the continuity requirements on the mesh are fulfilled naturally, that is, no hanging nodes appear, and it possible to construct automatically an interface conforming higher-order accurate discretization. The newly proposed method is able to treat weak discontinuities (or even strong discontinuities) as well as domain boundaries in a consistent manner. In fact, the resulting conforming higher-order mesh can be processed by any finite element solver, and the results indicate a true higher order behaviour at least up to an polynomial order of $p = 5$ (higher orders have not been studied herein). Actually, the methodology is valid also for higher orders, but in case of Lagrange elements with an equidistant node distribution Runge's phenomena may interfere at $p > 5$. The whole process and all involved mappings and transformations are visualized in Fig. 64 in one diagram.

The diagram is conceptually similar to the one shown in the integration section, with the im-

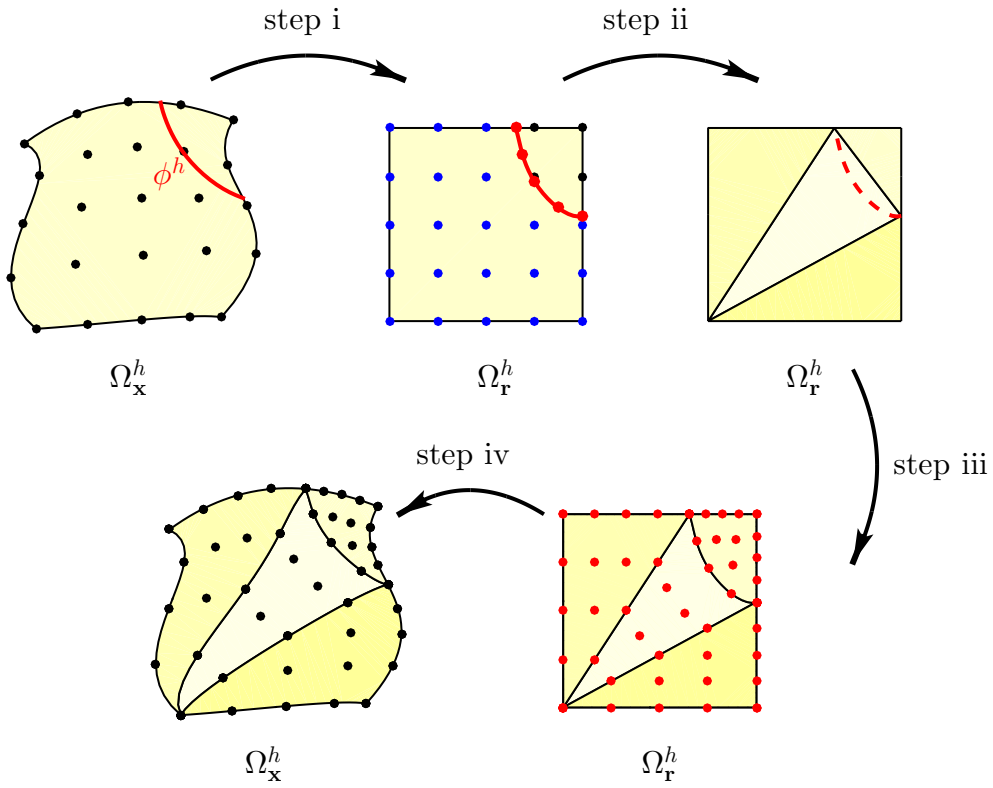


Figure 64: Overview of the complete conformal decomposition and mapping strategy in case that an *interpolated* level set function is used. Notice that the element in the first and the last step is defined in the physical space and all steps in-between are defined in some reference element.

portant exception that the nodes of two neighboring elements now *must* coincide. As before, the strategy has a local (element-wise) character, that is, the reconstruction of the zero isoline advances element by element, see Fig. 64, **step (i)**. This remark might be important if methods from high-performance computing are applied, e.g. parallel integration of the element stiffness matrices for large-scale problems. In fact, if a structured grid is used and the material is homogeneous, the element stiffness matrix for non-cut elements needs to be calculated only once, therefore it is possible to pre-compute essential parts of the system matrix. The stiffness matrix for cut elements, however, can be calculated independently for every element. Note, that each physical element that is cut by the zero level function in the global domain $\Omega_{\boldsymbol{x}}$, the zero isoline is reconstructed in the *reference element* $\Omega_{\boldsymbol{r}}$ in terms of the reference coordinates $\boldsymbol{r}$ and, finally, mapped to the physical domain using an isoparametric mapping. The detour through the reference elements can be avoided for some particular situations, we shall return to the subject in the further course of this section.

The initial situation is an arbitrarily, regular shaped domain bounded by some external interface Γ_{ext} and possibly also featuring an internal interface Γ_{int}, e.g. shown exemplary in Fig. 1. For our approach the distinction between internal and external interfaces is not necessary, therefore we set $\Gamma = \Gamma_{ext} = \Gamma_{int}$. Observe that similarly to the introduced finite element methods in Section 3, the domain boundaries in general do not conform with the spatial discretization, that is, the computational mesh. The goal of this section is to produce automatically a conforming

higher-order accurate mesh, specifically, to discretize possibly curved boundaries and/or sharp features using only the implicitly given level set data in the background mesh.

The starting point of the reconstruction procedure, **step (i)** is to transfer the level set data, which is given in the nodes of the background mesh of order m_Ω in the physical domain $\Omega^h_{\boldsymbol{x}}$, to the corresponding reference domain $\Omega^h_{\boldsymbol{r}}$. Once more, it is important to notice that for the general procedure the reconstruction process is executed completely in the reference element. Based on the information of the level set function in the nodes, elements are detected that are cut by the zero level set. Once the cut elements are identified, it must be ensured that the level set data is valid in the sense that a unique subdivision of the element in a few predefined topologies is possible. The predefined topologies are completely user-defined, and it is quite straightforward to include additional topological situations. If the element is invalid, some special treatment is required, e.g. the local subdivision strategy as introduced in Section 5 or an adaptive refinement in the immediate vicinity of the respective element. For topologically valid cut elements, in **step (ii)** a tailored scalar root finding algorithm is provided, giving any desired number of points on the interface further denoted with n_Γ. As we shall see in the numerical examples the specified search path for the root finding has a large influence on the numerical accuracy of the method. These interface element nodes are subsequently used to *approximate* the zero level set in the reference background elements by higher-order *interface* elements. Obviously, the number of interface element nodes n_Γ corresponds to the specified order of the background element m_Ω. In **step (iii)** the background element is subdivided into a connected number of elements, further referred to as *element patch*. The structure of this element patch depends on the topology which is defined by the interface element. Utilizing the special mappings from Section 5 element-wise, all element nodes are mapped to the element patch. Notice that this mapping is based *only* on the nodes of the interface element. Finally, in **step (iv),** to obtain the corresponding nodal positions in the physical space, the coordinates from the reference element are simply mapped isoparametrically to the physical domain using Eq. (2.33). The automatically obtained conforming mesh fulfills the compatibility requirements for standard FEM-meshes, meaning it can be processed by any standard finite element solver. Additionaly the reconstructed surface mesh (or a line mesh in 2D) is also conforming and therefore perfectly suitable as a computational mesh for PDEs on manifolds.

6.2 A modified version of the method

In order to introduce the general methodology, projections of the level set function onto the corresponding reference element were used to construct the interface element. However, for some special cases this detour via the *reference* elements is not necessary. The modified version is applicable if the following conditions are met

- The background mesh is obtained by an *affine* mapping of reference elements to the physical space. That is, in 2D, all element edges of the background mesh are straight and, in 3D, all element faces are planar.
- Only one level set function is used. Multiple curved zero level sets (or one zero level set with sharp features) can only be handled utilizing the projection to the reference element.

We use the overview diagram in Fig. 65 to clarify the slightly modified procedure. The elements

of the background mesh *must* be undeformed in the physical space and it is mandatory that all elements have straight edges. Even more, in terms of finite element vocabulary, the only permitted deviation from the reference element is a *linear* isoparametric mapping. This statement applies unchanged for all element types and element orders. The level set function may now not even be interpolated (with finite element functions) but rather given *exactly*. In **step (ii)** the projection to the reference element is neglected, instead the zero level set is sought directly in physical space. Once the interface element nodes are found, one of the few predefined topologies from Section 4 is chosen to decompose the element into an interface conforming element patch. Even though the search paths are defined in the physical space, it is still not possible to completely bypass the reference element approach. Namely, to find the other nodes in the element patch, reference elements are projected to the physical space, using only the node distribution of the interface element. As mentioned before, the regularity of this map is crucial for higher-order accurate interpolation and approximation properties. Once this projection is completed, the next cut element is tackled. Notice that for this approach, as for the original version, the mesh continuity requirements are fulfilled automatically.

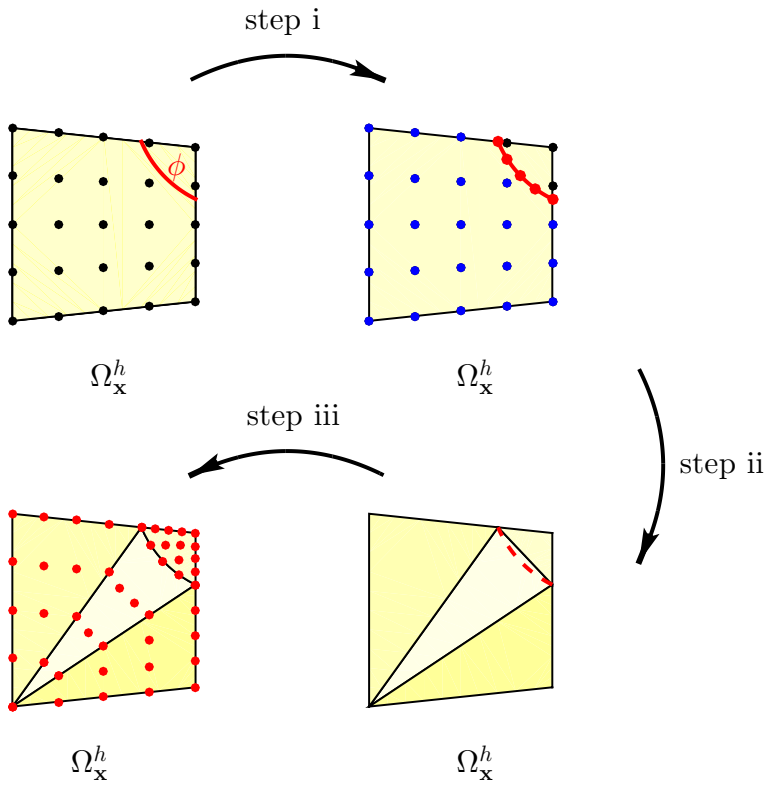


Figure 65: Overview of the modified element decomposition and mapping strategy. For the modified version the *exact* level set function is taken and the reconstruction procedure is defined completely in the physical space.

We mention this approach mainly for the sake of completeness. In fact, there is no overwhelming reason to use the modified procedure in favor of the "original" version. There are definitely some advantages as e.g. the gain in accuracy if the level set function is not interpolated and as the projection to the reference element can be omitted, the numerical errors due to the mapping can-

cel. However, the universality of the original version and especially the ability to handle multiple zero level sets clearly advocate to use the detour over the reference element.

6.3 Treatment of sharp features

One of the major benefits when applying the proposed method is that the approach enables the treatment of sharp features as e.g. corners and edges in a natural manner. Furthermore, it is possible to treat multiple, possibly intersecting, level set functions in the same framework. The basic idea is quite simple: instead of treating all level set functions at once, the procedure is applied successively for each level set function at a time.

Let us assume that multiple level sets are given, for our specific example two isosurfaces $\{\phi_1, \phi_2\}$, and their intersection forms multiple sharp features as visualized in Fig. 66(a)-(d). The decomposition procedure successively processes the given level set functions in the specified order and the decomposed mesh serves finally as a background mesh for the next level set function. As mentioned, this course of action is extendable to n level set functions.

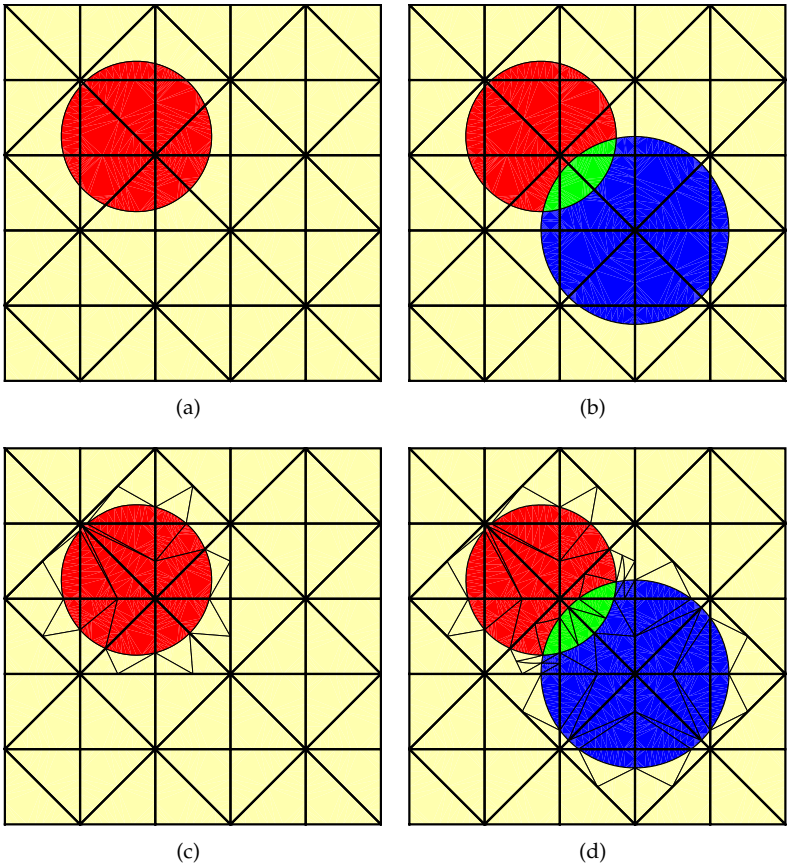


Figure 66: Decomposition technique applied to two overlapping circular zero level sets for approximation (a)-(b) and alternatively only for integration purposes (c)-(d). The triangular background mesh Ω^h is in the first place cut only by ϕ_1 in (a),(c). The decomposed mesh $\Omega^h \cap \phi_1$ serves then as a background mesh and is cut by ϕ_2 in (b),(d). Notice that the procedure is not commutative, another sequence of the employed zero level sets results in a dissimilar final mesh.

We see that the procedure indeed produces automatically conforming higher-order meshes. The method is unrestricted with respect to the background mesh, also deformed background meshes are applicable, and all element types and dimensions. The same approach, to successively tackle one level set after the other, is also valid if only the integration mesh is of interest. In this case the decomposition technique is even easier applicable, as recursive schemes can be employed without taking care of the neighboring elements, see Fig. 66(c)-(d).

Actually it is even advantageous if multiple level set functions are used to represent a sharp feature. If, on the contrary, *one* level set function is used to represent sharp features, e.g. a corner inside an element, it is necessary to customize the background mesh to capture that particular feature of the level set function. In detail, in order to capture a corner the topological requirement on the background mesh is that the corner *must* be located just onto one of the element edges. We mention that the order of the employed zero level sets slightly affects the results, depending on the temporal sequence the final meshes are in general different. Nevertheless, in terms of the convergence rate of the reconstructions the method is indifferent to the sequence of the applied level set functions as the interface feature is always captured.

To summarize, the domain decomposition with respect to several level set functions is a straightforward successive application of the meshing procedures described above, this is true for integration purposes only as well as in a simulation context. The ability to naturally consider an arbitrary number of level set functions as well as to handle sharp features in two as well as three dimensions seems to be a unique feature of the proposed method.

6.4 Influence of the decomposition scheme on the condition number

An obvious consequence of the proposed methodology is the presence of awkward element shapes: Very small inner angles may occur and the size of neighbouring elements can vary strongly. This issue does actually not influence the interpolation properties, which was already addressed in Section 2, but may lead to poorly conditioned system matrices. This concern is well known in the literature and was investigated in a number of publications, see e.g. [66]. Some remedies for this problem in the XFEM-context are to use preconditioning schemes or utilize simple mesh manipulation procedures to prevent very small angles.

The influence of cusp shaped elements on the condition number is shown by means of a boundary value problem in 2D, to be precise the example of an elastic plate with a hole. This particular example is later investigated in more detail in Section 7. The quadrilateral background mesh remains fixed while the element order is varied between $m_\Omega = 1$ and $m_\Omega = 5$. The circular inclusion is moved vertically in an interval of 10^{-4} units and for each step the condition number is computed as a function of the relative displacement, as shown in Fig. 67(b). The lowest element order, $m_\Omega = 1$, is plotted in black color and the largest, $m_\Omega = 5$, is the dotted red line.

Note that the peaks that represent a very high condition number in the system of equations (resulting from a FE-analysis of the plate with hole) emerge only for very few geometries in a small range of the relative displacement. In fact, the condition number can become unbounded for certain combinations of the zero level set and background mesh. However, it is not difficult to identify and modify the nodes in the background mesh which cause these high condition numbers. Some procedures to cure this problem are described below. Herein, for all numerical

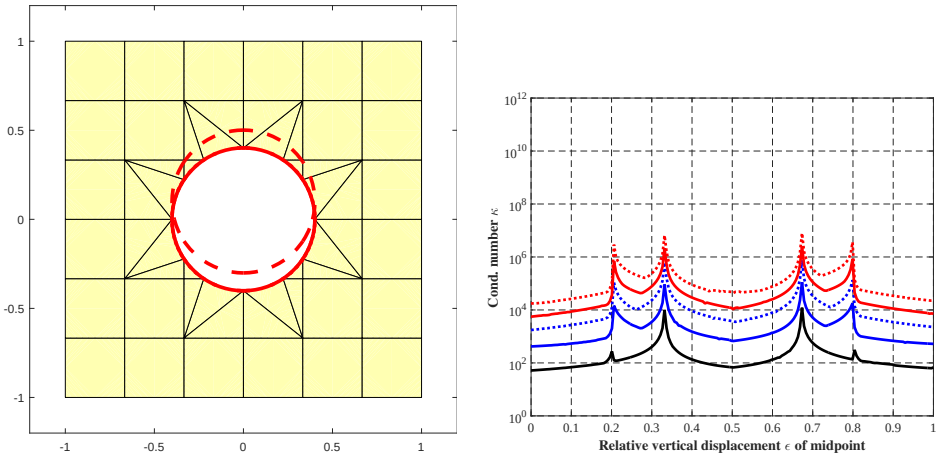


Figure 67: (a) Moving circular discontinuity in a structured quadrilateral background mesh and in (b) the condition number of the system matrix in dependence of the relative vertical displacement of the discontinuity.

examples direct solvers are employed for the solution of the system of equations and we mention that even if none of these techniques are employed, the results are still optimal in terms of convergence rates.

6.5 Remedies to ill-conditioning of the linear system of equations

As mentioned before, the ill-conditioning of the system of equations appears due to a linear dependence of the basis functions with a small support. This is actually not only a problem for the standard finite element method but also a well-known issue in fictitious and embedded domain methods, where the same phenomenon appears. Additionally also discontinuous Galerkin methods suffer from problems with ill-conditioning. The following approaches to cure this issue are known from the literature:

Mesh modification The background mesh is modified in order to increase the support of the functions. In fact, two possible approaches exist: first, to move the nodes away from the zero level set, the idea is described in more detail below. The other possibility is to shift some of the nodes directly *on* the interface. An interesting approach, utilizing node snapping on the zero level set and mesh transformations is shown in [147]. The method is suitable for higher-order approximations but has some serious restrictions. For one it requires a high regularity on the domain, furthermore there are limitations on the background mesh and the extendability to three dimensions is still an open topic.

DoF modification The troublesome basis functions with very small supports are linked with suitable neighboring elements, see [98, 148]. This method retains the convergence properties and scales the condition number to regular meshes. There is also the possibility to completely disregard some of the degrees of freedom, but this approach is not suited for higher-order approximations as it leads to a loss of accuracy.

Edge/Face stabilization Adding some stabilization terms in order to control the gradient over

the edges or faces (2D/3D), see [25, 29]. This idea is taken from the CutFEM approach and originates from discontinuous Galerkin methods. The underlying rationale is that gradient jumps between adjacent elements are penalized over element edges. There exist different possibilities to execute the stabilization, e.g. to apply the full gradient or only the normal gradient. The method can conceptually be extended to higher-orders and three dimensions. Some numerical examples for higher-order approximations in combination with the ghost penalty approach are shown in [113].

6.5.1 Nodal moving procedures

The mesh modification or nodal moving procedures is the most descriptive way to improve the quality of a background mesh in terms of the condition number. The basic concept is to define a narrow band of elements close to the zero level set and to relocate some of the nodes either away from the interface or snap it directly on the interface. We proceed here with the strategy to move it *away* from the interface. The important issues regarding the manipulation procedure are: (i) which elements need to be modified and which element nodes need to be relocated, (ii) in which direction and which distance should the nodes be moved, and (iii) how to ensure the stability of the procedure such that the resulting background mesh has guaranteed shape regular elements. The procedure is applicable to all element types and also to higher-order elements, nonetheless, in order to improve the stability we restrict the procedure to background meshes which are either structured or originating from the deformation of a linear background mesh. For simplicity reasons we choose to manipulate only a narrow band of nodes close to the zero level set, with the width as a user defined input value. The whole procedure is explained in Fig. 68. In Fig. 68(a) a non-modified mesh is shown, with some nodes clearly very close to the zero level set. In Fig. 68(b) the modified mesh is visualized, where a a narrow band around the zero level set with modification on some nodes is highly visible. In Fig. 68(c) the two regions are highlighted, that are defined by the two parameters of the procedure d_{crit} and d_{min}. The region where node modifications are performed in blue color and the zone where no nodes after the manipulations are allowed in red color. In Fig. 68 the original and the modified mesh are visualized, with the modified nodes highlighted with red points.

Concerning the methodology of the manipulation procedure, first the location of the involved nodes, $\boldsymbol{x}_i$, to the approximated zero level set must be calculated. To achieve this, a Newton-Raphson procedure with the starting point in $\boldsymbol{x}_i$ is used to find a location on ϕ^h, referred as $\boldsymbol{x}_i^\phi$. The search direction is chosen with the gradient $\nabla\phi$. Note that the approach bears some similarity to the tailored root search within the element, see Section 4. The sought point $\boldsymbol{x}_i^\phi$ is in general *not* the closest point to the zero level set, but this is nonetheless completely sufficient for our purposes. This is justified by the fact that we assume a rather moderate curvature of the zero level function as we consider only points close to the zero level set. Let us denote the distance $\overline{\boldsymbol{x}_i\,\boldsymbol{x}_i^\phi} = d_i$. If the distance d_i falls below a certain threshold d_{crit} the node is moved a distance $\Delta\boldsymbol{x}_i$ in direction of $\nabla\phi$, away from the zero isoline. In our numerical studies the value d_{crit} is chosen as $d_{crit} = 3 \cdot h$, with h as a characteristic length of an element of the background mesh, the minimum moving distance is specified by d_{min} which is prescribed as $d_{min} = 0.3 \cdot h$. These two parameters, d_{crit} and d_{min} therefore specify for one the manipulation domain, $d_{crit} < d < d_{min}$

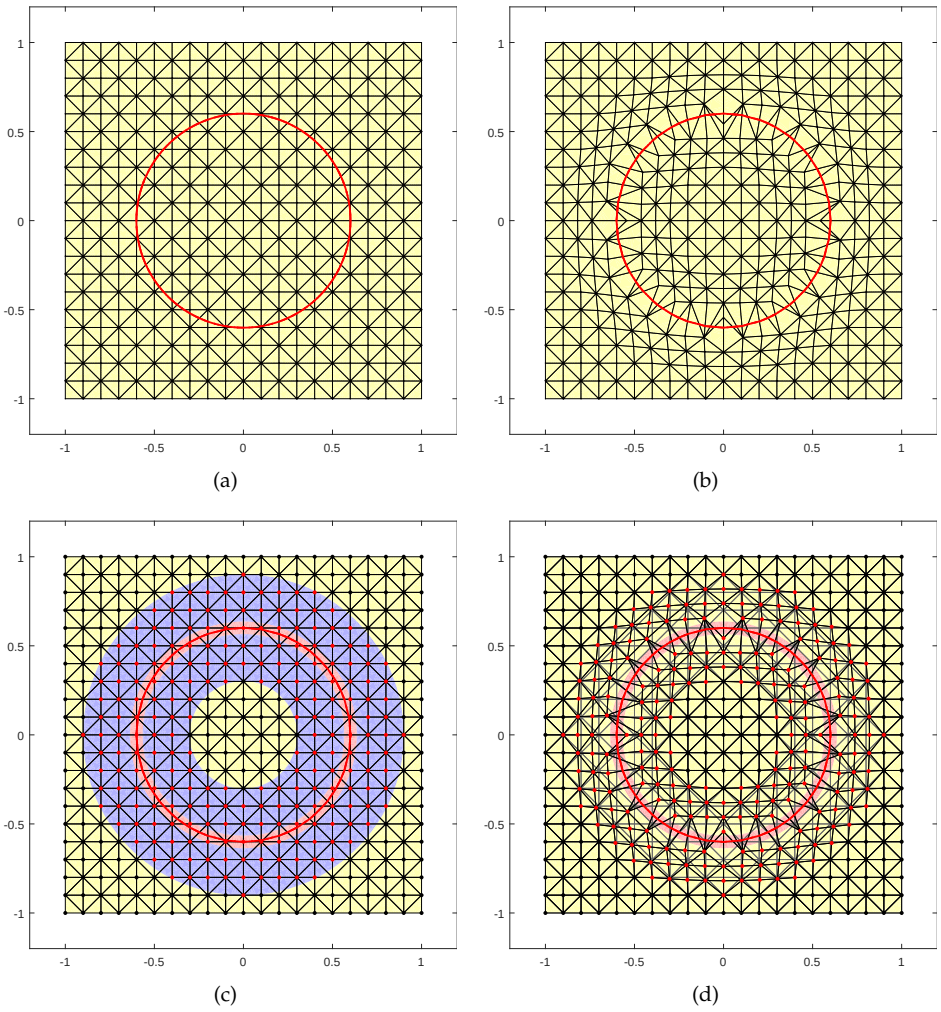


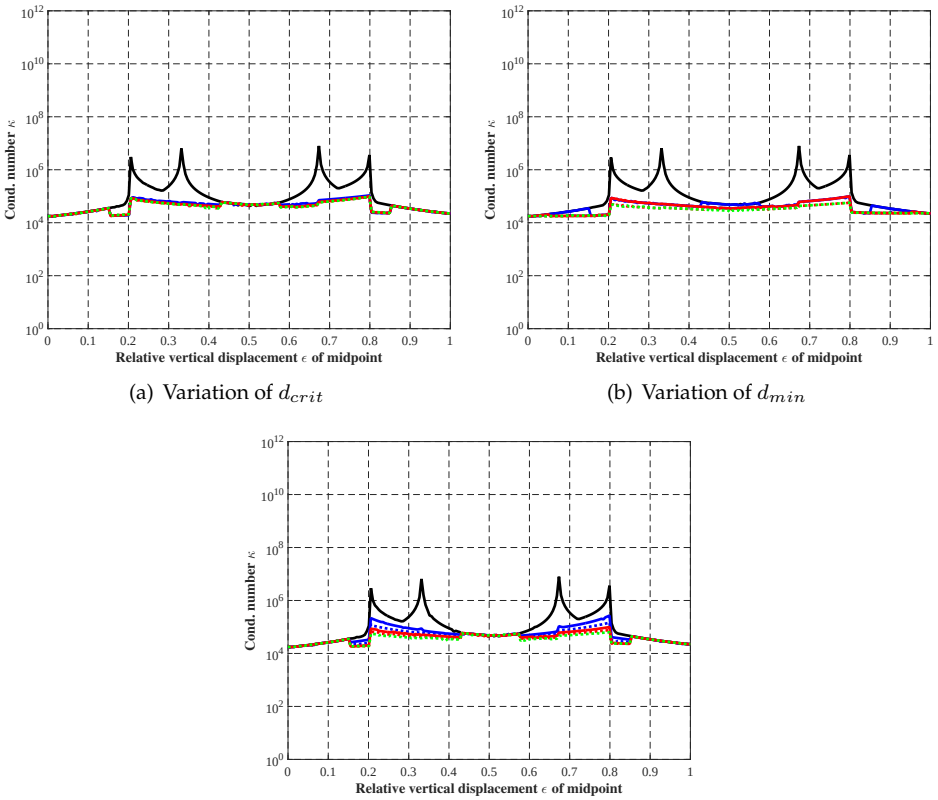
Figure 68: Original background mesh and mesh after node manipulations in (a) and (b), respectively. The zones where the mesh is manipulated are shown in (c) and the difference between original and modified mesh in (d).

and furthermore the domain where no nodes should be present $d_{min} < d < 0$.

To guarantee a certain mesh quality, the element domain ratios are checked, $\alpha = \frac{\Omega_h^{e,max}}{\Omega_h^{e,min}}$, with $\Omega_h^{e,max}$ and $\Omega_h^{e,min}$ as maximal and minimal element area and volumes for 2D and 3D problems, respectively. In order to increase the stability the whole moving procedure is done iteratively, meaning the nodes are not moved at once for the final distance d_{crit} but rather in steps $d_{min} = \beta \cdot d_{crit}$. β scales the maximum distance and is chosen typically with $\beta = 0.2 - 0.3$. Even if it is still possible that awkwardly shaped elements occur, by using the node moving the ratio α is significantly improved and numerical studies confirm that the condition number is then bounded.

We consider the same example as in Fig. 67 for different parameters d_{crit} and different values of β. As the diagrams for different element orders are quantitatively basically equal, we choose here a fixed element order $p = 5$.

As seen from the results, the procedure gives a largely improved distribution of the condition



(a) Variation of d_{crit}

(b) Variation of d_{min}

(c) Variation of β

Figure 69: Variations of the parameters of the nodal-based manipulation procedure: in (a) the maximum distance d_{crit} is varied from $0 - 5 \cdot h$ in intervals of $1 \cdot h$, in (b) the distance where no nodes should appear, d_{min}, from $0.0 - 0.5$ in steps of 0.1, and in (c) the factor of the critical step size β is varied from $0 - 0.5$ in intervals of 0.1.

number $\kappa(\varepsilon)$. The same studies were also performed for three dimensional problems, a sphere which is moved in the ambient tetrahedral mesh, with qualitatively identical results.

6.6 Invalid topologies/meshes

As mentioned in Section 4, the assessment whether a topology is valid or invalid is based solely on level set data on the boundary and does not take into account the shape of the zero isoline inside the element. In some valid configurations the reconstruction may even fail if the curvature of the level set function is too high. A topological situation leading to such a case is shown in Fig. 70(a). The reconstruction, see Fig. 70(b) fails clearly, the appearance of the Runge phenomenon is evident. However, as the mesh is successively refined, see Fig. 70(c) and Fig. 70(d), the situation is finally resolved. For the check, a finite number of points is chosen and the Jacobian is evaluated. Here the points are chosen as the integration points of the elements that emerge from the subdivision scheme but also other choices are possible. The sub-elements resulting from the decomposition are valid if the Jacobian is strictly positive at all discrete points.

Therefore, if the reconstruction of some zero level sets fails due to the curvature of the zero level function the inefficient way to improve the situation is to refine the background mesh glob-

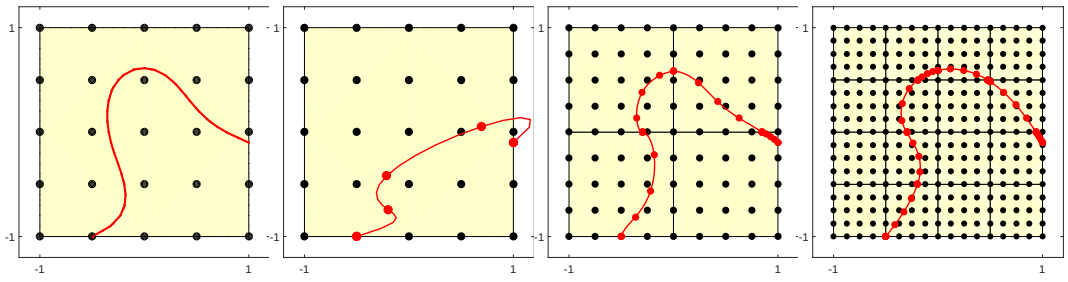


Figure 70: (a) Exact zero level set inside a higher-order quadrilateral element. (b) The reconstruction fails due to the high curvature of the level set. (c) and (d) The problem is resolved by a uniform refinement of the element until no negative Jacobians are detected.

ally until the function is sufficiently resolved. Another way to avoid invalid topologies in an efficient way is to adaptively refine the domain, with the curvature of the zero level set as an indicator for the local characteristic element length h. Especially in case of analytically given level set functions a criterion can be devised to refine cut elements (and their neighbors) according to some curvature measure.

6.7 Conformal higher-order decomposition finite element method for elastoplasticity

The assumption of elastoplastic material behaviour complicates the numerical (and the theoretical) treatment of a boundary value problem considerably. The main cause is that for elastoplastic problems, in addition to the global equilibrium considerations causing a boundary value problem, it becomes necessary to solve an initial value problem (IVP) for *every* point in the domain where the stresses succeed a certain bound (see Section 2). As mentioned, this results, in general, in weak (or even strong) discontinuities in the displacement field, which causes an accuracy loss if no further measures are taken. In addition to the weak discontinuity, for some commonly used elastoplastic models the use of low-order elements causes additional problems originating from numerical locking, see e.g. [167].

In past there were some approaches to tackle elastoplastic problems with p-type finite elements, however no general strategy emerged. For a rather theoretical point of view, including considerations regarding adaptivity, see [89]. Literature with focus on deformation theory of plasticity with a comparison of the h-FEM and p-FEM approach is found [57, 56], in context of rp-adaptivity see [134], in context of fictitious domain methods (FCM) [2]. For localization problems see [104] and an application of the p-FEM within the incremental theory of plasticity is found in [99].

We introduce now the CDFEM in context of elastoplastic problems. We emphasize beforehand that the general approach works fully robustly only for low-order elements. There are some cases, where optimal convergence rates could be obtained, but these are geometries with beforehand known location of the elastoplastic interface, see the numerical examples in Section 7. The main problem, preventing a general higher-order accurate treatment, is an issue with the capturing of the zero level function, which describes the elastoplastic interface, this is described in more detail

below.

6.8 Detection of the elastoplastic zone

As mentioned, the main problem when treating elastoplastic problems is the regularity of the elastoplastic interface. The location of the interface is unknown, and it depends on the numerical discretization. To obtain higher-order accuracy, it is necessary either to enrich the approximation space (e.g. utilizing the XFEM) or to subdivide the mesh along the interface (CDFEM). In any case, it is necessary to capture location of the interface as accurate as possible. To simplify the treatment we consider only the deformation theory of plasticity. This is a rather special case but nonetheless important for the treatment of single-load events. The methodology for the incremental type of plasticity, the more frequently occurring type, is methodologically similar, however, an additional mapping of internal variables from the initial mesh to the refined mesh would be necessary, further complicating the analysis.

In order to describe the elastoplastic interface, the yield function is chosen as the leading parameter for our considerations. In fact, a regular yield function does not represent a level set function due the condition $\phi(\boldsymbol{\sigma}, \boldsymbol{q}) \leq 0$, which is enforced globally. As a consequence, instead of the regular yield function we use the yield function with *trial stresses*, $\phi^{tr}(\boldsymbol{\sigma}^{tr}, \boldsymbol{q}^{tr})$ as the leading function to initiate the level set function. Note, that the yield function is a function of all stress components, which are in general discontinuous along element faces. This means that the yield function and therefore consequently the level set function is *discontinuous* along element faces. However, as with the stresses, we assume that this discontinuity tends to zero for $h \to 0$.

The crucial point is that the information regarding the elastoplastic interface, the stresses and the hardening variables, are given only in the integration points. But as our decomposition scheme utilizes an interpolation of the level set function using the finite element functions, these values are actually needed in the nodes of the background element. Incidentally, the same problem occurs if the elastoplastic problem is tackled using the XFEM. The sought higher-order accurate extrapolation of the level set values, or rather the yield function, to the nodal points is *the* crucial point and a rather difficult topic. The topic of data transfer procedures emerge also for adaptive schemes in plasticity and is anything but new, see [138, 143, 159]. However, there exists no satisfying solution which is applicable for higher-order approximations.

In detail, we investigated different methods as

- Element *local* extrapolation: Using the element mass matrix, the values in the integration points are extrapolated to the element nodal points. This extrapolation strategy works well for element internal nodes, as it also captures the inherent local nature of plasticity. However, at the element edges some discontinuities occur and a naïve averaging destroys the higher-order accuracy of the scheme.

- System *global* extrapolation: Using the system mass matrix, the values are extrapolated globally, therefore enforcing automatically the continuity on the element edges. Nevertheless, it seems that this approach is also not optimal. For one, it does not capture the local nature of the problem and falsifies the values of the zero level function significantly due to the smoothening nature of the procedure. Furthermore, it might happen that some artificial plastic zones are induced when the gradient of the zero level set varies strongly.

- Superconvergent patch recovery: Basically the same method as element local extrapolation, however, the elements are always considered in patches. Discontinuities on the element edges does not occur, however, the method again smoothens the zero level set and is therefore not suited for higher-order approximations.

- Using mixed finite element approach: There are two possibilities in this case, one is to use a rather standard $(\boldsymbol{u}-\boldsymbol{\sigma})$ approach to get the stress-components as nodal values, therefore the stress components are smooth and the yield function value in the nodes can be computed automatically. An alternative is to use a mixed finite element method with the values of the plastic multiplier as an independent variable. This approach was suggested in [146, 154]. However, it is not clear, whether this course of action is applicable to higher-order approximations.

- Inner element root search based on integration points: This approach was introduced in [134] within a rp-refinement scheme. In order to describe the zero level set some NURBS functions are utilized and additionaly the actual reconstruction is performed on a finer background mesh which corresponds with the courser FEM mesh.

- Describe the zero level set within the context of moving boundaries. This approach seems to be the most promising one: it was introduced in context of XFEM discretizations of contact and incompressibility in [23]. There the evolution of the zero level set is described by additional equations. As contact and incompressibility as well as plasticity lead to variational inequalities, the problem types are actually very similar. The drawback of this approach is that the implementation, especially for higher-orders, seems not to be without effort. In addition, the method is based on a functional approach, it is not clear whether it can be transferred easily to the incremental type of plasticity.

6.9 Mesh update strategy

The solution of an elastoplastic problem is due to its inherent non-linearity found within an iterative scheme, usually a Newton-Raphson procedure. Our proposition is to decompose the initial mesh *at the end* of the procedure, i.e. the converged solution of the Newton-Raphson scheme. Let us assume for simplicity reasons that we actually have a boundary fitted mesh and we only deal with an interface problem, the interface being the elastoplastic front. The goal is to decompose the mesh such that the element edges align with the *final* elastoplastic interface. The background mesh is therefore in general non-Cartesian, but this poses no problem for the decomposition utilizing the reference elements.

We start with the initial mesh $\mathcal{T}^0$ and perform a standard elastoplastic analysis. In a post-processing step, the zero level set ϕ^0 is initialized using the trial stresses, therefore neglecting the plastic strains. We mention that for low-order meshes any of the introduced data transfer procedures can be used in order to initialize ϕ^0. In the next step the initial mesh is subdivided along ϕ^0 leading to $\mathcal{T}^1$. Then the whole elastoplastic analysis is performed with $\mathcal{T}^1$ as computational mesh. The level set function is again evaluated and will, in general, not yet align with the element boundaries. Then the new level set ϕ^1 is used to decompose $\mathcal{T}^0$. This whole procedure is embedded within an iterative scheme with the value of the level set function along the newly

created boundary as an error indicator. The scheme is performed as long as the zero level set does not align with the decomposed mesh within a user specified tolerance.

7 Numerical studies

In order to study and illustrate the convergence properties and the stability of the proposed method we present some well known benchmark examples, mostly stemming from the field of solid mechanics. The starting point for all examples are different types of generic background meshes, both structured and non-structured meshes are investigated. We treat problems with non-fitting interfaces, this includes unfitted internal interfaces as well as unfitted boundaries. In the test cases, we focus in the following order on: (i) the quality of the interface reconstructions, (ii) integration of smooth functions on the reconstructed zero level sets as well as within the resulting volumes, (iii) the interpolation and approximation properties of the resulting meshes in the context of boundary value problems with inner-element boundaries and interfaces as well as the conditioning of the system of equations. We treat two-dimensional and three-dimensional problems with an increasing complexity of the zero level function. In addition, some of the results for the boundary value problems obtained by the here presented higher-order CDFEM are compared with a Nitsche-XFEM (NXFEM) version as introduced in Section 3.2.

Firstly, we focus exclusively on the interface reconstruction, that is, the quality of the root finding and meshing given *one* sufficiently smooth zero level set. The here introduced approach can be extended straightforwardly to multiple level sets and may also include sharp features as edges and corners in two and three dimensions, for a thorough discussion we refer to [74]. The reconstruction forms the basis of all further considerations and the accuracy of the reconstruction will significantly affect the integration as well as the interpolation and approximation properties.

The second section investigates the accuracy of the numerical integration schemes *inside* the cut elements as well as *on* the resulting interface mesh. Closely related to the integration accuracy, also the interpolation properties of the mesh are considered. As shown in Section 5 and confirmed by numerical examples, once an optimal type of reconstruction strategy is found, meaning the search path for the element nodes is established, the key point to higher-order convergence rates is the applied type of mapping for the resulting sub-elements within the cut background mesh elements. This is in particularly true for triangles and the three-dimensional generalizations of the triangles, where the naïve approach, the blending function mapping, has considerable negative influence on the interpolation properties.

In the third section boundary value problems for scalar valued as well as for linear elastic and for elastoplastic problems are considered. The examples for scalar problems and linear elasticity are the well known bi-material problem, a circular inclusion in two dimensions and a spherical inclusion in three dimensions. These examples are representative for a number of methods dealing with (weak) discontinuities as e.g. the XFEM. The other class of examples dealing with linear elastic problems is a circular and spherical cavity, for two dimensions and three dimensions, respectively, and is used to demonstrate the capability of the procedure for a problem typical for fictitious domain methods as e.g. the FCM. Both BVPs are often used as benchmarks for the performance of the p-version of the FEM. Finally we show the capability of the method to deal with nonlinear problems on examples with an elastoplastic interface. Very few analytical solutions are available, in two dimensions we focus on the circular elastoplastic interface in a cylindrical domain and a spherical interface in three dimensions.

Along with the approximation properties for BVPs, the influence of awkwardly shaped ele-

ments on the condition number as well as on the accuracy of the scheme is investigated. Most of the examples are compared with an NXFEM approach. It is important to emphasize that the proposed reconstruction scheme yields effectively not only the integration points within the cut elements but also the integration points *on* the interface for the additional terms in the weak form. The utilized stabilization procedure is for both methods the node manipulation scheme, as introduced in Section 6. As shown within the numerical examples, this approach works quite well for higher-order schemes in 2D and 3D.

In all numerical studies presented herein, in order to reduce the number of variables, we focus on meshes where the ansatz order of the background mesh m_Ω and the order of the reconstructed interface mesh m_Γ always *coincide*s, meaning $m_\Omega = m_\Gamma$. Note that this approach is not necessarily needed as these two orders are freely selectable. However, as shown in [73], this choice gives optimal results for the reconstruction as well as for the numerical integration. When different orders are chosen, the overall convergence properties of the resulting meshes are governed by the lowest order.

7.1 Interface reconstruction

The quality of the interface reconstruction governs clearly the interpolation and consequently the approximation properties of the resulting mesh. The first example is a rather simple geometry and demonstrates how the reconstruction quality is influenced by a number of variables, e.g. different element types and different element arrangements and, foremost, different search paths for the root search of the level set function. Based on the results of this section some of the background meshes are later on used for the approximation of a boundary value problem. Therefore, the goal of this section is to evaluate the geometrical error produced *only* by the root search within the reference elements and draw conclusions from this. The second two-dimensional example is geometrically more intricated, a flower-shaped inclusion, and includes parts with a high curvature of the zero level function. In three dimensions we consider analogies to the two dimensional examples, that is, a sphere and the three dimensional analogy to the flower-like inclusion, a surface sometimes referred to as Goursat's surface.

It is emphasized, that the error analysis of the interface reconstruction basically amounts to the integration of a constant function $f(\boldsymbol{x}) = 1$ on the zero level set. In particular, in order to numerically evaluate the error, it is necessary to modify the integration weights using the Jacobian or the Gramian. This topic will be taken up further in Section 7.2, in this section we focus only on the reconstruction quality and take the integration weights for granted.

7.1.1 Numerical results in two dimensions - reconstruction of a circular interface

We continue with the problem as shown in Fig. 71, that is, two non-overlapping domains Ω^+ and Ω^- that are separated by a circular zero level set ϕ. The zero level set, given as

$$\phi(\boldsymbol{x}) = ||\boldsymbol{x}|| - r \tag{7.1}$$

with $||\boldsymbol{x}|| = \sqrt{x^2 + y^2}$ as the Euclidian distance to the current point $\boldsymbol{x}$, can hereby represent any boundary or interface and is denoted as Γ. The radius of the circle is chosen with $r = 0.401$

units and the outer domain boundary in the numerical examples is $[-1,1] \times [-1,1]$. The signs of the two domains indicate the signs of the zero level set within the corresponding domain. The chosen convention here, positive within the inclusion and negative in the surrounding domain, is not mandatory. A change of the sign in the level set function would merely change the direction of the normal vector on the zero level function and consequently the numbering of the interface nodes.

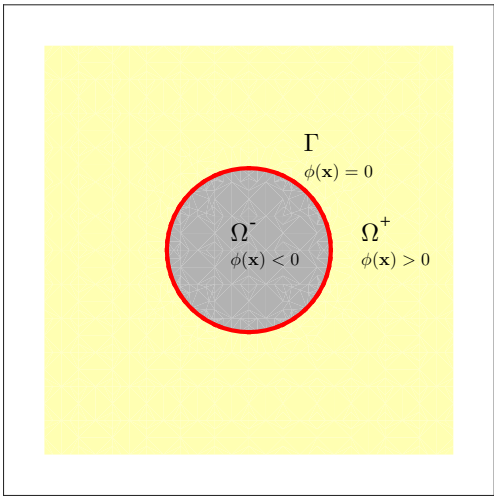


Figure 71: Two domains, Ω^+ and Ω^-, separated by the zero contour of $\phi(\boldsymbol{x})$, here denoted with Γ. The signs of the two domains correspond with the signs of the level set function ϕ within the domain.

Integrated zero level set error for different search directions

In Section 4 we introduced a number of different search paths for the Newton-Raphson iteration within the reference element, varying in search direction and starting point. To recall, for triangular elements variant 1 and variant 2 use one special node and evenly distributed nodes on the opposite edge and evenly distributed points on the linear reconstruction, respectively. Variant 3 and variant 4 use both a linear reconstruction and the search direction is for one the normal vector to the linear reconstruction and the other the gradient on the level set function. The task is to extract the most accurate and stable search path for the investigations that will follow. For this study we use a structured triangular background mesh as shown in Fig. 74(a). It is noted that also quadrilateral meshes were investigated, as e.g. Fig. 74(b), but as the qualitative results of triangular and quadrilateral meshes qualitatively coincide, only the results of the triangular meshes are shown.

For the sake of clarity in most of the forthcoming figures the element nodes of the background mesh are not plotted. Notice that the mesh configuration for triangles always arises by subdividing one quadrilateral in a criss-cross pattern into two triangular elements. In the convergence studies, the sequence of the background meshes is nested, meaning the number of elements per side doubles in each refinement step. For the two dimensional studies, $\{5, 10, 20, 40, 80, 160, 250\}$ elements per dimension are employed. In the diagrams the element size is indicated by the characteristic element length $h = diam(K)$, measured in the background mesh *before* the element

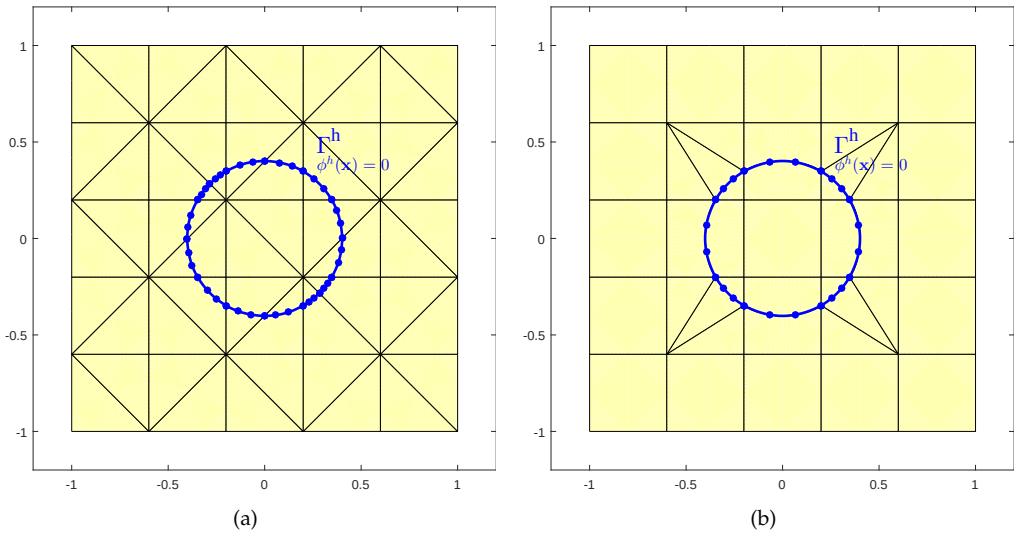


Figure 72: Background meshes with $m_\Omega = 3$ for a circular zero level set ϕ here approximated by interface elements: (a) a structured triangular background mesh, (b) Cartesian background mesh.

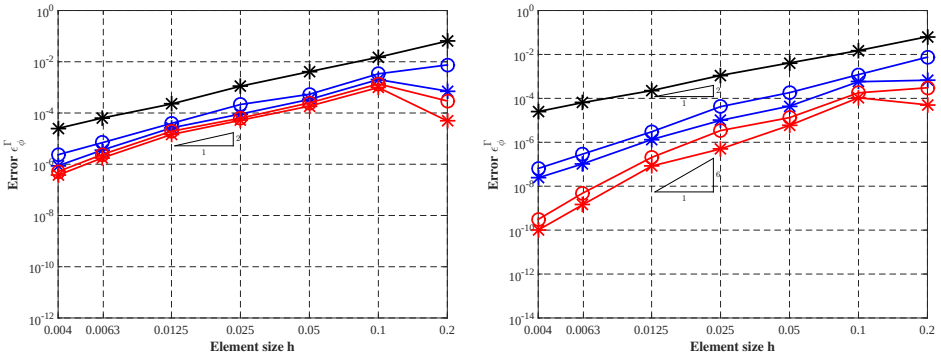
decomposition. The element order ranges from linear elements to elements of 5^{th} order. For very coarse meshes, one may observe Runge's phenomenon for very high orders, e.g. $p \geq 5$. The following tailored error norm is used for the analysis

$$\varepsilon_\phi^\Gamma = \sqrt{\int_\Gamma \phi(\boldsymbol{x}_\Gamma)^2 \, \mathrm{d}\Gamma}. \tag{7.2}$$

The error is studied at integration points which are distributed *inside* the reconstructed interface elements. Here, $\boldsymbol{x}_\Gamma$ are the integration points in the reconstructed interface and $\phi(\boldsymbol{x}_\Gamma)$ is the *analytical* level set function evaluated at $\boldsymbol{x}_\Gamma$. The level set function is evaluated at the nodes of the background mesh and interpolated in the integration points of the interface elements using standard finite element functions. It is noted that the Newton-Raphson procedure from Section 4 ensures that the element *nodes* are "exactly" on the zero level set of $\phi^h(\boldsymbol{x})$ (with a tolerance of $\approx 10^{-12}$). For the implied interface element *in-between*, and consequently also in the integration points $\boldsymbol{x}_\Gamma$, a slight deviation from the zero level set of ϕ^h occurs. The error norm in Eq. (7.2) accounts therefore for the error in ϕ^h compared to ϕ *and* for the error in the interface elements in approximating the zero level set of ϕ^h.

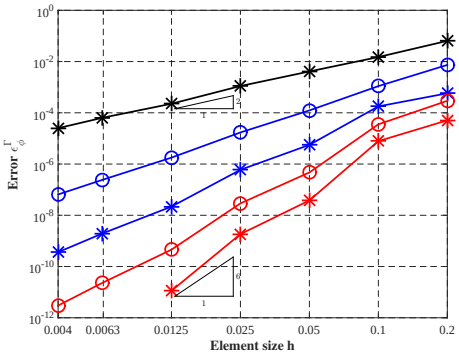
For the error analysis of the reconstruction we use a *fixed* number of integration points in the interface elements, independently of the order m_Ω of the background mesh. In detail, 21 integration points per reconstructed 1D line element are used. For all convergence diagrams it is understood that every curve represents a particular element order on a background mesh that is refined by subdividing successively the coarsest mesh configuration.

Convergence results are depicted in Fig. 73 for the structured triangular background mesh visualized in Fig. 72(a). It is seen that the search paths 1 and 2 both perform only sub-optimal, which is also true for quadrilateral background meshes. Both search variants give a rather nonuniform distribution of interface element nodes in case of awkwardly cut elements, see Sec-

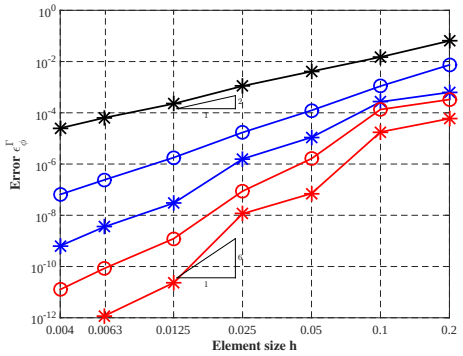


(a) Variant 1

(b) Variant 2



(c) Variant 3



(d) Variant 4

Figure 73: Quality of the reconstruction of a circular interface on a structured triangular background mesh using the error measure ε_ϕ^Γ, as defined in Eq. (7.2). Figures (a)-(d) correspond in each case to search variants 1-4.

tion 4. This, however, has severe consequences when approximating the zero level set. The search variants 3 and 4, on the other hand, perform both optimal, with a slightly smaller absolute error for variant 3. Similar behaviour was also observed for more complicated level set functions, that is, variants 1 and 2 with sub-optimal results, 3 and 4 with optimal convergence properties and additionaly variant 3 with slightly lower absolute error compared to variant 4. Due to this fact, for all forthcoming examples, only search variant 3 will be used for interface reconstructions.

The next task is to investigate the influence of different types of background meshes on the accuracy. The background meshes differ for one in the element type, that is, triangular elements (Fig. 72(a)) and quadrilateral elements (Fig. 72(b)) are investigated on regular Cartesian grids, with the global coordinate system $\boldsymbol{x}$ aligning with the element edges. In addition, deformed triangular background meshes and manipulated background meshes are investigated, as shown in Figs. 74(a) and Fig. 74(b), respectively. The deformed mesh is constructed by applying an isoparametric transformation (a cubic map) to an undeformed grid. By manipulated we mean that some "nodal movements" according to Section 6 are applied, namely, element nodes close to the zero level set are moved *away* from the interface.

Except for the deformed triangular mesh, depicted in Fig. 74(c), all decomposed meshes are fully functional without applying special techniques as local subdivision or recursive schemes. However, the mesh in Fig. 74(c), is cut twice on some element edges, here the local subdivision

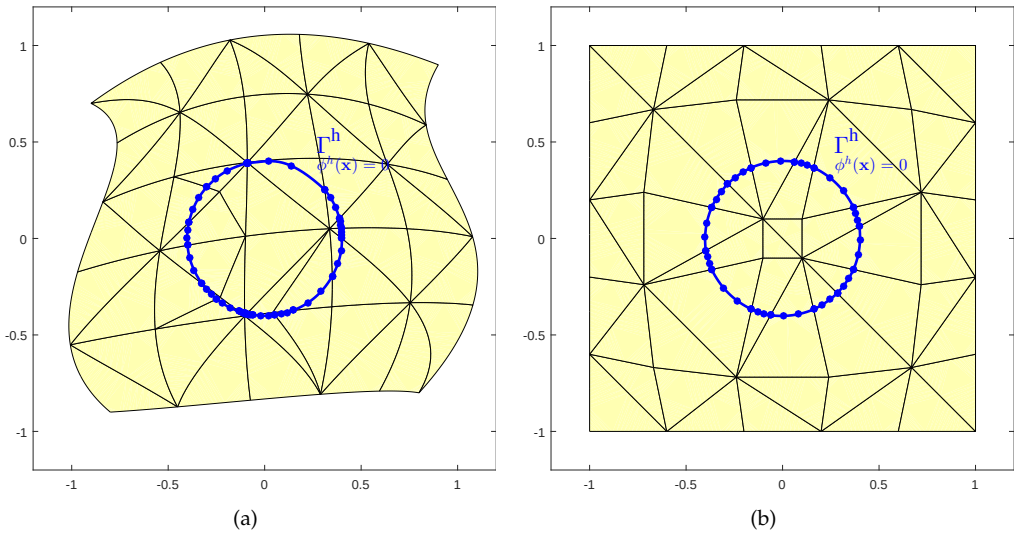


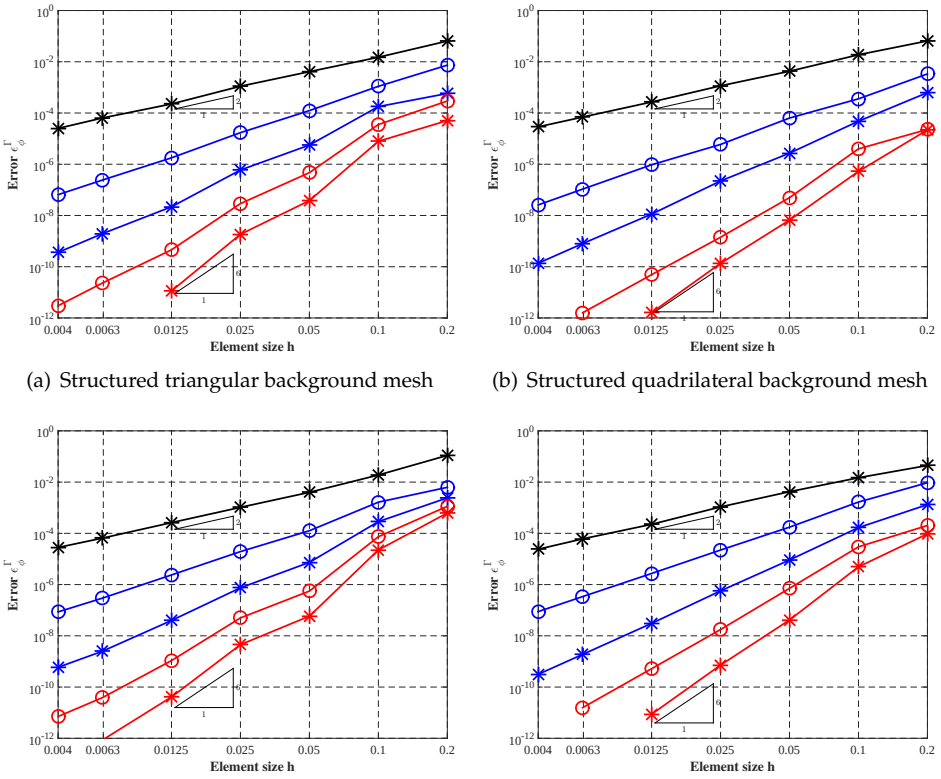
Figure 74: Triangular background meshes with $m_\Omega = 3$ for a circular zero level set ϕ here approximated by interface elements: (a) a deformed background mesh, which is constructed by applying a polynomial (cubic) mapping to a structured mesh, and (b) a structured background mesh with node movements away from the interface.

was applied. As visible from the figures, all resulting meshes are actually of a *mixed type*, meaning in general triangular *and* quadrilateral elements occur. We emphasize that if the resulting meshes are used only for integration purposes, e.g. to integrate the weak form along the interface for the Nitsche-XFEM, cut sub-elements can be refined locally resulting in meshes with hanging nodes. For the circular zero level set, however, a recursive refinement was not necessary and the resulting meshes can be used directly for simulation purposes.

Integrated error in the level set function for different background meshes

The reconstruction quality is studied using two different types of interface meshes, both evaluated using the error norm as defined in Eq. (7.2). The first study uses the interface element nodes from the reconstruction algorithm. As mentioned before, this norm comprises two different types of errors, the error from the numerical approximation of the nodal points, as well as the interpolation error for points between the nodes. To get rid of the error of approximating the nodes, we manually place the nodes *exactly* onto the zero level set along the corresponding search path. Then, the location of the integration points is found by interpolation using standard finite element functions. This error measure basically captures exclusively the "interpolation" error. However, we use this rather artificial error measure only to confirm that Eq. (7.2) is fully sufficient and the difference between ϕ and ϕ^h can be neglected. Furthermore, Eq. (7.2) seems more natural and comprehends the approximate location of the interface element nodes.

In the results visualized in Figs. 75(a)-(d), the precision of the interface mesh location based on structured meshes for both triangular and quadrilateral as well as on the deformed background mesh, is optimal after the automatic remeshing. Note that for the meshes in Figs. 75(a)-(c) no node moving was applied and therefore some of the elements look quite awkward. Nevertheless, all elements resulting from the automatic decomposition procedure are topologically valid and



(a) Structured triangular background mesh

(b) Structured quadrilateral background mesh

(c) Deformed triangular background mesh

(d) Triangular mesh with moved nodes

Figure 75: Convergence results for the reconstruction of a circular interface on different background meshes using the error measure ε_ϕ^Γ, as defined in Eq. (7.2). The interface element nodes are found by an iterative algorithm.

provide optimal convergence properties.

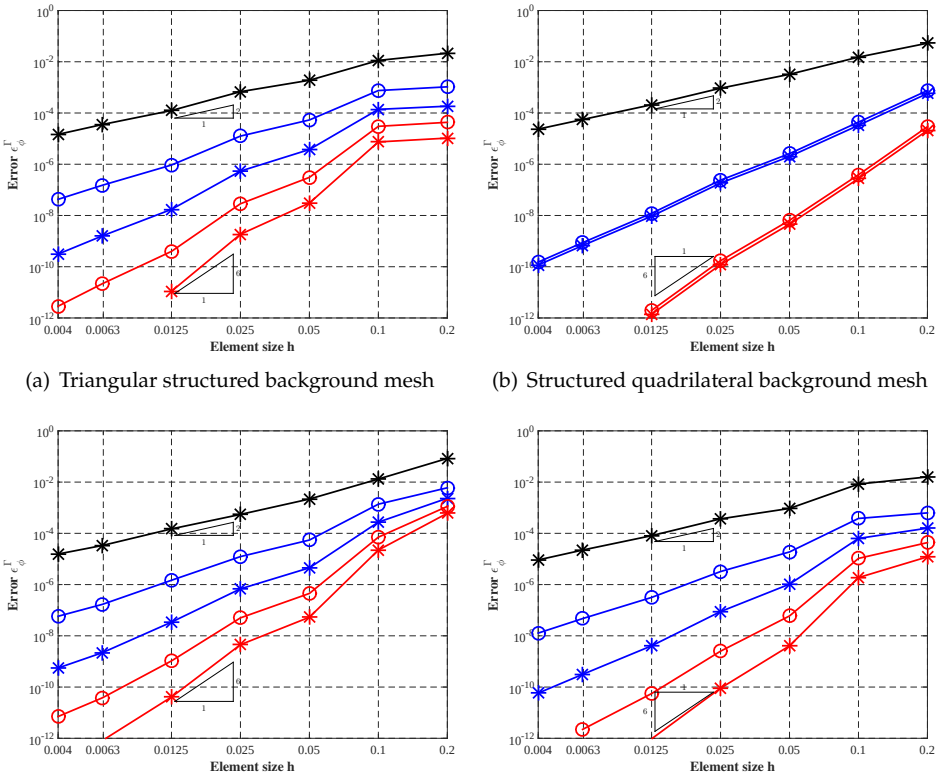
In the results in Figs. 76(a)-(d) we use the *exact* location of the interface element nodes on the zero level set. Except for the quadrilateral meshes in Fig. 76 all results look very much similar to the meshes with iteratively found nodes. It is noted that in this case the quadrilateral meshes for the orders 2 and 4 show superconvergent behaviour, meaning the convergence rates are higher than generally expected. It is confirmed that the utilization of nodes which are located exactly on the zero level set does, in general, not provide any significant advantage.

Integrated distance error for different background meshes

We mention that a number of other error measures, as e.g. using the distance norm given as

$$\varepsilon_r^\Gamma = \sqrt{\int_\Gamma (r^h(\boldsymbol{x}_\Gamma) - r)^2 \, \mathrm{d}\Gamma}. \tag{7.3}$$

were investigated, again with approximated and exact nodal positions. However, all error norms yield qualitatively similar optimal convergence rates as in Figs. 75(a)-(d) with superconvergent behaviour for quadrilateral meshes in case of exact nodal positions. Therefore for the remainder of the reconstruction sections we focus exclusively on the error norm defined by the zero level



(a) Triangular structured background mesh

(b) Structured quadrilateral background mesh

(c) Deformed triangular background mesh

(d) Triangular mesh with moved nodes

Figure 76: Convergence results for the reconstruction of a circular interface on different background meshes using the error measure ε_ϕ^Γ, as defined in Eq. (7.2). The interface element nodes are set manually to lie *exactly* on the zero level set.

set in Eq. (7.2).

7.1.2 Numerical results in two dimensions - reconstruction of a flower-shaped inclusion

Given a reasonably well graded background mesh, this example shows that using the proposed reconstruction scheme fairly complicated domains can be automatically remeshed. The example is inspired by [116, 113], with the analytical level set function (which could e.g. describe the boundary of an inclusion) given as

$$\phi(\boldsymbol{x}, \theta) = ||\boldsymbol{x}|| - (r + \alpha \sin(\omega\theta)) \tag{7.4}$$

where $||\boldsymbol{x}||$ is the location of the considered point in Ω^h, $\theta \in [0, 2\pi]$ is the angle connecting the origin and the current point $\boldsymbol{x}$, and $\{r, \alpha, \omega\}$ a set of variables determining the shape of the inclusion. The variables represent the size of the flower shape (r) , the number of blossoms (ω) and the sharpness or the curvature of the flower, given by the variable α. For the considered example these variables are chosen as $\{r = 0.5, \alpha = 0.1, \omega = 6\}$. The complexity of this example is the relatively high curvature of the zero level set. Note that with an adequate resolution of the background mesh, or an adaptive subdivision in the cut elements containing a high curvature of the

zero level set, this difficulty can always be resolved. However, here we show the performance of the method for relatively coarse background meshes.

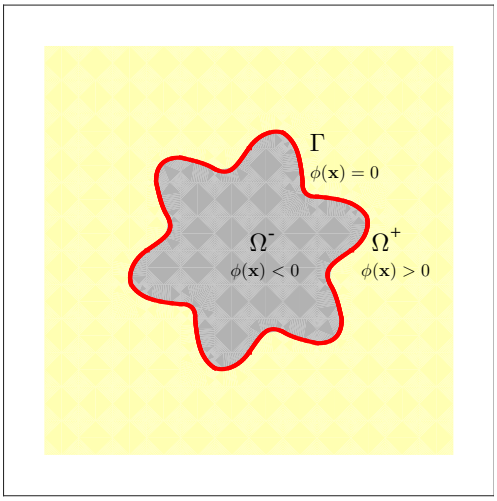


Figure 77: Two domains, Ω^+ and Ω^-, separated by a flower-shaped zero level set ϕ, here denoted with Γ. The signs of the two domains correspond with the signs of the level set function ϕ within the domain

In order to study the quality of the interface reconstruction the first three background meshes are identical with the background meshes used for the first example, see Fig. 74, including a structured triangular mesh in (a), a quadrilateral mesh in (b) and deformed background mesh in (c). The mesh with node movements (d) must obviously be adjusted to a different zero level set. The analysis is again carried out using a series of embedded meshes, indicated by the characteristic element length h. The reconstructed interfaces are visualized in Figs. 78(a)-(d), including the interface element nodes.

For some of the meshes, in particular the structured triangular mesh, a valid interface mesh results, but the subsequent element decomposition for the domain is invalid. This means that the Jacobian of the cut elements becomes negative in some points. This can be explained with the high curvature of the zero level set, and can be controlled by the search direction and the use of alternative elements as e.g. a Chen-Babuška node distribution, [70]. An alternative would be to adaptively refine the elements close to the high curvature of the zero level set. Another way to deal with this problem, especially in the context of pure integration tasks is to recursively refine some of the elements until a user defined error tolerance is reached. This would result in a non-conforming mesh with hanging nodes, but these nodes would not appear explicitly in the system of equations.

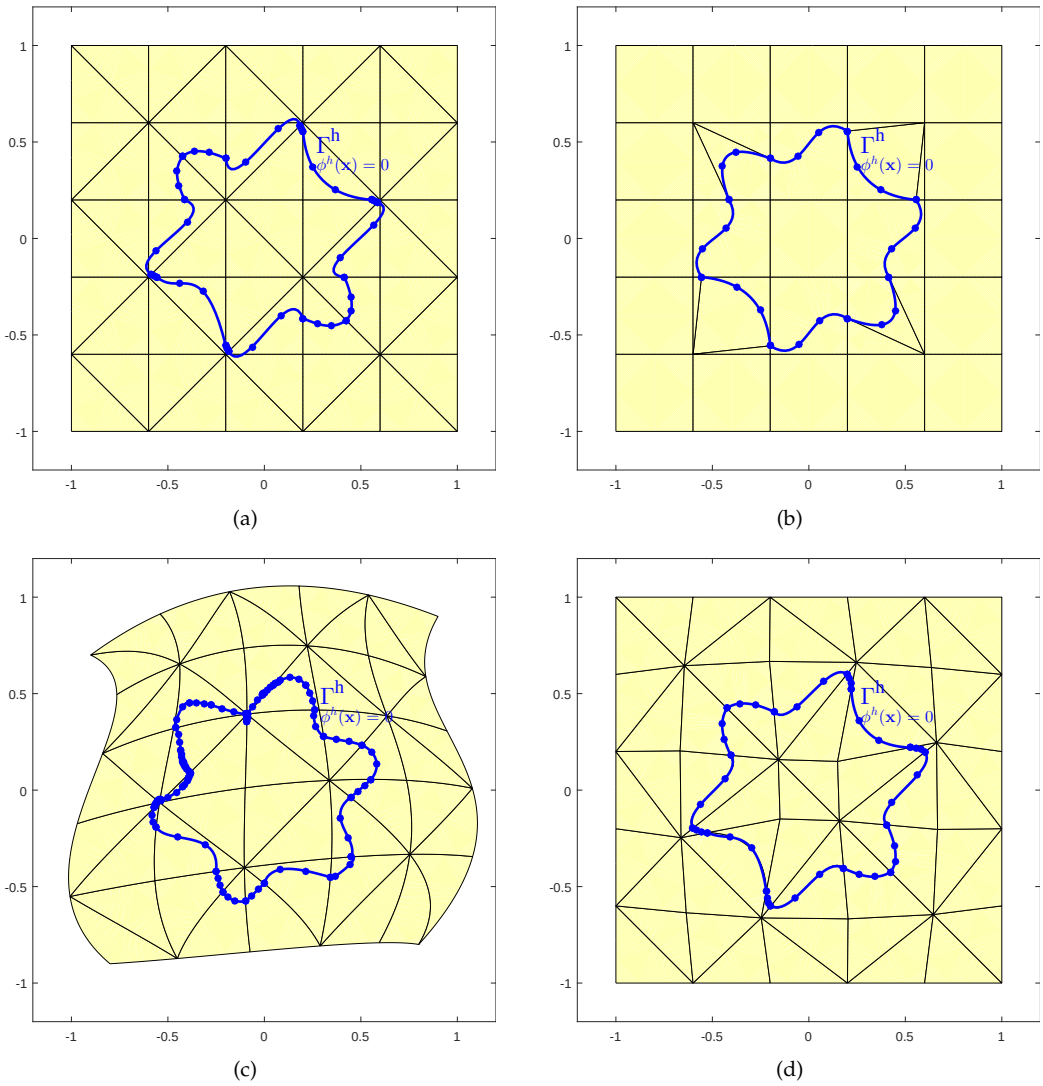
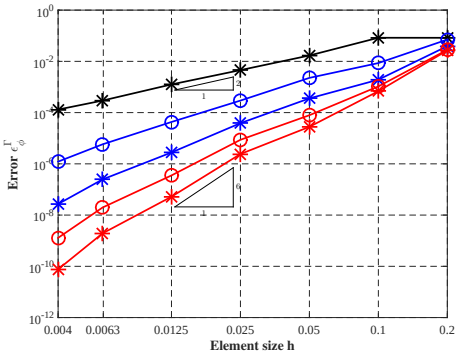


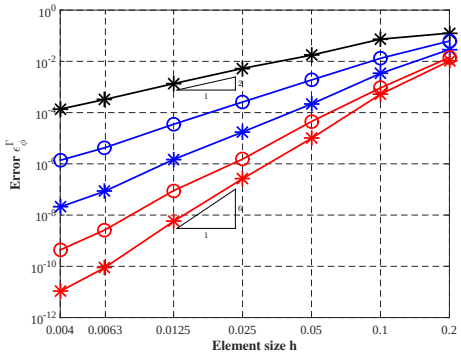
Figure 78: Background meshes with the applied decomposition scheme for $m_\Omega = 3$ for a flower-shaped zero level set ϕ: (a) a triangular background mesh, (b) regular Cartesian background mesh, (c) deformed background mesh constructed using a polynomial (cubic) mapping, and (d) background mesh with node movements.

As shown in Fig. 78(a)-(d) the proposed method performs reasonably good even for coarse background meshes and a zero level set with a high curvature. For the structured triangular background mesh in Fig. 78(a), the convergence rates for the coarser meshes are not quite optimal, but they tend asymptotically to optimal rates. For the structured quadrilateral as well as the triangular mesh with node movements the situation is basically similar. For the deformed background mesh, however, it was necessary to recursively refine the background mesh. The emerging sub-elements, including the hanging nodes, are shown in Fig. 78(c). As mentioned, this mesh can not be used directly for simulation purposes. However, it is worth noting, that the deformed background mesh in combination with the high curvature of the zero level set is a rather artificially created topological situation. All other meshes resulting from the subdivision

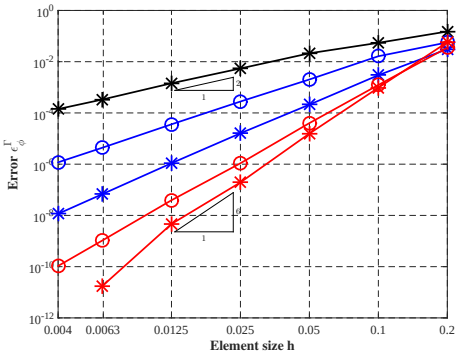
procedure are topologically valid and can immediately be used for simulation purposes.



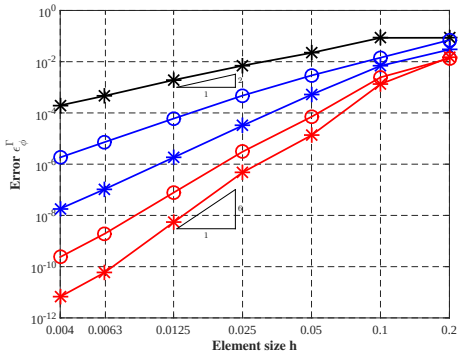
(a) Triangular structured background mesh



(b) Structured quadrilateral background mesh



(c) Deformed triangular background mesh



(d) Triangular mesh with moved nodes

Figure 79: Convergence results of the interface reconstruction of a flower-shaped interface using the error measure ε_ϕ^Γ, as defined in Eq. (7.2). Figures (a)-(d) correspond to different background meshes.

7.1.3 Numerical results in three dimensions - reconstruction of a spherical interface

This example is the three dimensional analogy to the circle and represents the easiest curved three-dimensional geometry. The level set function is given as

$$\phi(\boldsymbol{x}) = ||\boldsymbol{x}|| - r \tag{7.5}$$

with $||\boldsymbol{x}|| = \sqrt{x^2 + y^2 + z^2}$ and r as the radius of the sphere chosen with $r = 0.401$. We proceed further with two types of background meshes, a structured and a deformed hexahedral mesh, see Fig. 81(a)-(b). The deformed mesh in Fig. 81(b) results from a smooth isoparametric map from a reference hexahedral element to the physical domain. For all three-dimensional problems we will use tetrahedral meshes which are constructed utilizing a mapping of five tetrahedral elements into one (hexahedral) cell of a Cartesian grid. Additionaly, for all meshes some node moving procedures are applied in order to improve the stability. The spherical zero level set function from Eq. (7.5) is depicted in Fig. 80. The external domain boundary is given as $[-1, 1]^3$.

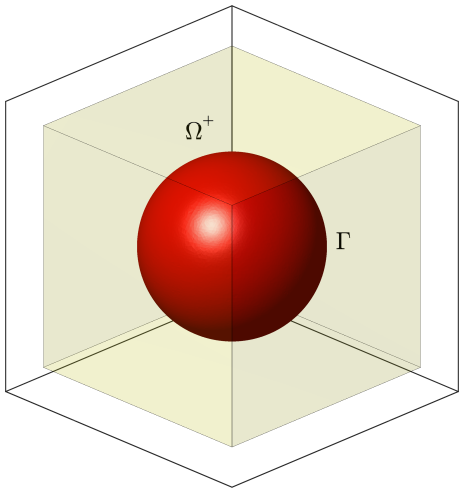
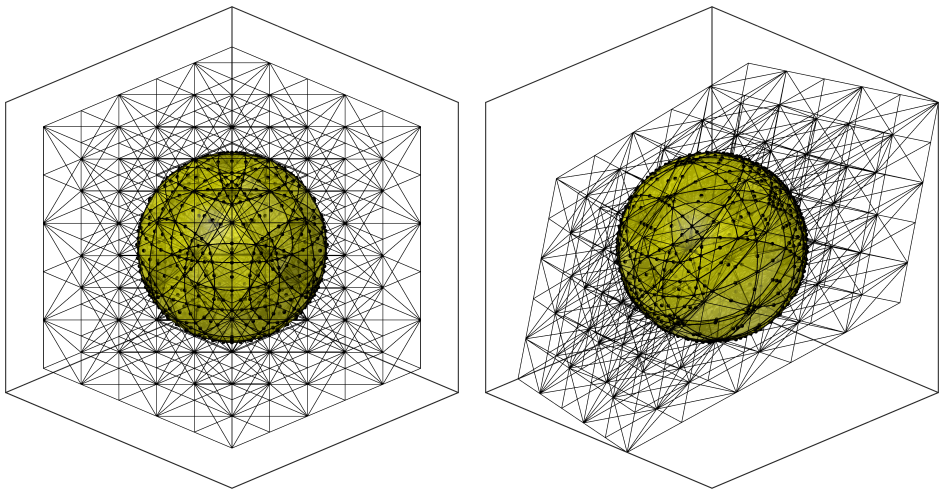


Figure 80: (a) Two domains, Ω^+ and Ω^-, separated by a spherical zero level set ϕ, here denoted with Γ. The signs of the domains correspond with the signs of the level set function within the domain.

The sequence of background meshes is again nested, the number of hexahedral elements per side is $\{5, 10, 20, 40, 80\}$; the element order ranges from tri-linear elements to elements of 5^{th} order.

The same error measures and the same course of action are taken here as for the two dimensional case, that is, the absolute value of the analytical level set function is evaluated in these points and a fixed number of integration points are distributed inside the reconstructed interface elements, 12 integration points in reconstructed triangles and 16 integration points in reconstructed quadrilaterals. We present now the results form different error norms and vary some of the parameters of the reconstruction procedure.



(a) Structured tetrahedral background mesh (b) Deformed tetrahedral background mesh

Figure 81: (a) Two domains, Ω^+ and Ω^-, separated by the discretized zero level set ϕ^h on a undeformed quadratic tetrahedral background mesh. In (b) the zero level set is discretized using a deformed quadratic tetrahedral background mesh.

Integrated error in the level set function for different search directions

Convergence results are now presented in style of the two-dimensional investigations for the proposed four different variants of the search directions of the nodes on the zero level set, see Section 4.10.5. We show only the results for the undeformed hexahedral mesh, the deformed tetrahedral mesh yields very similar results. The main difference between reconstruction studies in 2D and 3D is the fact that in three dimensions the search paths of the face reconstructions represent another variable that can be chosen independent of the search path inside the element. For all our studies, in order to keep the number of varied parameters low, we choose the normal to the linear reconstruction of the zero level set.

The results are shown in Fig. 82, in which every diagram represents one search variant. Variant 1, Fig. 82(a), is the search path which is independent on the level set data and instead depends only on the current topology. It is obvious that the convergence rates for this search direction are by far not optimal, this property is shared with the two-dimensional analogy, see Fig. 73(a). Variant 2 uses a linear reconstruction of the zero level function as a starting point and yields similar sub-optimal convergence rates. Variants 3 and 4 use the normal vector to a quadratic Serendipity reconstruction of the zero level set and the gradient to the approximated level set function, respectively, as search direction. In both cases the results indicate that no optimal higher-order discretization is obtained. Our preferred search direction for further investigations will be variant 5, which utilizes a reconstruction of surface elements normal to the linear reconstruction and the search path of inner points uses the normal vector to the Šolin-reconstruction of the zero level set. As shown in see Fig. 82(e), the results are optimal at least up to the order $p = 5$.

Basically, the results show that the insights of the two-dimensional investigations are also applicable to three dimensions. It is again confirmed that the matter of suitable search directions is of critical importance for the accuracy of the reconstruction.

Integrated distance error error for different background meshes

Repeating the two-dimensional study for the three-dimensional case and measuring the error ε_r^Γ confirms that especially for the reconstruction studies the type of norm is basically interchangeable. Furthermore, it shows once more that the accuracy of the reconstruction is practically independent whether the background mesh is deformed or undeformed. See Fig. 83(a)-(b) for the integrated distance error ε_r^Γ on the undeformed and the deformed tetrahedral mesh, respectively. Optimal convergence rates are achieved for both types of background meshes. Notice that for the deformed mesh, for all investigated orders $p > 1$, the reconstruction fails for the coarsest meshes. However, once the mesh is refined, optimal rates are obtained.

7.1.4 Numerical results in three dimensions - Goursat's surface

The last reconstruction example is the three-dimensional generalization of the flower shape, and is in fact a particular case of Goursat's surfaces, see [108]. It represents the topologically most challenging zero level set as there are multiple changes in curvature over the domain and furthermore there are some distinguished points with locally high curvature, see Fig. 84(a). The level set function is defined as

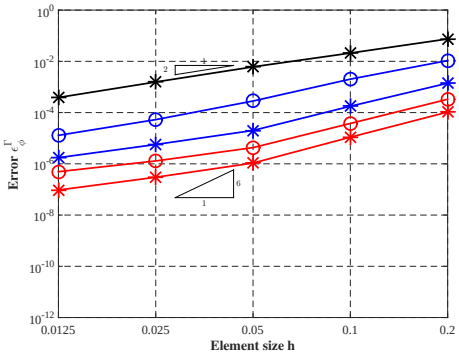
$$\phi(\boldsymbol{x}, \theta) = ||\boldsymbol{x}|| - r + \alpha \cdot (\cos(\omega \cdot x) + \cos(\omega \cdot y) + \cos(\omega \cdot z)) \tag{7.6}$$

with the Euclidian distance $||\boldsymbol{x}||$, the radius r is chosen with $r = 0.5$ and the other factors are chosen with $\alpha = 0.1$ and $\omega = \pi$. The external domain boundary is given as $[-1, 1]^3$.

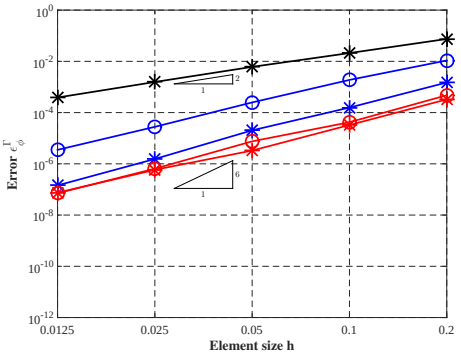
For this particular level set function we choose only the undeformed background mesh, as visualized in Fig. 84(b), because the reconstruction on coarser meshes fails due to the combination of the curvature on the background mesh and the level set function. The results in Fig. 85 indicate optimal convergence properties.

7.1.5 Summary

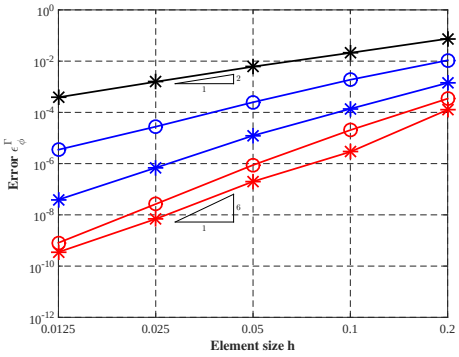
The conclusion from the reconstruction section is that for two-dimensional as well as three-dimensional problems the search paths significantly influence the accuracy of the interface approximation. In two dimensions the search variants 3 and 4 achieve optimal convergence rates in different error norms. Search paths 1 and 2 give clearly sub-optimal results. The numerical studies show that the influence of the element type, or the structure of the background mesh is in most cases negligible. Reasonably deformed background meshes can be handled just as good as structured, triangular meshes perform in terms of convergence rates as good as quadrilateral meshes. An important topic, especially for practical applications, is the curvature of the zero level set. For large curvatures, the reconstruction (or the subsequent decomposition) is likely to fail in coarse elements, wherefore a recursive refinement scheme must be implemented as well. Upon refinement the curvature relative to the element size is decreased until the reconstruction is successful. In three dimensions, the important difference is that an additional variable appears, which is the type of reconstruction for the two-dimensional faces. However, the results for the two-dimensional search paths are more or less transferable to the three-dimensional case. The variants 3 and 4 converge significantly better than 1 and 2. It is important to mention that the Šolin-mapping needs to be applied to the face reconstructions in order to obtain optimal convergence rates.



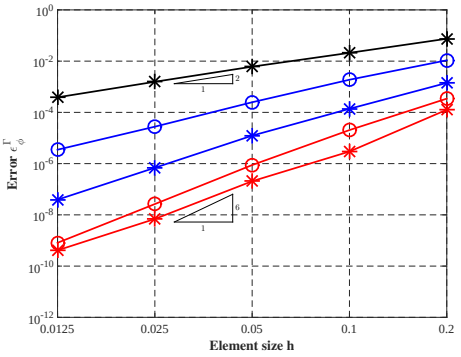
(a) Search variant 1



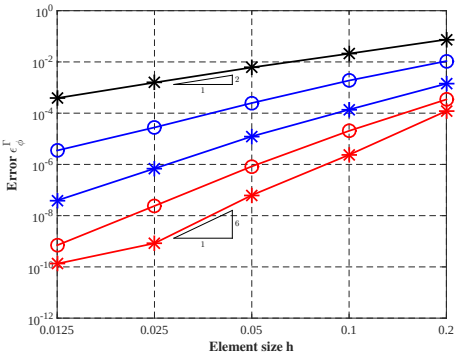
(b) Search variant 2



(c) Search variant 3

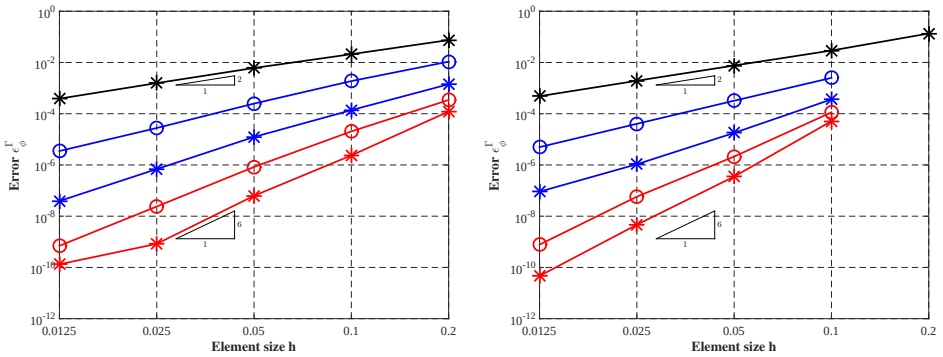


(d) Search variant 4



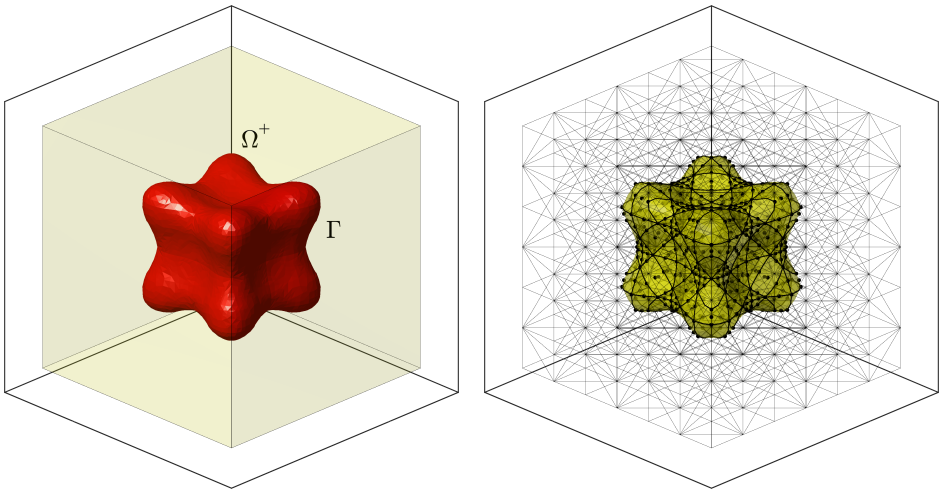
(e) Search variant 5

Figure 82: Convergence results in 3D on implicitly defined spherical zero level sets: an undeformed hexahedral background mesh is used and the five diagrams correspond with the different variants of the search directions. In all cases the error norm ε_ϕ^Γ from Eq. (7.2) is used.



(a) Search variant 1 (b) Search variant 2

Figure 83: Convergence results in 3D on implicitly defined spherical zero level sets: an undeformed hexahedral background mesh in (a) and a deformed background mesh in (b) are used in combination with the error norm ε_r^Γ as defined in Eq. (7.3).



(a) (b)

Figure 84: Two domains, Ω^+ and Ω^-, separated by the zero level set of ϕ, here denoted with Γ. Notice that the domain Ω^+ is unbounded, that is, it extends to infinity.

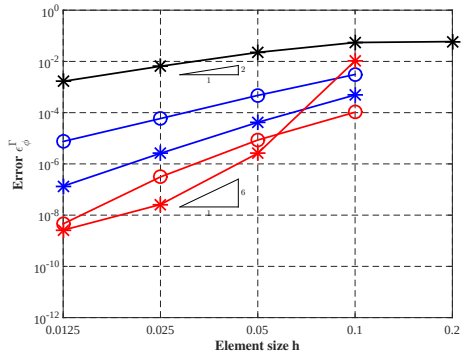


Figure 85: Convergence results in 3D on an implicitly defined Goursat's surface: the reconstruction is conducted on an undeformed hexahedral background mesh. The quality of the reconstruction is measured in the error norm ε_ϕ^Γ as defined in Eq. (7.2).

7.2 Integration of functions and their interpolation in/on zero level sets

The integration on implicit zero level sets in 2D and 3D is a simple task, provided that a reconstructed interface mesh is available. This task is also called "surface integration" herein, although, in two dimensions, it is rather an integration on the zero-iso*line*. For the numerical integration standard Gauß-Legendre points are used herein, see Section 5. As usual, each integration point for standard domain integration is defined by its position and the corresponding weight, i.e. tuples $(\boldsymbol{a}_i, w_i^{\boldsymbol{a}}) \in \mathbb{R}^d$, note however, that $\boldsymbol{a}_i \in \mathbb{R}^{d-1}$ for the integration *on* zero level sets. These integration points may be mapped by means of a standard isoparametric mapping as in Eq. (5.2) to the reference background element in coordinates $\boldsymbol{r}$ and then further to the physical background element in coordinates $\boldsymbol{x}$. Another possibility is to map the *element nodes* $\boldsymbol{r}_i^\Gamma$ of the reconstructed elements via an isoparametric mapping to obtain $\boldsymbol{x}_i^\Gamma$ and then map the integration points directly from $\boldsymbol{a}$ to the global coordinates $\boldsymbol{x}$.

The essential issue is that the integration weights are defined in the reference space and therefore have to be modified during the mappings. In standard Gaussian quadrature this task is done using the Jacobian, for this type of integration the so-called *Gramian* or *Gram's determinant* is used, i.e.

$$w_i^{\boldsymbol{r}} = \sqrt{\det(\mathbf{J}_{\boldsymbol{r}}^{\mathrm{T}} \cdot \mathbf{J}_{\boldsymbol{r}})} \cdot w_i^{\boldsymbol{a}},$$

where $\mathbf{J}_{\boldsymbol{r}}$ is the Jacobi matrix of the mapping $\boldsymbol{r}\,(\boldsymbol{a})$. Note that for the integration on zero level sets, $\boldsymbol{a}_i \in \mathbb{R}^{d-1}$ but $\boldsymbol{r}_i, \boldsymbol{x}_i \in \mathbb{R}^d$, so that the Jacobi-matrix is rectangular. In contrast, when integrating implicitly defined volumes also $\boldsymbol{a}_i \in \mathbb{R}^d$, and the Jacobi matrix is a square matrix. See also Fig. 55 for the difference of the dimension of the reference elements used for surface and volume integration.

7.2.1 Numerical results in two dimensions

Convergence studies similar to the ones in the reconstruction part are conducted on higher-order background meshes of different orders. Both, triangular and quadrilateral elements are investigated, each in a Cartesian arrangement where the element edges align with the coordinate system $\boldsymbol{x}$ ("undeformed mesh") and in a deformed mesh (where the elements are perturbed by a cubic mapping). The level set function Eq. (4.7) is used with the same specifications as before. See Fig. 72 and Fig. 74 for example meshes with $m_\Omega = 3$ and 5×5 quadrilateral elements and double the number of triangular elements. In the convergence studies, $\{5, 10, 20, 40, 160, 250\}$ elements are used per dimension. As before, we use a fixed number of integration points per reconstructed interface element no matter what the order of the elements $m_\Omega = m_\Gamma$ is.

7.2.2 Integrating functions *on* the zero level set

Next, we define another error norm with the aim to study the accuracy when integrating a function $f\,(\boldsymbol{x})$ *on* the zero isoline:

$$\varepsilon_\Gamma = \frac{\left|I^h - I_{\mathrm{ex}}\right|}{|I_{\mathrm{ex}}|} \quad \text{with} \quad I^h = \int_\Gamma f\left(\boldsymbol{x}^h\right) \,\mathrm{d}\Gamma,\; I_{\mathrm{ex}} = \int_\Gamma f\left(\boldsymbol{x}_{\mathrm{ex}}\right) \,\mathrm{d}\Gamma \tag{7.7}$$

For I_{ex}, the function is evaluated exactly on the circle with radius r, i.e. at the points $\boldsymbol{x}_{\text{ex}}$. For I^h, the integral is evaluated numerically using the integration points $\boldsymbol{x}_i^h$ inside the reconstructed line elements. At $\boldsymbol{x}_i^h$, the function is evaluated exactly so this is *not* an interpolation task. The function is defined as

$$f(\boldsymbol{x}) = 1/2 \cdot x + 1/4 \cdot y + x^2 + 2 \cdot y^3 \tag{7.8}$$

which yields $I_{\text{ex}} = \pi \cdot r^3$.

Convergence results are shown in Fig. 86 for the background meshes that are used in the reconstruction section. Variant 3 is chosen for the search directions. It can be seen that for even orders of $m_\Omega = m_\Gamma$, the convergence rate is $m_\Omega + 2$, i.e. one order higher than the expected, optimal convergence rate. This is well-known also for the (standard) integration of functions on *explicit* surface meshes. Odd orders of $m_\Omega = m_\Gamma$ converge with the order $m_\Omega + 1$. Hereby a structured triangular background mesh is used in Fig. 86(a), a Cartesian background mesh in Fig. 86(b), and Fig. 86(c),(d) is a deformed mesh as well as a mesh with node manipulation, respectively. We also note some minor wiggles in the convergence plots because we cannot expect convergence results that are as smooth as those obtained with explicit surface meshes and nested elements upon refinement. However, all meshes show the expected asymptotic convergence rates.

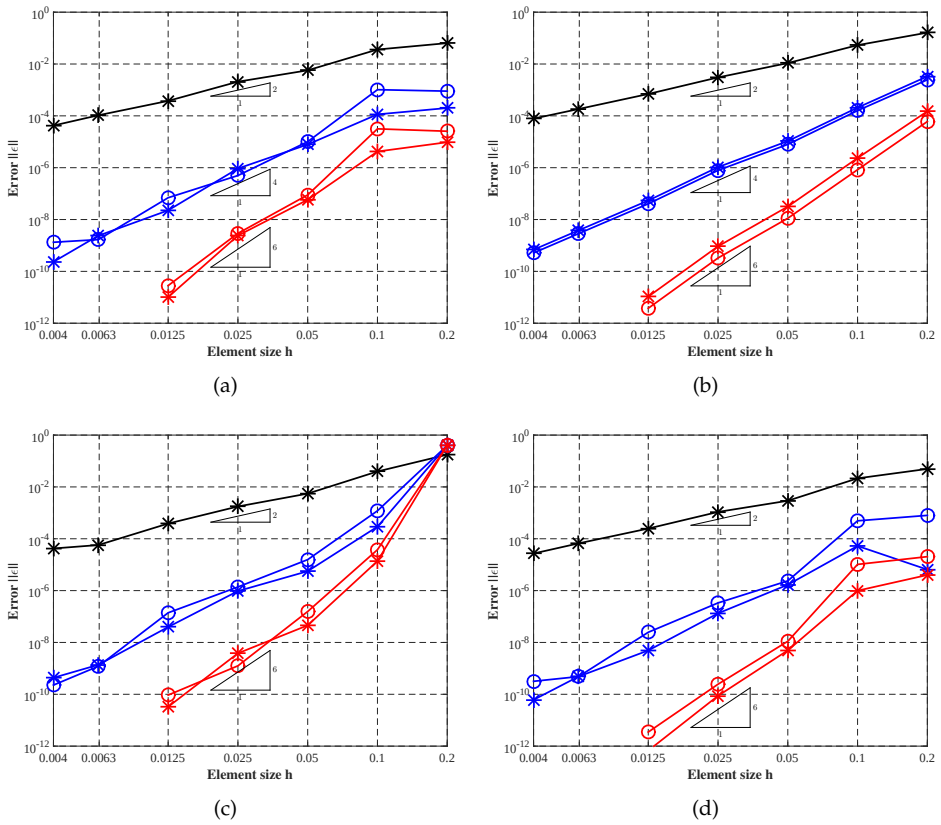


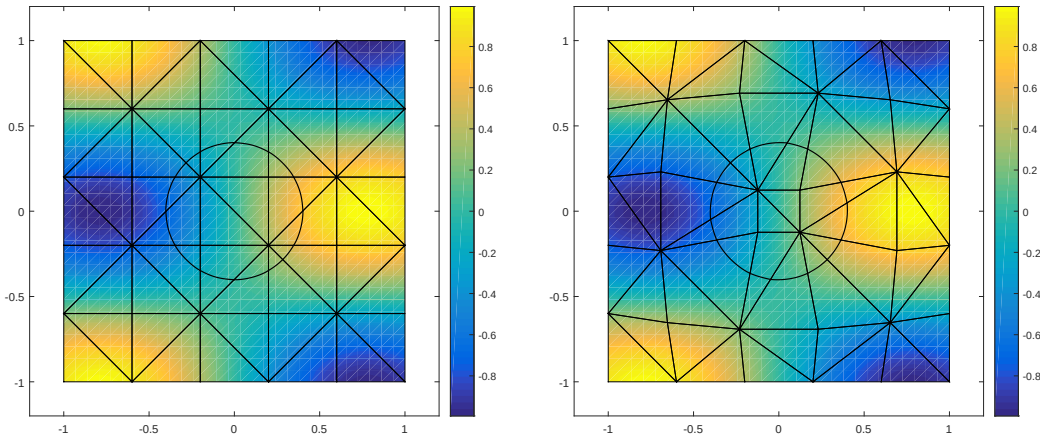
Figure 86: Convergence results in 2D for integration of smooth functions *on* implicit zero level sets: structured triangular/quadrilateral meshes in (a)-(b) and deformed background meshes and a mesh with node manipulations in (c)-(d), respectively.

7.2.3 Error for integrating functions *in* implicitly defined domains

After investigating the behaviour of functions when integrating them *on* the zero level set, it is important to investigate the convergence properties within the resulting implicitly defined domains, which are in 2D areas. The question is, therefore, whether the automatic decomposition within the of the background mesh retains optimal approximation properties of the standard L_2-projection. Also it is crucial to investigate the convergence properties of the different mappings, as introduced in Section 5. Two different mesh configurations are used that are automatically decomposed to align with a circular inclusion, similar to Fig. 71(a)-(b). All geometry specifications are the same as for the circular reconstruction example. The function to be interpolated is defined as

$$f(x, y) = \sin(2x)\cos(3y) \tag{7.9}$$

Clearly, the function $f(x, y) \in C^{\infty}(\Omega)$, that is, the function is infinitely smooth over the whole domain. In Fig. 87(a) and Fig. 87(b) show plots of the function over the deformed background mesh and over the background mesh with manipulated nodes.



(a) Structured triangular background mesh

(b) Triangular background mesh with node manipulations

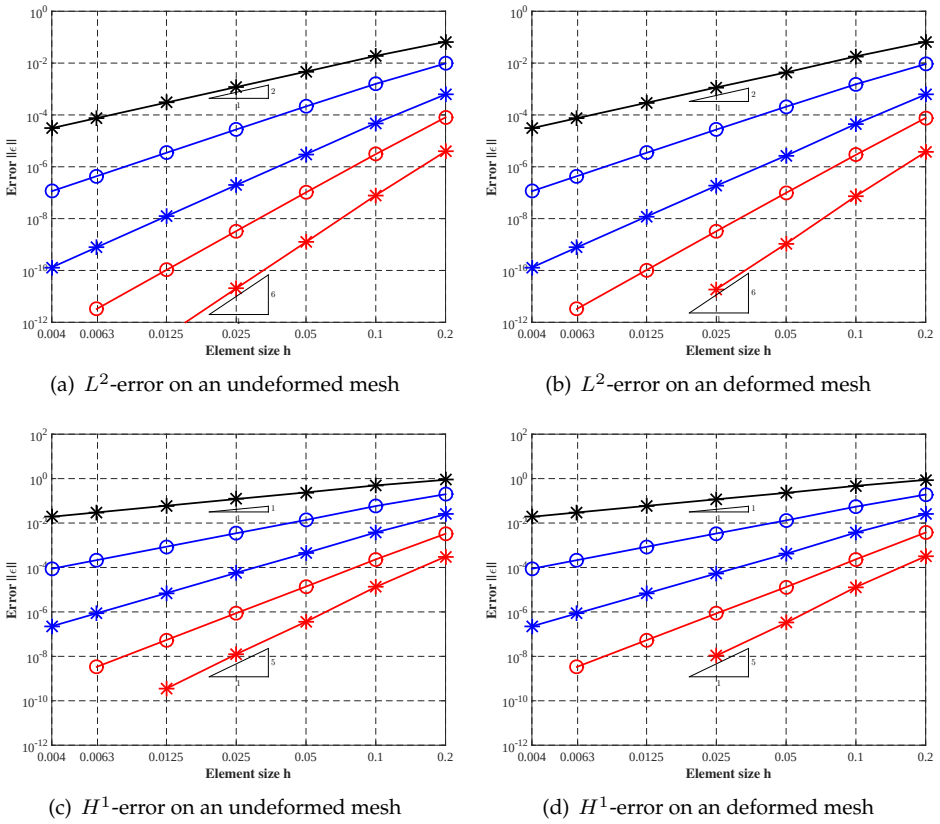
Figure 87: Plot of the function for which an L^2-projection is made on automatically decomposed background meshes. In (a) a triangular mesh and in (b) triangular mesh with node manipulations close to the zero isoline.

The error is measured in the standard L^2-norm as well as in the H^1-norm, to show that in both norms optimal convergence rates are obtained. To begin with, we consider the approximation error of f in the L^2-norm as well as the *gradient* approximation error measured in the H^1-space, defined by

$$\begin{aligned} ||\varepsilon||_{L^2} &= \sqrt{\int_{\Omega} \frac{(f(\boldsymbol{x}) - f^h(\boldsymbol{x}))^2}{f(\boldsymbol{x})^2} \, \mathrm{d}\Omega} \\ ||\varepsilon||_{H^1} &= \sqrt{\int_{\Omega} \frac{(\nabla f(\boldsymbol{x}) - \nabla f^h(\boldsymbol{x}))^2}{\nabla f(\boldsymbol{x})^2} \, \mathrm{d}\Omega}. \end{aligned}$$

The results in Fig. 88(a)-(d) indicate that the remeshing strategy, indeed, achieves asymptotically

optimal convergence properties, in both the L^2-norm and the H^1-norm. We show only the results for the undeformed as well as the mesh with node manipulations, the results for the other back qualitatively similar.



(a) L^2-error on an undeformed mesh

(b) L^2-error on an deformed mesh

(c) H^1-error on an undeformed mesh

(d) H^1-error on an deformed mesh

Figure 88: Convergence results in 2D for integration of smooth functions *in* decomposed meshes given by zero level sets. The interpolation quality is measured in the L^2-norm and in the H^1-norm: In (a),(c) for undeformed meshes and in (b),(d) for meshes with manipulated nodes.

In Section 5 we introduced various types of coordinate mappings based on boundary information only. For all the investigations carried out so far the so-called *Šolin*-mapping was used as the default-map. As seen from the results, the map gives optimal properties for interpolation. Now, for the sake of completeness we discuss the interpolation properties of the other introduced mappings, for simplicity, only the undeformed triangular background mesh is considered. In detail, we investigate the the L_2-error and the H_1-error for the *Lenoir-mapping* and the *blending function mapping* for the interpolation problem of the function defined in Eq. (7.9). The two mappings are hereby used to map element nodes from a reference element to curved the sub-cells. As visualized in Fig. 89(a)-(c) the Lenoir-mapping gives optimal approximation properties, however, the drawback is that it is more difficult to implement than the actually applied mapping. The blending function mapping, on the contrary, clearly gives sub-optimal results and degrades the convergence rates significantly, in this example the rates are bounded to the interpolation properties of cubic elements. The problems with this mapping were mentioned in the integration section, see Section 5, and this issue can be explained with the regularity properties

of the Jacobian of the map. We emphasize that the source of the problem are the triangular elements, nevertheless, if only the blending function mapping is applied also the performance of quadrilateral background meshes would be sub-optimal. This is due to the fact that the automatic decomposition in general always gives mixed meshes, therefore the final mesh will contain triangular elements as well, except for some special zero level sets.

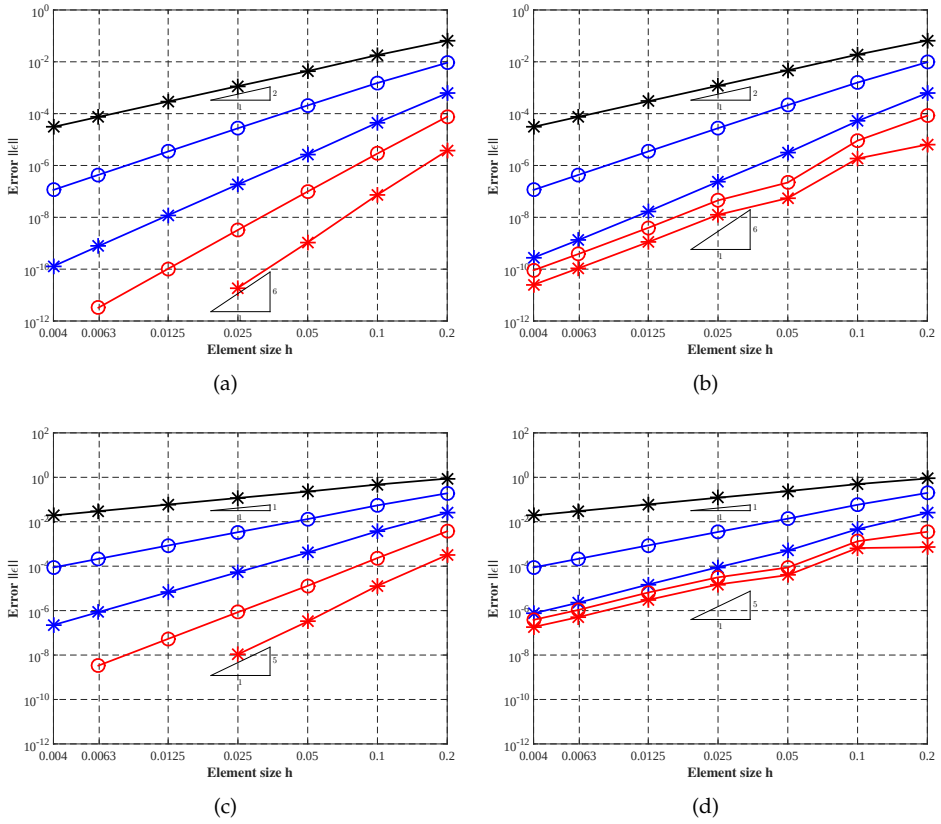


Figure 89: Convergence results in 2D for integration of smooth functions *in* decomposed meshes given by zero level sets. The interpolation quality is measured in the L^2-norm and in the H^1-norm: for the Lenoir-mapping (a),(c) and for the blending function mapping in (b),(d).

7.2.4 Numerical results in three dimensions

In three dimensions, similar convergence studies are conducted on higher-order meshes of different order m_Ω. The interpolation properties of automatically decomposed deformed and undeformed meshes composed by tetrahedral elements are investigated, the meshes are visualized in Fig. 81. The spherical level set function defined in Eq. (4.7) is used. In the convergence studies, $\{5, 10, 20, 40, 80\}$ hexahedral elements are used per dimension, therefore five times the number of tetrahedra.

The following studies are in the order and style of the above two-dimensional results.

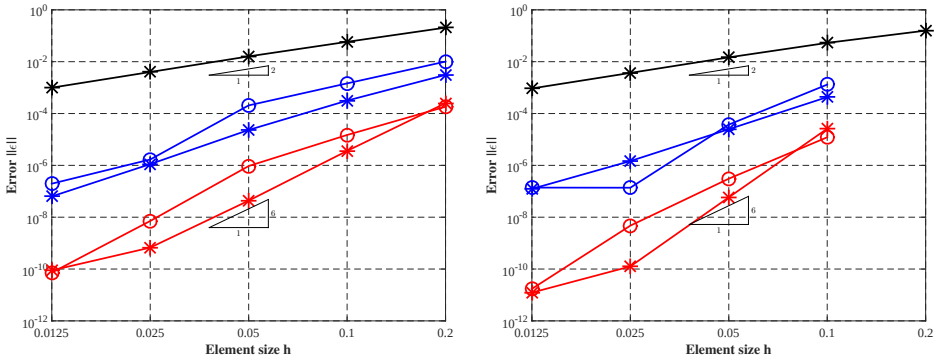
7.2.5 Error for integrating functions *on* the zero level set

The error norm in Eq. 7.7 is investigated, i.e. the task is to integrate a function $f(\boldsymbol{x})$ the sphere. We choose

$$f(\boldsymbol{x}) = x^2 \cdot y^2 + 1/2 \cdot \cos(z) \tag{7.10}$$

which, when integrated analytically on a sphere with radius r, yields $I_{\text{ex}} = 4/15 \cdot \pi \cdot r^6 + 2 \cdot \pi \cdot r \cdot \sin(r)$.

Convergence results are visualized in Fig. 90. Variant 3 is chosen for the search directions. It is seen that the convergence is again higher-order. Some minor wiggles in the convergence plots are seen, similar to the two-dimensional results.



(a) L^2-error on an undeformed mesh (b) L^2-error on an deformed mesh

Figure 90: Convergence results in 3D for integration of smooth functions *on* automatically decomposed meshes defined by an implicit spherical zero level set. The interpolation quality is measured in the ε_Γ-norm as defined in Eq. (7.7), in (a) for an undeformed mesh and in (b) for a deformed mesh.

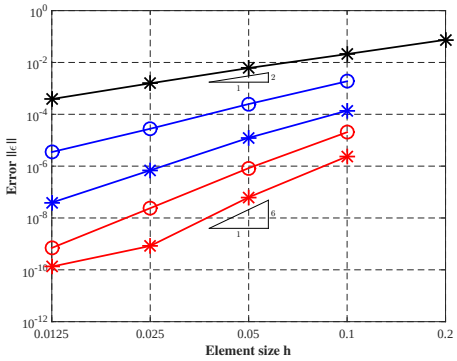
7.2.6 Error for integrating functions *inside* implicitly defined domains

Now we conduct the same investigations for integration of smooth functions in implicitly defined domains in 3D, that is, in volumes. We take the undeformed as well as the deformed tetrahedral background meshes and automatically decompose them to align with a spherical inclusion, see Fig. 81(a)-(b). The function to be interpolated is given as

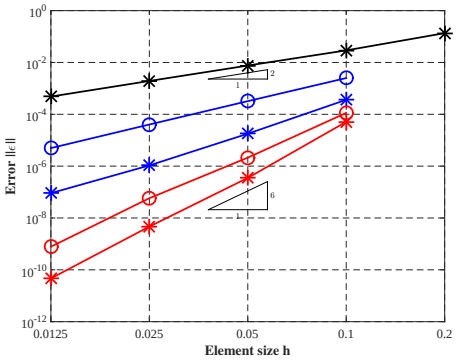
$$f(x, y, z) = \sin(2x) \cdot \cos(3y) \cdot \cos(3z)$$

and is therefore $f(\boldsymbol{x}) \in C^\infty(\Omega)$. We investigate again the error in the function in a L^2-norm and the error in the function derivatives in the H^1-norm.

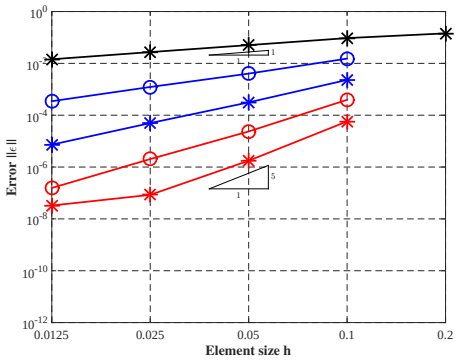
The results in Fig. 91 indicate that the remeshing strategy, indeed, achieves optimal convergence properties in both the L^2-norm and the H^1-norm. Some minor wiggles are expected, especially for the coarser meshes, and asymptotically the convergence rates show optimal behaviour.



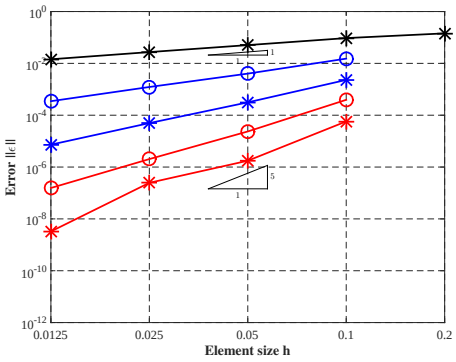
(a) L^2-error on an undeformed mesh



(b) L^2-error on an deformed mesh



(c) H^1-error on an undeformed mesh



(d) H^1-error on an deformed mesh

Figure 91: Convergence results in 3D for integration of smooth functions *inside* automatically decomposed meshes defined by an implicit spherical zero level set. The interpolation quality is measured in the L^2-norm and in the H^1-norm: In (a),(c) for undeformed meshes and in (b),(d) for deformed meshes.

7.2.7 Summary

In two dimensions, variant 3 and 4 achieve optimal convergence rates in different error norms for arbitrary orders $m_\Omega = m_\Gamma$. Variant 1 and 2 are sub-optimal, showing the important influence of the search directions on the accuracy. In three dimensions, the important difference is that the convergence rates are no longer optimal for any of the variants. However, variants 3 and 4 converge significantly better than 1 and 2. An additional reason for the sub-optimal convergence rates in three dimensions was found in the constraint that the nodes on the edges of interface elements must remain on the faces of the corresponding background elements.

7.3 Boundary value problems I - Interface problems

For the analysis of boundary value problems identical meshes and level set functions as for the reconstruction and interpolation analysis are used. The problems investigated herein belong to the class of the so-called interface problems, meaning that a weak discontinuity is present within elements. Actually, for the considered examples this is interchangeable with bi-material problems. As discussed in Section 2, if not accounted for the interface, the convergence rates will drop significantly, therefore leading the application of higher-order finite elements ad absurdum. In the remainder of this paragraph two methods are compared for typical benchmark examples for interface problems. For one, the higher-order conformal decomposition method is used, using a decomposition strategy and conformal meshes to achieve higher-order convergence rates. As a counterpart, a Nitsche-XFEM, as described in Section 3 is utilized. For both methods the same background meshes are used, in order to account for awkwardly cut elements, a nodal movement procedure is used. Scalar-valued and vector-valued problems in two and three dimensions are considered, respectively. Most of the problems stem from solid mechanics and, as customary, a normalized L^2-norm, ε_{L^2}, is used to investigate the convergence properties of the primary fields as well as the energy norm, here referred to as ε_{H^1}, to examine the convergence properties of the first derivatives. In order to concentrate on the essentials, we focus on two types of background meshes, the structured triangular and the triangular meshes with manipulated nodes, as well as structured tetrahedral and tetrahedral with manipulated nodes, they are shown in Fig. 74 and Fig. 81, respectively.

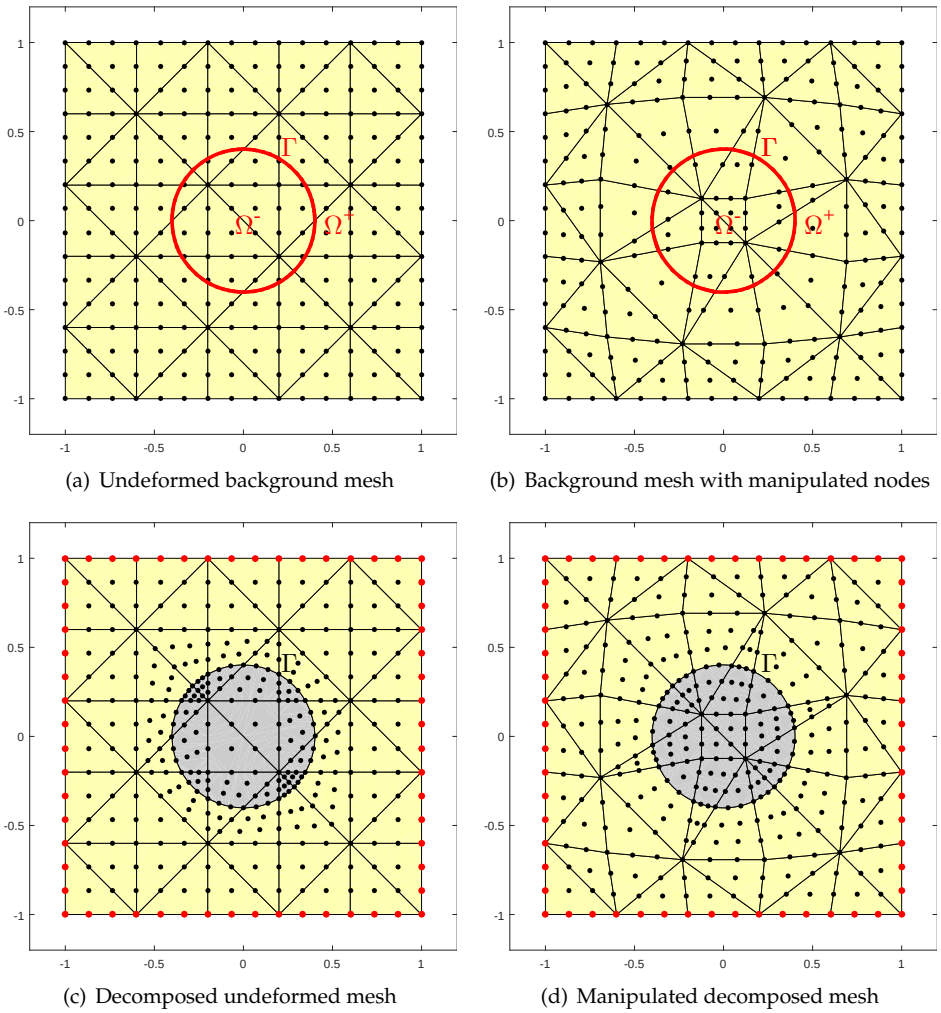
7.3.1 Numerical results in two dimensions - bi-material scalar-valued problem

The considered example is a classical Laplace problem and taken from [117]. The computational domain is the square $[-1,1]^2$ and the interface is a circle centered in the domain origin. The coarsest meshes are shown in Fig. 92, the background meshes in (a)-(b), the decomposed meshes in (c)-(d).

For the computational analysis, the outer boundary of Ω^+, represented by red circles in Fig. 92, is taken as Dirichlet boundary. The prescribed values are known from an analytical solution, which is given as

$$u(\boldsymbol{x}) = \begin{cases} \dfrac{r^2}{\alpha_1}, & 0 \leq r \leq r_0, \\ \dfrac{r^2}{\alpha_2} - \dfrac{r_0^2}{\alpha_2} + \dfrac{r_0^2}{\alpha_1}, & r_0 \leq r \leq b, \end{cases} \tag{7.11}$$

with $r = \sqrt{x^2 + y^2}$ and the radius of the inclusion taken as $r_0 = 0.401$. Herein, (α_1, α_2) are material coefficients corresponding to domains (Ω_1, Ω_2) and chosen as $\alpha_1 = 1$ and $\alpha_2 = 1000$. The source function yielding this exact solution is $f = -4$. To calculate the ε_{H^1}-norm, the gradient of



(a) Undeformed background mesh (b) Background mesh with manipulated nodes

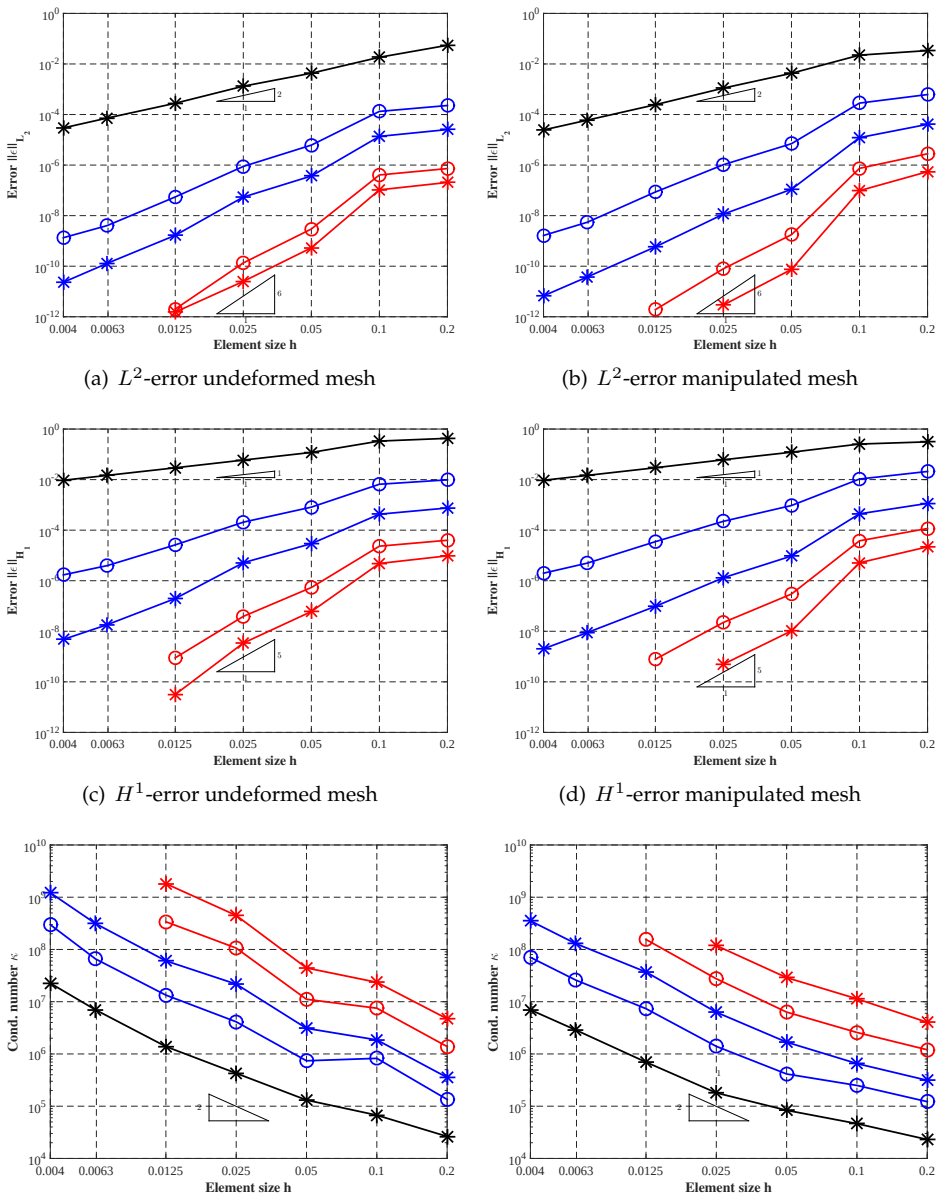
(c) Decomposed undeformed mesh (d) Manipulated decomposed mesh

Figure 92: Meshes used for the 2D interface problems. In (a)-(b) the undeformed background mesh as well as the background mesh with manipulated nodes are visualized. In (c)-(d) the corresponding meshes are depicted after the automatic conformal decomposition.

of Eq. (7.11) is needed, which is as

$$\nabla u(\boldsymbol{x}) = \begin{cases} \left[\frac{2x}{\alpha_1}; \frac{2y}{\alpha_1}\right], & 0 \leq r \leq r_0, \\ \left[\frac{2x}{\alpha_2}; \frac{2y}{\alpha_2}\right], & r_0 \leq r \leq b, \end{cases} \tag{7.12}$$

As evident from the analytical solution for the primary field in Eq. (7.11), as well as for the gradient in Eq. (7.12), the solution function is infinitely smooth within each of the subdomains (Ω_1, Ω_2). However, due to the material interface the function is globally only limited smooth, that is, $u \in C^0(\Omega)$. The results for the CDFEM are visualized in Fig. 93. The same sequence of meshes is used as for the reconstruction example.



(a) L^2-error undeformed mesh

(b) L^2-error manipulated mesh

(c) H^1-error undeformed mesh

(d) H^1-error manipulated mesh

(e) Condition number undeformed mesh

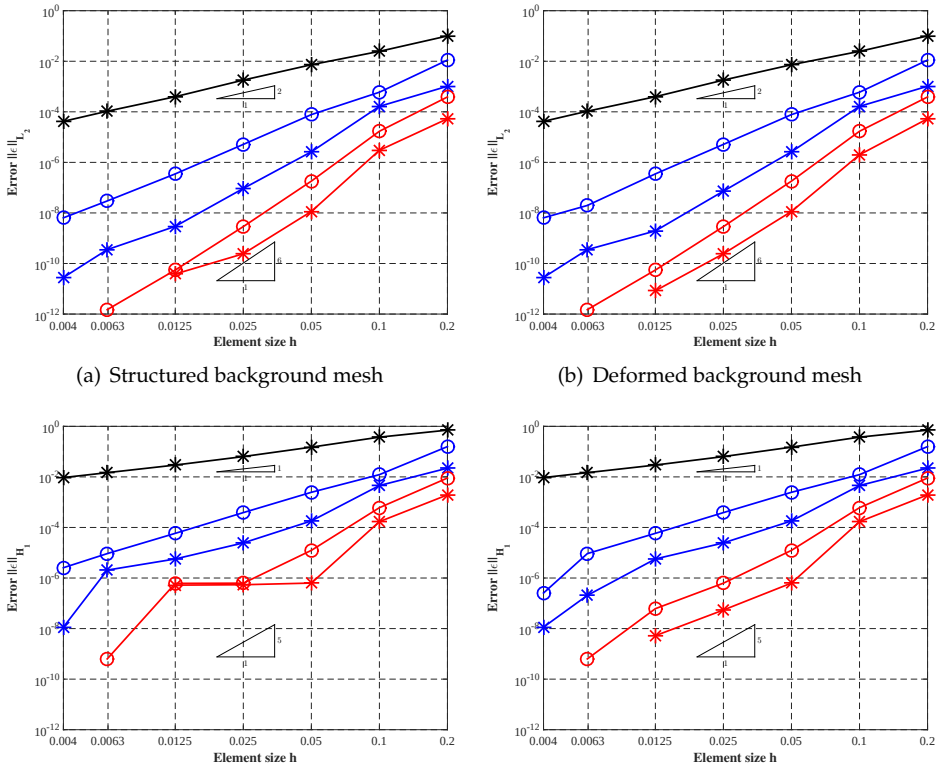
(f) Condition number manipulated mesh

Figure 93: Results of a CDFEM analysis on an undeformed background mesh and a background mesh with manipulated nodes in the left and right column, respectively. The error is measured in the L^2-norm (first row), in the H^1-norm (second row) and finally the condition number of the system matrix is recorded in the third row.

As evident from Fig. 93(a)-(d), we see for both background meshes visually indistinguishable results, that is, the expected optimal convergence rates for the primary field as well as for the derivatives. It is noticeable, that the increase of the condition number for both background meshes, as seen in Fig. 93(e)-(f), is close to the condition number of a standard FE analysis.

For comparison reasons we solve the same example using the NXFEM. As explained in Section 3, for the NXFEM it is necessary to modify the weak form, and to integrate additional terms along the zero level set. The integration points along the discontinuity are generated by creating

interface elements utilizing the reconstruction procedure in Section 4 and then mapping standard Gauß-Legendre points into these interface elements. We use the Hansbo-weighting for the choice of the averaging operator, see Section 3, and the stabilization parameter γ is chosen with $\gamma = \gamma_0 \left(\alpha_1 + \alpha_2\right) / h$ and $\gamma_0 = 10^2$.



(a) Structured background mesh

(b) Deformed background mesh

(c) Structured background mesh

(d) Deformed background mesh

Figure 94: Results of a NXFEM analysis on an undeformed background mesh and a background mesh with manipulated nodes in the left and right column, respectively. The error is measured in the L^2-norm (first row), in the H^1-norm (second row).

7.3.2 Numerical results in two dimensions - bi-material vector-valued problem

Next we consider a vector-valued problem in solid mechanics using the same geometry and the same background meshes as before. The example is a well known benchmark problem for higher-order extended finite element procedures, see among others [165, 34]. The error in the displacements is measured in the L^2-norm, the error in displacement derivatives is measured in the strain energy norm H_E defined as

$$\varepsilon_{H^{\hat{E}}} = ||\boldsymbol{u} - \boldsymbol{u}^h||_{H^E(\Omega)} = \sqrt{\int (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^h)^{\intercal} \mathbf{C} (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^h) \, \mathrm{d}\Omega}. \tag{7.13}$$

hereby $\mathbf{C}$ is the elastic stiffness tensor and $\boldsymbol{\epsilon}(\boldsymbol{u})$ and $\boldsymbol{\epsilon}^h(\boldsymbol{u}^h)$ the strain fields of the exact solution and the finite element approximation, respectively. We use relative versions of these two norms, therefore $\varepsilon_{L^2} = ||\boldsymbol{u} - \boldsymbol{u}^h||_{L^2(\Omega)}/||\boldsymbol{u}||_{L^2(\Omega)}$ and $\varepsilon_{H^E} = ||\boldsymbol{u} - \boldsymbol{u}^h||_{H^E(\Omega)}/||\boldsymbol{u}||_{H^E(\Omega)}$. Additionaly, the condition number of the system matrix is recorded. As before, assuming that the element edges align with the interface, optimal convergence rates are expected, see [168, 161].

The physical problem statement is an infinite domain consisting of material A with a circular inclusion of material B. As before, two different background meshes are chosen, with identical geometry specifications as the scalar problem in Fig. 92. On the entire outer domain boundary, represented by red circles in in Fig 92, Dirichlet boundary conditions are imposed using the analytical solution given in Eqs. (7.14) and (7.15).

The solution for the displacement components u_x and u_y in polar coordinates is presented in [165, 71] as

$$u_r(r,\theta) = \begin{cases} \left(1 - \dfrac{b^2}{a^2}\right)\alpha + \dfrac{b^2}{a^2} r, & 0 \leq r \leq a \\ \left(r - \dfrac{b^2}{r}\right)\alpha + \dfrac{b^2}{r}, & a \leq r \leq b, \end{cases} \tag{7.14}$$

$$u_\theta(r,\theta) = 0 \tag{7.15}$$

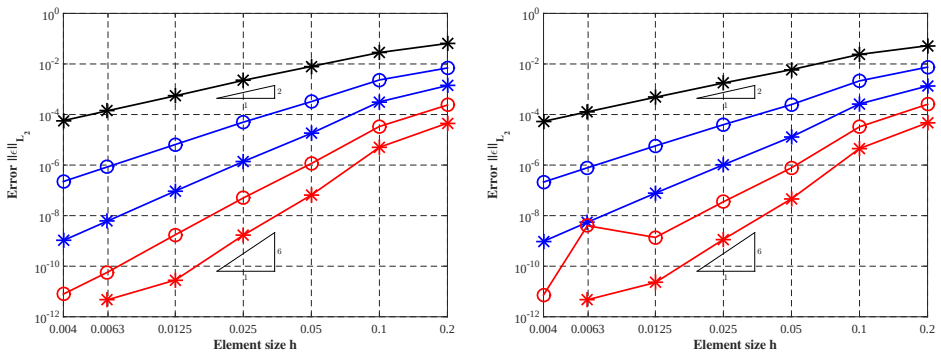
with the parameter α defined as

$$\alpha = \frac{(\lambda_1 + \mu_1 + \mu_2) b^2}{(\lambda_2 + \mu_{2)} a^2 + (\lambda_1 + \mu_1)(b^2 - a^2) + \mu_2 b^2}.$$

(θ, r) are the standard polar coordinates and μ the shear modulus. For the numerical computations, the material constants for material A and material B are chosen as $\{E_1 = 10, \nu_1 = 0.3\}$ and $\{E_2 = 1, \nu_2 = 0.25\}$, respectively. First we use a standard finite element formulation to solve the equations.

Fig. 95 displays the L^2-error as a function of the characteristic element size h. The results validate the proposed approach and show excellent agreement between the optimal and the calculated convergence rates.

The error in strain energy is calculated using the analytical solution for stresses and strains as presented in [165, 71]. The analytical strains in radial (ε_{rr}) and hoop ($\varepsilon_{\theta\theta}$) direction are given as



(a) Undeformed background mesh (b) Manipulated background mesh

Figure 95: Convergence results for the bi-material solid mechanics problem for an undeformed background mesh in (a) and for a mesh with manipulated nodes in (b). The error is measured in the standard L^2-norm.

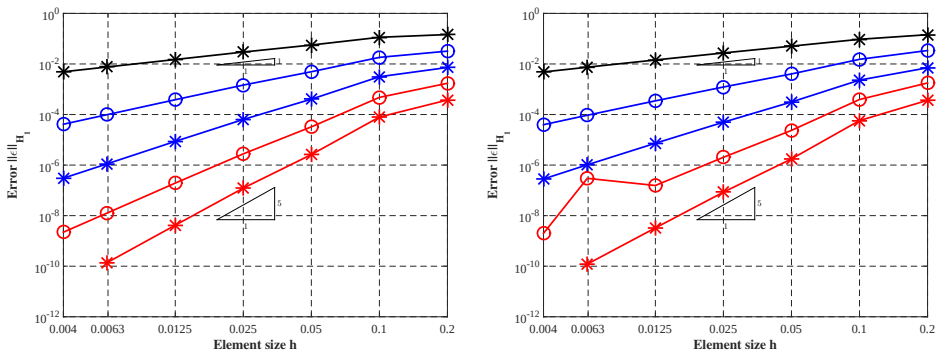
$$\varepsilon_{rr}(r,\theta) = \begin{cases} \left(1 - \dfrac{b^2}{a^2}\right)\alpha + \dfrac{b^2}{a^2}, & 0 \leq r \leq a, \\ \left(1 + \dfrac{b^2}{r^2}\right)\alpha + \dfrac{b^2}{r^2}, & a \leq r \leq b, \end{cases}$$

$$\varepsilon_{\theta\theta}(r,\theta) = \begin{cases} \left(1 - \dfrac{b^2}{a^2}\right)\alpha + \dfrac{b^2}{a^2}, & 0 \leq r \leq a, \\ \left(1 - \dfrac{b^2}{r}\right)\alpha + \dfrac{b^2}{r^2}, & a \leq r \leq b, \end{cases}$$

Using the analytical strains the stresses can be calculated easily, resulting in radial (σ_{rr}) and hoop ($\sigma_{\theta\theta}$) components given as

$$\sigma_{rr}(r,\theta) = 2\mu\varepsilon_{rr} + \lambda(\varepsilon_{rr} + \varepsilon_{\theta\theta})$$
$$\sigma_{\theta\theta}(r,\theta) = 2\mu\varepsilon_{\theta\theta} + \lambda(\varepsilon_{rr} + \varepsilon_{\theta\theta})$$

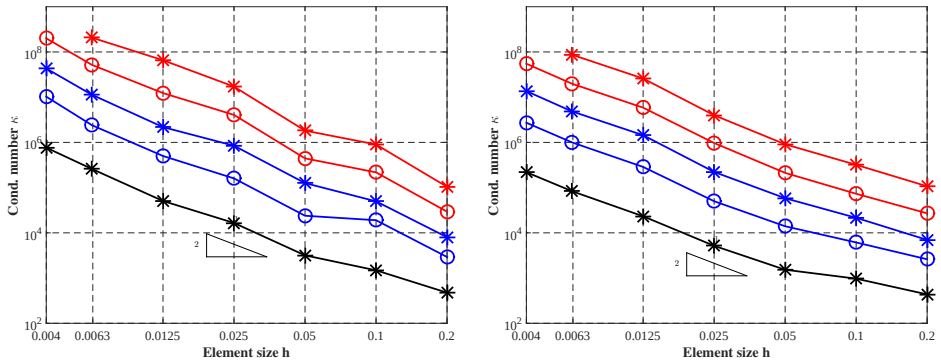
The results in Fig. 96 clearly show optimal convergence rates also for the strain energy norm.



(a) Undeformed background mesh (b) Manipulated background mesh

Figure 96: Convergence results for the bi-material solid mechanics problem for an undeformed background mesh in (a) and for a mesh with manipulated nodes in (b). The error in stresses and strains is measured in the standard H^E-norm.

Investigating the condition number κ of the system stiffness matrix and plotting κ over the characteristic element length h, Fig. 97(a) is obtained for the undeformed background mesh and Fig. 97(b) for the manipulated background mesh. It is seen, that, in this example for the undeformed background mesh, the condition number κ is not excessively high, therefore the utilization of the node manipulation procedure is not crucial here.



(a) Cartesian background mesh (b) Deformed background mesh

Figure 97: Dependency of the condition number on the element size for different element orders.

7.3.3 Spherical elastic inclusion under uniaxial tension in an infinite domain

The analytical solution for the bi-material three-dimensional problem was firstly established in a study of Goodier, see [84]. There, reasoning based on physical principles was used to determine the equations for displacements and stresses. Harmonic functions are chosen for displacements and stresses for both materials independently and are equated at the material interface. The presented solution is valid for a spherical inclusion embedded in an infinite domain with the remote uniaxial tension T as load function. Both materials, the inclusion and the surrounding matrix, are assumed homogeneous, isotropic and linear elastic.

The original solution is defined in terms of standard spherical coordinates (r, ϕ, θ), with r as

the radius of the sphere emanating from the center of the sphere, θ as azimuth angle and ϕ as the unity vector from the center of the sphere to the considered point $\boldsymbol{x}$. It is worth mentioning that symmetry around the angle θ.

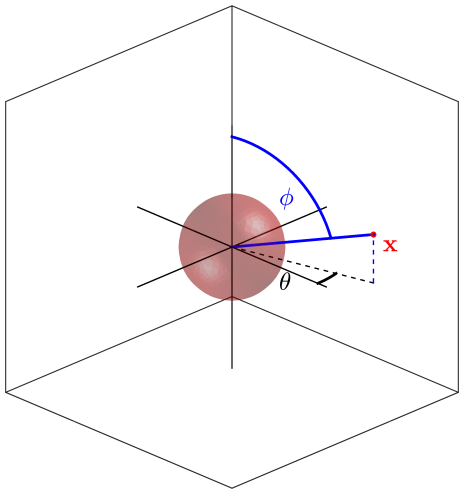


Figure 98: Bimaterial spherical inclusion in an infinite domain and the definition of the used spherical coordinates r, ϕ, θ.

The solution has the same regularity properties as the circular inclusion in 2D, a weak discontinuity on the material interface, therefore it is necessary to subdivide the solution in two parts. The components in the external infinitely extended domain are further on denoted with the subscript m for **m**atrix, whereas the solution in the **i**nclusion is denoted with the subscript i.

We start with the displacements and the stress components for a *homogeneous* elastic solid with the material coefficients E, ν and a uniform tension T in direction $\theta = 0$ as a loading function. The solution on a spherical surface of radius R is then

$$u_r^0 = \frac{Tr}{2E}\left[(1-\nu) + (1+\nu)\cos(2\theta)\right] \tag{7.16}$$

$$u_\theta^0 = -\frac{Tr}{2 \cdot E}(1+\nu)\sin(2\theta) \tag{7.17}$$

$$\sigma_{rr}^0 = \frac{T}{2}(1+\cos(2\theta)) \tag{7.18}$$

$$\sigma_{r\theta}^0 = -\frac{T}{2}\sin(2\theta) \tag{7.19}$$

For the final displacement field within the matrix it is necessary to include additional terms to the Eq. (7.16) and Eq. (7.17)

$$
\begin{aligned}
u_r^m &= u_r^0 - \frac{A}{r^2} - \frac{3B}{r^4} + \left[\left(\frac{5-4\nu_m}{1-2\nu_m}\right)\frac{C}{r^2} - \frac{9B}{r^4}\right]\cos(2\theta) \\
u_\theta^m &= u_\theta^0 - \left(\frac{2C}{r^2} + \frac{6B}{r^4}\right)\sin(2\theta) \\
\sigma_{rr} &= 2\mu\left[\frac{2A}{r^3} - \frac{2\nu_m}{1-2\nu_m}\frac{C}{r^3} + 12\frac{B}{r^5} + \left(-\frac{2\,(5-\nu_m)}{1-2\nu_m}\frac{C}{r^3} + 36\frac{B}{r^5}\right)\cos(2\theta)\right] \\
\sigma_{\theta\theta} &= 2\mu\left[-\frac{A}{r^3} - \frac{2\nu_m}{1-2\nu_m}\frac{C}{r^3} - 3\frac{B}{r^5} + \left(\frac{C}{r^3} - 21\frac{B}{r^5}\right)\cos(2\theta)\right] \\
\sigma_{\psi\psi} &= 2\mu\left[-\frac{A}{r^3} - \frac{2\,(1-\nu_m)}{1-2\nu_m}\frac{C}{r^3} - 9\frac{B}{r^5} + \left(3\frac{C}{r^3} - 15\frac{B}{r^5}\right)\cos(2\theta)\right] \\
\sigma_{r\theta} &= 2\mu\left[-\frac{2\,(1+\nu_m)}{1-2\nu_m}\frac{C}{r^3} + 24\frac{B}{r^5}\right]\sin(2\theta)
\end{aligned}
$$

with A, B, C as arbitrary coefficients which are yet to be determined. An analogous system of equations can be set up for the region inside the spherical inclusion, yielding

$$
\begin{aligned}
u_r^i &= Hr + Fr + 2\nu_i G r^3 + \left(3Fr + 6\nu_i G r^3\right)\cos(2\theta) \\
u_\theta^i &= -\left[3Fr + (7-4\nu)\,Gr^3\right]\sin(2\theta) \\
\sigma_{rr} &= 2\mu\left[\frac{1+\nu_i}{1-2\nu_i}H + F - \nu_i G r^2 + \left(3F - 3\nu_i G r^2\right)\cos(2\theta)\right] \\
\sigma_{\theta\theta} &= 2\mu\left[\frac{1+\nu_i}{1-2\nu_i}H + F - 5\nu_i G r^2 - \left(3F + 7\,(2-\nu_i)\,Gr^2\right)\cos(2\theta)\right] \\
\sigma_{\psi\psi} &= 2\mu\left[\frac{1+\nu_i}{1-2\nu_i}H - 2F - (15-7\nu_i)Gr^2 - (7+11\nu_i)\,Gr^2\cos(2\theta)\right] \\
\sigma_{r\theta} &= -2\mu\left[3F + (7+2\nu_i)\,Gr^2\right]\sin(2\theta)
\end{aligned}
$$

with H, F, R as coefficients to be determined by compatibility conditions. Equating the displacements and the tractions at the boundary of the inclusion, at $r = R$, the unknown coefficients can be obtained as

$$
\begin{aligned}
\frac{A}{R^3} &= -\frac{T}{8\nu_m}\frac{\mu_m-\mu_i}{(7-5\nu_m)\,\mu_m+(8-10\nu_m)\,\mu_i}+\dots \\
&\quad \frac{(1-2\nu_i)\,(6-5\nu_m)\,2\mu_m+(3+19\nu_i-20\nu_m\nu_i)\,\mu_i}{(1-2\nu_i)\,2\mu_m+(1+\nu_i)\,\mu_i}+\dots \\
&\quad \frac{T}{4\mu_m}\frac{\left[(1-\nu_m)\,\frac{1+\nu_i}{1+\nu_m}-\nu_i\right]\mu_i-(1-2\nu_i)\,\mu_m}{(1-2\nu_i)\,2\mu_m+(1+\nu_i)\,\mu_m} \\
\frac{B}{R^5} &= \frac{T}{8\mu_m}\frac{\mu_m-\mu_i}{(7-5\nu_m)\,\mu_m+(8-10\nu_m)\,\mu_i} \\
\frac{C}{R^3} &= \frac{T}{8\mu_m}\frac{5\,(1-2\nu_m)\,(\mu_m-\mu_i)}{(7-5\nu_m)\,\mu_m+(8-10\nu_m)\,\mu_i} \\
F &= \frac{5\cdot T}{4}\frac{1-\mu_m}{(7-5\nu_m)\,\mu_m+(8-10\nu_m)\,\mu_i} \\
G &= 0 \\
H &= \frac{T\,(1-\mu_m)}{2\,(1+\mu_m)}\frac{1-2\mu_i}{(2-4\mu_i)\,\mu_m+(1+\nu_i)\,\mu_i}
\end{aligned}
$$

Notice that the stress function does not depend on the radius due to $G = 0$. In order to get the Cartesian components of the equations some coordinate transformation is necessary. Here, for the displacement field, vector components have to be transformed, this is done by

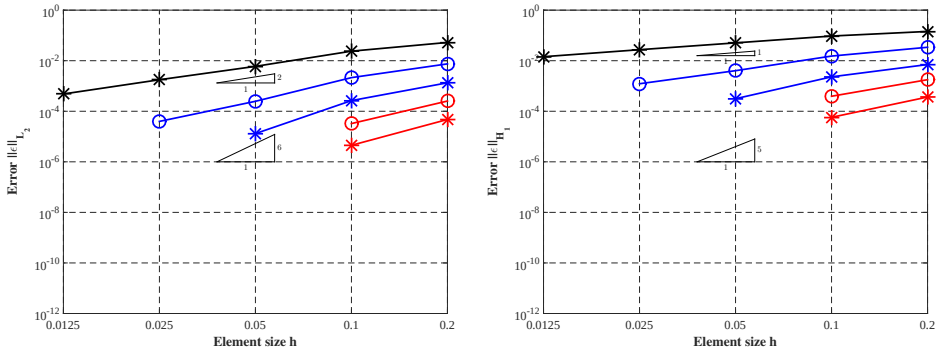
$$
\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \sin(\theta)\cos(\phi) & \cos(\theta)\cos(\phi) & -\sin(\phi) \\ \sin(\theta)\sin(\phi) & \cos(\theta)\sin(\phi) & \cos(\phi) \\ \cos(\theta) & -\sin(\theta) & 0 \end{bmatrix} \begin{bmatrix} u_r \\ u_\phi \\ u_\theta \end{bmatrix}
$$

The transformation matrix is referred to as rotation matrix further on designated as $\boldsymbol{T}^{\boldsymbol{r}}_{\boldsymbol{x}}$. The transformation of tensor components of the stresses are then given as

$$
\boldsymbol{\sigma_r} = \boldsymbol{T}^{\boldsymbol{r}}_{\boldsymbol{x}}\boldsymbol{\sigma_x}\boldsymbol{T}^{\boldsymbol{r}}_{\boldsymbol{x}}.
$$

We conduct the same numerical investigations as in the reconstruction section. We use $\{5, 10, 20, 40, 80\}$ elements per dimension and therefore five times the number of tetrahedra. Starting with the unfitted background mesh the automatic decomposition procedure yields a mesh that accounts for the sphere. The radius of the sphere is defined as $r = 0.4$. The material parameters are chosen with $E_m = 1000, \nu_m = 0.3$ for the surrounding matrix and $E_i = 1, \nu_i = 0.25$ for the inclusion. The error is investigated in the L^2-norm in the energy norm H^E and also the condition number is recorded.

The results in Fig. 99(a)-(b) indicate optimal convergence properties in both investigated norms. It is noted that the resulting large number of degrees of freedom restricts the number of mesh refinements. Only results for less than ≈ 500.000 DOFs are shown. Similar to the two-dimensional studies, the condition number of the system matrix increases also linearly with the number of elements. In sum, the findings of the two-dimensional studies can be transferred to three dimensions.



(a) L^2-error undeformed mesh (b) H^E-error manipulated mesh

Figure 99: Results of a CDFEM analysis for a problem with a bi-material spherical inclusion on an undeformed background mesh. The error is measured in the L^2-norm in (a) and in the H^E-norm in (b).

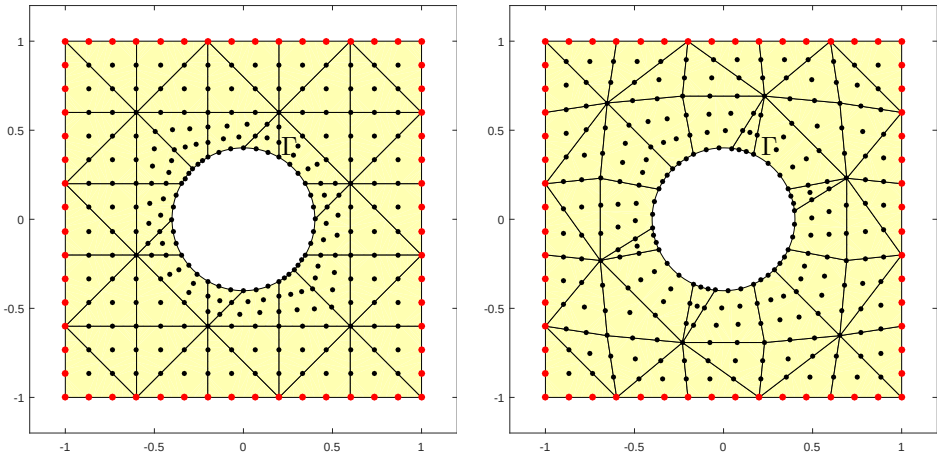
7.4 Boundary value problems II - Fictitious domain problems

Next we treat fictitious domain problems in a similar fashion than the interface problems of the previous section. Instead of having two different materials, there is only one material in the domain and a void region on the other side. As a consequence, now generally curved boundaries, either Dirichlet or Neumann type, are within of the elements, which complicates the situation in FDMs.. As before, the boundaries are described implicitly using level set functions. The treatment with the CDFEM is very much alike to the procedure for interfaces, the only difference being that the elements outside the physical domain are neglected when integrating the stiffness matrix. When using the CDFEM, the treatment of boundary conditions very much simplifies compared to FDMs and is the same than for classical FEM on conforming meshes.

7.4.1 Infinite plate with a circular hole

The example is, as before, a benchmark problem for higher order finite element procedures, see e.g. [48, 34]. The error in the displacements (measured in the L^2-norm), the error in displacement derivatives (measured in H^E-norm) and also the condition number of the system of equations are investigated.

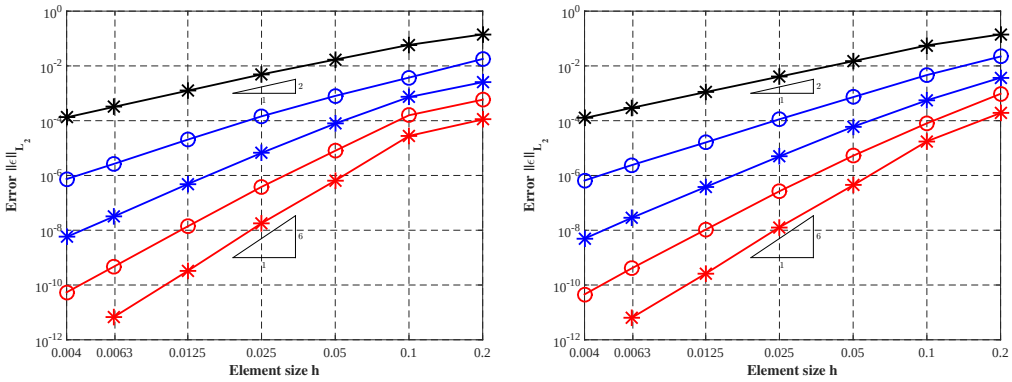
The problem statement is an infinite domain with a circular inclusion and unidirectional tension in x-direction. As before, two different background meshes are chosen, an undeformed background mesh and a background mesh with node manipulations, both are depicted in Fig. 100(a) and (b). The circular hole with the radius $a = 0.4$ is centered in the origin. Instead of traction boundary conditions, on the entire outer domain boundary, represented by red circles in Fig 100, Dirichlet boundary conditions are imposed using the analytical solution given in Eq. (7.20) and Eq. (7.21).



(a) Cartesian background mesh (b) Deformed mesh

Figure 100: Undeformed (a) and mesh with node manipulations (b) with a circular hole and Dirichlet boundary conditions on the outer boundary.

The solution for the displacement components u_x and u_y in Cartesian coordinates is given in



(a) Structured background mesh (b) Deformed background mesh

Figure 101: Convergence study for the displacements in the L^2-norm, Eq. (2.42).

[168] as

$$u_x(r,\theta) = \frac{t_x a}{8\mu}\left[\frac{r}{a}(\kappa+1)\cos\theta + \frac{2a}{r}((1+\kappa)\cos\theta + \cos 3\theta) - \frac{2a^3}{r^3}\cos 3\theta\right] \tag{7.20}$$

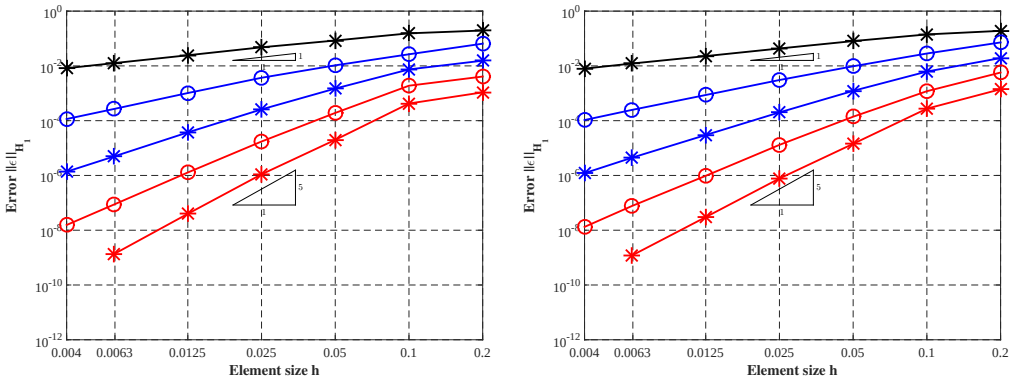
$$u_y(r,\theta) = \frac{t_x a}{8\mu}\left[\frac{r}{a}(\kappa-3)\sin\theta + \frac{2a}{r}((1-\kappa)\sin\theta + \sin 3\theta) - \frac{2a^3}{r^3}\sin 3\theta\right] \tag{7.21}$$

with (θ, r) being the standard polar coordinates, t_x the traction in x-direction, and μ the shear modulus. For our example, plane strain conditions are considered, therefore the Kolosov constant is given as $\kappa = 3 - 4\nu$, with ν as Poisson ratio. For the following numerical computations, the material constants are chosen as $E = 10^4$ and $\nu = 0.3$.

$$\begin{aligned}
\sigma_{xx}(r,\theta) &= t_x - t_x\frac{a^2}{r^2}\left(\frac{3}{2}\cos 2\theta + \cos 4\theta\right) + t_x\frac{3a^4}{2r^4}\cos 4\theta \\
\sigma_{yy}(r,\theta) &= -t_x\frac{a^2}{r^2}\left(\frac{1}{2}\cos 2\theta - \cos 4\theta\right) - t_x\frac{3a^4}{2r^4}\cos 4\theta \\
\tau_{xy}(r,\theta) &= -t_x\frac{a^2}{r^2}\left(\frac{1}{2}\sin 2\theta + \sin 4\theta\right) + t_x\frac{3a^4}{2r^4}\sin 4\theta
\end{aligned}$$

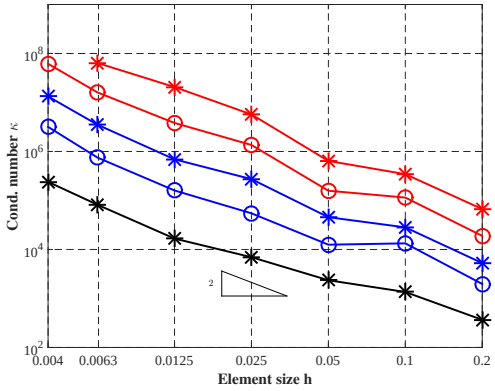
As before, the mesh is sequentially subdivided in $\ell_h = \{8, 16, 32, 64, 128\}$ quadrilateral elements per side. The automatically generated elements inside the inclusion are simply neglected when integrating the weak form of the partial differential equation. The first convergence study measures the displacement error in the L^2-norm, ε_{L^2} as introduced in Eq. (2.42). The dotted lines show the theoretically optimal convergence rates. Fig. 101 displays the error as a function of the characteristic element length h. The results show again excellent agreement between the optimal and the calculated convergence rates.

The exact stresses and strains are given in e.g. [48, 34]. The error in strain energy is calculated using the error measure in Eq. (2.43). The results are shown in Fig. 102 and are again optimal. The condition number for the Cartesian and the deformed background mesh are plotted in Fig. 102(c)-(d).

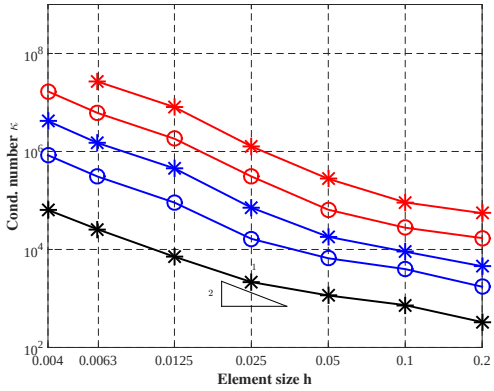


(a) Structured background mesh

(b) Deformed background mesh



(c) Structured background mesh



(d) Deformed background mesh

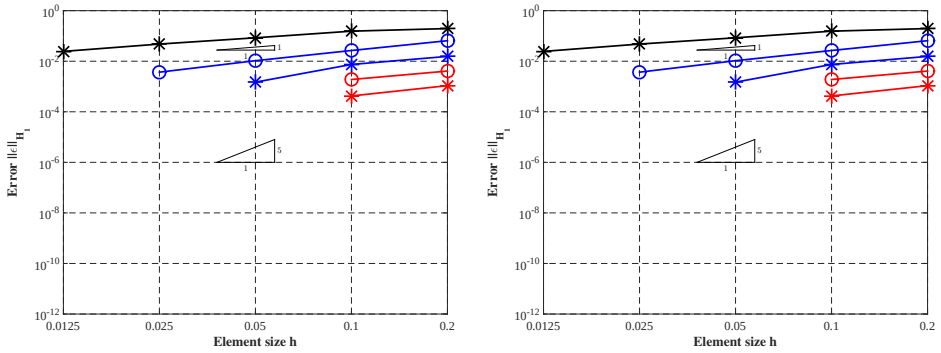
Figure 102: Convergence results for the plate with a circular hole problem in the H^E-norm in (a),(b) and the condition number κ in (c), (d) for a undeformed and a deformed background mesh, respectively.

7.4.2 Spherical cavity under uniaxial tension in an infinite domain

The analytical solution for the spherical cavity is also found in [84]. In fact, the equations from the example with the bi-material inclusion for the displacement field as well of the stress field are all valid, however, the coefficients have to be adjusted to the situation of a hole instead of an elastic inclusion. There the shear modulus $\mu_i = 0$, then the coefficients $A - C$ follow as

$$\begin{aligned} A &= -R^3 \frac{T}{8\mu_m} \frac{13 - 10\nu_1}{7 - 5\nu_1} \\ B &= R^5 \frac{T}{8\mu_m} \frac{1}{7 - 5\nu_1} \\ C &= R^3 \frac{T}{8\mu_m} \frac{5\,(1 - 2\nu_1)}{7 - 5\nu_1} \\ F &= \frac{5 \cdot T}{4} \frac{1 - \mu_m}{(7 - 5\nu_m)\,\mu_m} \\ G &= 0 \\ H &= \frac{T\,(1 - \mu_m)}{2\,(1 + \mu_m)} \frac{1}{2\mu_m} \end{aligned}$$

The same investigations as for the elastic inclusion are made, also the same meshes are used, however, the elements within the inclusion are neglected for the integration of the stiffness matrix. The results shown if Fig. 103(a)-(b) indicate similar behaviour as for the bi-material problem.



(a) L^2-error undeformed mesh (b) H^E-error undeformed mesh

Figure 103: Results of a CDFEM analysis for a problem with a spherical hole on an undeformed background mesh. The error is measured in the L^2-norm in (a) and in the H^E-norm in (b).

7.5 Elastoplastic problems

7.5.1 Elastoplastic hollow cylinder

The thick-walled elastoplastic cylinder subjected to an internal pressure is one of the very few examples in two dimensions which is susceptible to analytical methods for the displacement field and the stress field. The problem geometry is a thick-walled cylindrical tube, as shown in Fig. 104, with an internal radius r_i, and outer radius r_o. The tube is subjected to normal pressure p_i on the inner surface. Due to the geometry and the loading function plane strain conditions are assumed, therefore, the stresses and the strains are independent of the axial coordinate z. These assumptions are justified when the length of the cylinder is great enough.

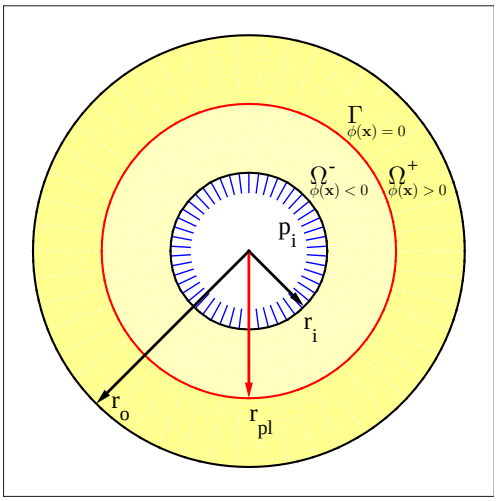


Figure 104: The overall domain of interest is a hollow cylinder. The domain is composed of the two sub-domains Ω^+ and Ω^-, representing the elastic and the plastic domain, respectively. The zero contour of $\phi(\boldsymbol{x})$, here denoted with Γ, is the elastoplastic interface. The loading function is a constant internal pressure p_i

The cylinder is made of an homogeneous elastoplastic material. The analytical solution is valid for a Tresca yield criterion, which is mostly applied to model isotropic ductile materials. The implementation of the stress integration algorithm is somewhat troublesome, which is due the sharp features of the yield surface, namely, an infinitely long hexagonal cylinder in the stress space. The numerical treatment of the edges requires particular attention, in the return-mapping algorithm as well as in the consistent elastoplastic tangent modulus, for more details see [50]. The crucial point for this particular problem is that the shape and the location of the elastoplastic interface is known beforehand. Due to the symmetry of the system and the loading function, the elastoplastic interface is circular, the exact location is a function of the material parameters.

The closed-form expressions, for both displacements and stresses, are given in [96, 121]. In addition, in the cited literature other analytical solutions for a alternative assumptions, as pure elasticity and plane stress conditions, can be found. The exact location of the boundary, denoted with r_{pl} can be obtained noting that the elastic stresses *must* be equal to the yield stresses at $r = r_{pl}$. This results in the following relationship between the loading and the elastoplastic interface

$$p_i = \sigma_y \left(1 - \frac{r_{pl}^2}{r_o^2} + \ln\left(\frac{r_{pl}}{r_i}\right)\right)$$

Consequently, given a certain loading amplitude, the location of the interface can be calculated, or, given the desired location the absolute value of the loading function can be determined.

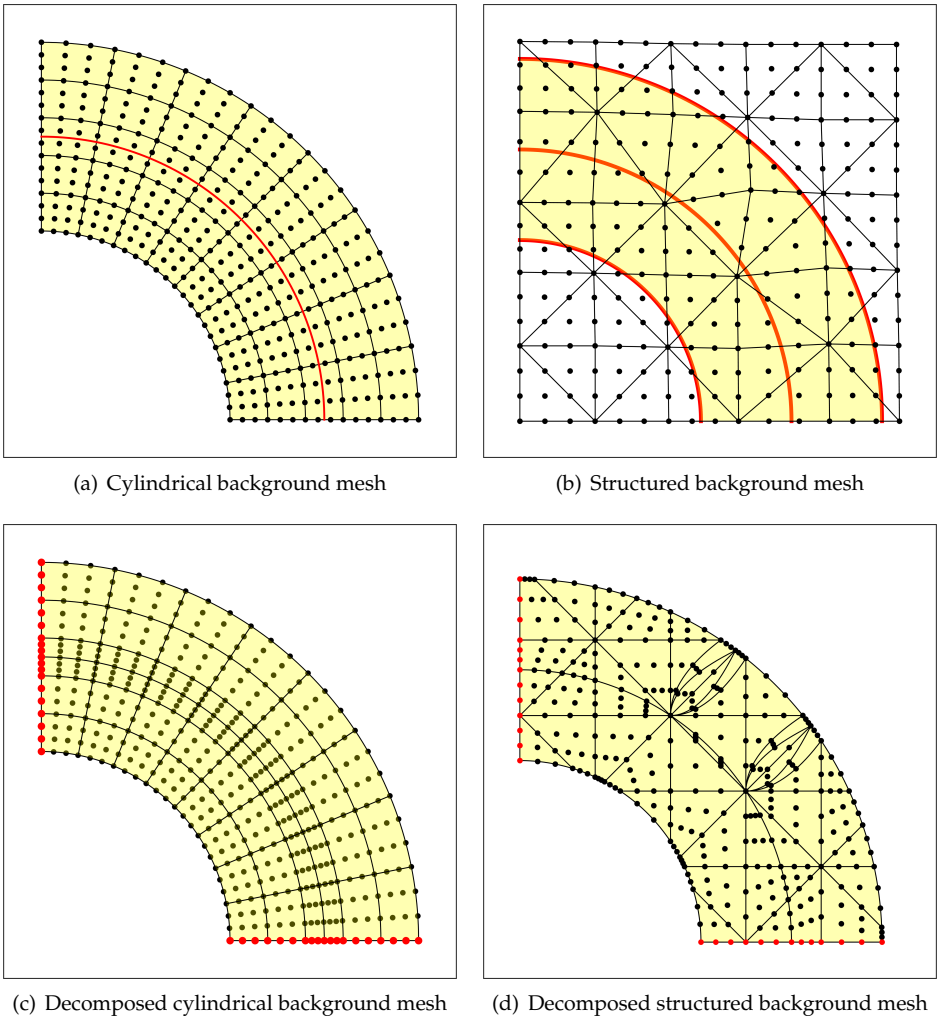
In the elastic zone, $r_{pl} < r < r_o$, the analytical solution is given as

$$\begin{aligned}
u_r &= \sigma_y \frac{(1+\nu)\, r_{pl}^2}{E r_o^2} \left((1-2\nu)\, r + \frac{r_o^2}{2r}\right) \\
\sigma_r &= -\sigma_y \left(\frac{r_{pl}^2}{r^2} - \frac{r_{pl}^2}{r_o^2}\right) \\
\sigma_\theta &= \sigma_y \left(\frac{r_{pl}^2}{r^2} + \frac{r_{pl}^2}{r_o^2}\right)
\end{aligned}$$

In the elastoplastic regime, $0 < r < r_{pl}$, the following equations are valid

$$\begin{aligned}
u_r &= \frac{(1+\nu)\,(1-2\nu)}{E} r\sigma_r + \frac{2\sigma_y r_{pl}^2 \left(1-\nu^2\right)}{Er}. \\
\sigma_r &= -\sigma_y \left(1 - \frac{r_{pl}^2}{r_o^2} + \ln\left(\frac{r_o}{r^2}\right)\right) \\
\sigma_\theta &= \sigma_y \left(1 + \frac{r_{pl}^2}{r_o^2} - \ln\left(\frac{r_o}{r^2}\right)\right).
\end{aligned}$$

For the convergence studies, two different types of meshes are used. In Fig. 105(a) only the elastoplastic interface is of interest, therefore background meshes are used which align with the external boundaries. A circular zero level set is used to describe the elastoplastic interface, the decomposed mesh is shown in Fig. 105(c). This approach is not mandatory, it is possible to use e.g. a Cartesian grid or, in fact, any other background mesh and define all external and internal interfaces using level set functions. To demonstrate this, a structured triangular mesh is used, as shown in Fig. 105(b), and the mesh to carry out the simulations is shown in Fig. 105(d). The material properties are chosen with $E = 210, \nu = 0.3$ and the yield stress is $\sigma_y = 10$.



(a) Cylindrical background mesh

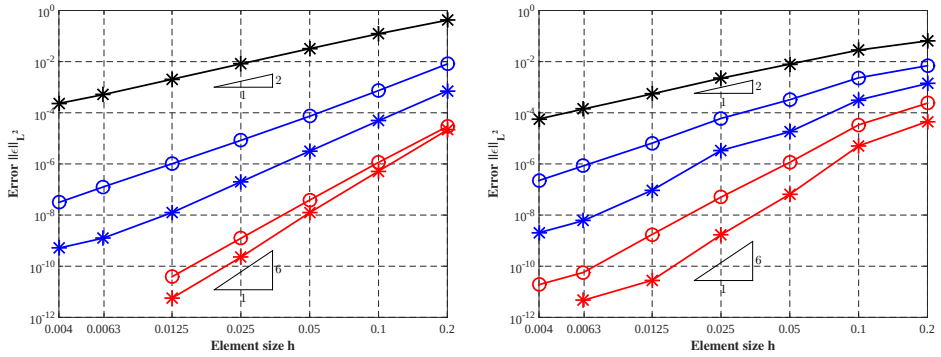
(b) Structured background mesh

(c) Decomposed cylindrical background mesh

(d) Decomposed structured background mesh

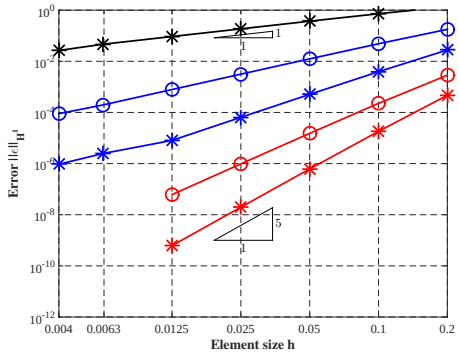
Figure 105: Exemplary background meshes for $m_\Omega = 3$ not taking into account the elastoplastic interface in (a) nor the other external interfaces in (b). Meshes after the automatic subdivision, conforming with all interfaces in (c),(d).

As seen from the results in Fig. 106, the convergence behaviour in all investigated norms is optimal. It is noted, that studies were also made for a purely elastic tube, with similar results, meaning, all element orders converged optimally.

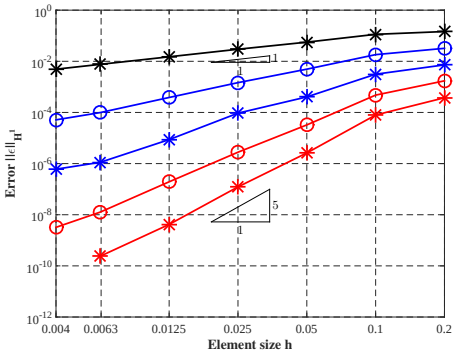


(a) Cylindrical background mesh

(b) Structured background mesh



(c) Cylindrical background mesh



(d) Structured background mesh

Figure 106: Convergence results for the hollow cylinder and an elastoplastic material. The error is measured in the L^2-norm in (a) and in the H^1-norm in (b).

7.5.2 Elastoplastic hollow sphere under internal pressure

The problem is basically the three dimensional analogy to the elastoplastic hollow cylinder and the analytical solution is found in [96, 121]. In fact, to the authors best knowledge, it is the only analytical solution to three dimensional elastoplastic problems. Two of the principal stresses are equal, and therefore, the yield criteria for both the Tresca and von Mises model coincide. Similar to the elastoplastic tube, the location of the elastoplastic interface is known. It separates the two shell-shaped domains as shown in Fig. 107.

The analytical solution in the elastic range is defined as

$$p_i = \sigma_y \frac{2}{3} \left(1 + 3 \ln \left(\frac{r_{pl}}{r_i} \right) - \frac{r_{pl}^3}{r_o^3} \right)$$

As for the cylinder, the exact location of the interface can be calculated, or, given the desired location the absolute value of the loading function can be determined. In the elastic zone, $r_{pl} < r < r_o$, the analytical solution for the displacement and stresses is given as

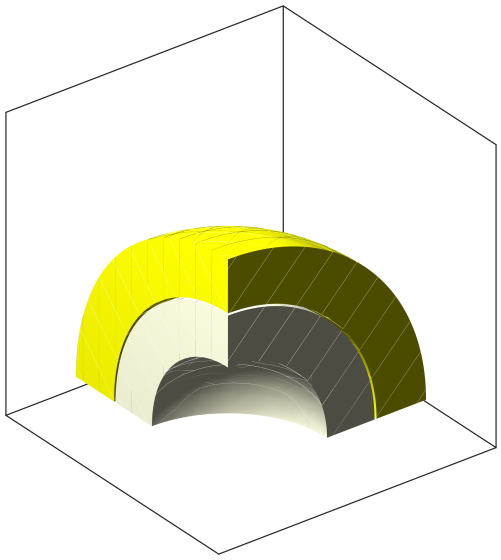
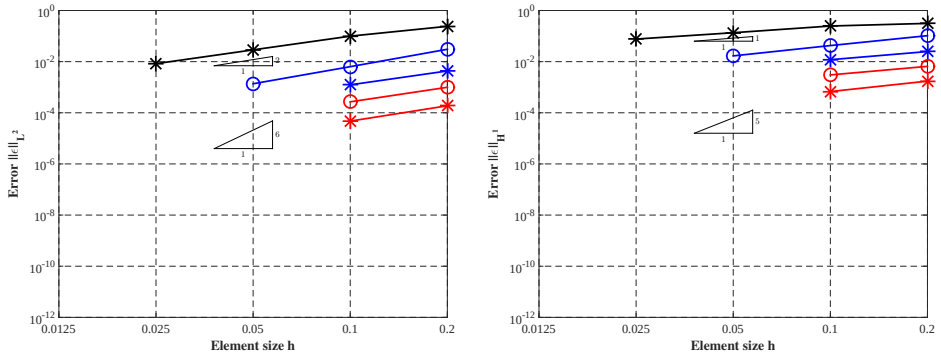


Figure 107: The overall domain of interest is a homogeneous hollow sphere. The domain is composed of the two sub-domains representing the elastic and the plastic domain, respectively.

$$
\begin{aligned}
u_r &= \sigma_y \frac{2r_{pl}^3}{3Er_o^3}\left((1-2\nu)\,r + (1+\nu)\,\frac{r_o^3}{2r^2} - 1\right)\\
\sigma_r &= \sigma_y \frac{-2r_{pl}^3}{3r_o^3}\left(\frac{r_o^3}{r^2} - 1\right)\\
\sigma_\theta &= \sigma_y \frac{2r_{pl}^3}{3r_o^3}\left(1 + \frac{r_o^3}{2r^2}\right).
\end{aligned}
$$

In the elastoplastic regime, $0 < r < r_{pl}$, the following equations are valid

$$
\begin{aligned}
u_r &= \sigma_y \frac{r}{E}\left((1-2)\,\frac{r_{pl}^3}{r^3} - \frac{2}{3}(1-2\nu)\left(1 + 3\ln\left(\frac{r_{pl}}{r}\right) - \frac{r_{pl}^3}{r_o^3}\right)\right)\\
\sigma_r &= -\sigma_y \frac{2}{3}\left(1 + 3\ln\left(\frac{r_{pl}}{r}\right) - \frac{r_{pl}^3}{r_o^3}\right)\\
\sigma_\theta &= \sigma_y \frac{2}{3}\left(\frac{1}{2} - 3\ln\left(\frac{r_{pl}}{r}\right) + \frac{r_{pl}^3}{r_o^3}\right).
\end{aligned}
$$



(a) L^2-error undeformed mesh (b) H^1-error undeformed mesh

Figure 108: Results of a CDFEM analysis for a elastoplastic thick shell on an undeformed background mesh. The error is measured in the L^2-norm in (a) and in the H^1-norm in (b).

For our examples the same background meshes as for the elastic inclusion are used. The material properties are chosen as for the elastoplastic tube. The results in Fig. 108(a)-(b) indicate optimal convergence properties in both investigated norms.

8 Conclusions and outlook

In this thesis we presented a new higher-order accurate method for the approximation of interface problems and embedded domain problems. The considered mathematical models include classical interface problems such as bi-material problems with a sharp material interface, which is in general not fitted by the computational mesh. The second group of problems are unfitted boundary value problems, where the Dirichlet or Neumann boundaries are located within elements and the true computational domain is contained in a larger fictitious one. The proposed method may be considered as a fictitious domain method or alternatively a higher-order remeshing scheme. We summarize the most important contributions of this thesis and discuss open problems.

8.1 Summary

The contribution consists of an integration and an approximation part.

The introduced method requires an implicit description of the geometry. This can be given from the outset, otherwise, in case of CAD surface representations, the explicit information can be transformed into the level set context using geometric distance operations. The proposed technique couples the level set method with a customized Newton-Raphson solver to obtain interface meshes of arbitrary order, also recursive refinement is possible in order to capture local, complex features. This results in an explicit, higher-order accurate, geometrical description of implicit features on an arbitrary background mesh. This interface mesh can be used e.g. for pre- or post-processing purposes, or even more useful, in the context of fictitious domain methods as an integration mesh. The validity and the accuracy of the mesh is shown on a number of examples in various contexts, including the approximation of BVPs utilizing the NXFEM.

If the method is from the outset employed for the approximation of BVPs, a conforming mesh is necessary and recursive refinement and hanging nodes are not permitted. The problem, particularly in a higher-order context, is the shape regularity of the elements. It is shown that the interpolation properties are not affected by the awkwardly shaped elements, but the condition number can become unbounded. It is demonstrated that small node manipulations can help, however, it must be ensured to finally yield in a valid mesh.

In conclusion, the current thesis provides a new promising approach for interface and unfitted problems. The application field in this thesis is mainly in the realm of solid mechanics, however, the features render the method a candidate for a broader class of application fields.

8.2 Open problems and outlook

The method was already applied successfully to partial differential equations on manifolds, [75]. There, the manifolds are given implicitly and the reconstructed interface mesh was used as a problem domain. An interesting application field of the method would be the context of moving and evolving geometries and interfaces, along the lines of the first publications of the CDFEM but with higher-order accuracy. Such problems are omnipresent, including interactions between different phases, problems with free boundaries and moving interfaces, and at last also topology optimization problems fall in this category. Also, it is not clear yet how the method performs in

the context of mixed FE applications, that is, when the primary and the secondary variables are discretized, e.g. for incompressible problems.

In this thesis we assumed some restrictions on the curvature of the zero level set. It was argued, that either a global or a local refinement would resolve the problem. An open topic is an efficient, adaptive refinement of the background mesh, possibly for moving domains. Some general refinement criteria need to be found, and a robust implementation in two and three dimensions is not effortless.

The nodal based stabilization works for the investigated problems quite well. Nevertheless some alternative methods need to be considered. One candidate are stabilization methods from other fictitious domain methods, first and foremost the ghost penalty approach in the CutFEM. It can be argued that for both methods the problem cause, the small elements and the eventually unbounded gradient, is very similar. Therefore, this stabilization approach might work also for the CDFEM. Another issue, which was up to now barely investigated, is the effect of the ghost penalty on curved higher-order elements.

The application of the CDFEM for plasticity was here introduced in context of the deformation theory of plasticity and low-order approximations and in addition to the rather particular case of beforehand known location of the elastoplastic interface. An extension to higher-orders for general problems is desirable and conceptually feasible, however, some difficulties are to be expected, especially in the context of history-dependent materials. The problem is the higher-order accurate detection of the zero level set and the transformation of history variables between different meshes.

References

[1] Abedian, A.; Parvizian, J.; Düster, A.; Khademyzadeh, H.; Rank, E.: Performance of different integration schmenes in facing discontinuities in the Finite Cell Method. *Int. J. Comput. Methods*, **10**, 2013.

[2] Abedian, A.; Parvizian, J.; Düster, A.; Rank, E.: The finite cell method for the J_2 flow theory of plasticity. *Finite Elem. Anal. Des.*, **69**, 37–47, 2013.

[3] Abedian, A.; Parvizian, J.; Düster, A.; Rank, E.: Finite cell method compared to h-version finite element method for elasto-plastic problems. *Appl. Math. Mech.*, **35**(10), 1239–1248, 2014.

[4] Auricchio, F.; Brezzi, F.; Lefieux, A.; Reali, A.: An 'immersed' finite element method based on a locally anisotropic remeshing for the incompressible Stokes problem. *Comp. Methods Appl. Mech. Engrg.*, **294**, 428 – 448, 2015.

[5] Aziz, A. K.: *Mathematical foundations of the finite element method with applications to partial differential equations*. Academic press, Baltimore, MD US, 1972.

[6] Babuška, I.: Error-bounds for finite element method. *Numer. Math.*, **16**, 322 – 333, 1971.

[7] Babuška, I.: The finite element method with Lagrangian multipliers. *Numer. Math.*, **20**(3), 179–192, 1973.

[8] Babuška, I.: The finite element method with penalty. *Math. Comput.*, **27**(122), 221–228, 1973.

[9] Babuška, I.: Courant element: Before and after. In *Finite element methods : fifty years of the Courant element*. (Křižek, M.; et al., Eds.), Lecture notes in pure and applied mathematics, M. Dekker, New York, 1994.

[10] Babuška, I.; Banerjee, U.; Osborn, J. E.: Meshless and generalized finite element methods: A survey of some major results. In *Meshfree Methods for Partial Differential Equations*. (Griebel, M.; Schweitzer, M.A., Eds.), Vol. 26, *Lectures Notes in Computational Science and Engineering*, 2002.

[11] Babuška, I.; Banerjee, U.; Osborn, J. E.: Generalized finite element methods: Main Ideas, results and perspective. *Int. J. Comp. Meth.*, **1**, 67 – 103, 2004.

[12] Babuška, I.; Melenk, J. M.: The partition of unity method. *Internat. J. Numer. Methods Engrg.*, **40**, 727 – 758, 1997.

[13] Babuška, I.; Szabó, B. A.: On the rates of convergence of the finite element method. *Internat. J. Numer. Methods Engrg.*, **18**, 1982.

[14] Barnhill, R. E; Birkhoff, G.; Gordon, W. J.: Smooth interpolation in triangles. *Journal of Approximation Theory*, **8**(2), 114 – 128, 1973.

[15] Bathe, K. J.: *Finite Element Procedures*. Prentice-Hall, Englewood Cliffs, NJ, 1996.

[16] Belytschko, T.; Black, T.: Elastic crack growth in finite elements with minimal remeshing. *Internat. J. Numer. Methods Engrg.*, **45**, 601–620, 1999.

[17] Belytschko, T.; Moës, N.; Usui, S.; Parimi, C.: Arbitrary discontinuities in finite elements. *Internat. J. Numer. Methods Engrg.*, **50**, 993 – 1013, 2001.

[18] Belytschko, T.; Parimi, C.; Moës, N.; Sukumar, N.; Usui, S.: Structured extended finite element methods for solids defined by implicit surfaces. *Internat. J. Numer. Methods Engrg.*, **56**, 609 – 635, 2003.

[19] Bernardi, C.: Optimal finite-element interpolation on curved domains. *SIAM J. Numer. Anal.*, **26**(5), 1212–1240, 1989.

[20] Bittencourt, T. N.; Wawrzynek, P. A.; Ingraffea, A. R.; Sousa, J. L.: Quasi-automatic simulation of crack propagation for 2D LEFM problems. *Eng. Fract. Mech.*, **55**, 321–334, 1996.

[21] Boffi, D.; Brezzi, F.; Fortin, M.: *Mixed Finite Element Methods and Applications*. Springer, 2013.

[22] Boffi, D.; Gastaldi, L.: A finite element approach for the immersed boundary method. *Computers & Structures*, **81**(8-11), 491–501, 2003. K. J. Bathe 60th Anniversary Issue.

[23] Bonfils, N.; Chevaugeon, N.; Moës, N.: Treating volumetric inequality constraint in a continuum media with a coupled X-FEM/level-set strategy. *Comp. Methods Appl. Mech. Engrg.*, **205-208**, 16 – 28, 2012. Special Issue on Advances in Computational Methods in Contact Mechanics dedicated to the memory of Professor J. A. C. Martins.

[24] Braess, D.: *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer Spektrum, Berlin, 5. überarb. aufl. edition, 2013.

[25] Burman, E.: Ghost penalty. *Comptes Rendus Mathematique*, **348**(21), 1217 – 1220, 2010.

[26] Burman, E.; Claus, S.; Hansbo, P.; Larson, M. G.; Massing, A.: CutFEM: Discretizing geometry and partial differential equations. *Internat. J. Numer. Methods Engrg.*, **104**(7), 472–501, 2015.

[27] Burman, E.; Elfverson, D.; Hansbo, P.; Larson, M. G.; Larsson, K.: Shape optimization using the cut finite element method. *ArXiv e-prints*, 2016.

[28] Burman, E.; Hansbo, P.: Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Comp. Methods Appl. Mech. Engrg.*, 2680–2686, 2010.

[29] Burman, E.; Hansbo, P.: Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Applied Numerical Mathematics*, **62**, 328 – 341, 2012.

[30] Buzbee, B. L.; Dorr, F. W.; George, J. A.; Golub, G. H.: The direct solution of the discrete Poisson equation on irregular regions. *SIAM J. Numer. Anal.*, **8**(4), 722–736, 1971.

[31] Cervera, M.; Chiumenti, M.; Codina, R.: Mixed stabilized finite element methods in nonlinear solid mechanics: Part I: Formulation. *Comp. Methods Appl. Mech. Engrg.*, **199**(37-40), 2559 – 2570, 2010.

[32] Chen, Q.; Babuška, I.: Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle. *Comp. Methods Appl. Mech. Engrg.*, **128**(3-4), 405–417, 1995.

[33] Chen, Z.; Xiao, Y.; Zhang, L.: The adaptive immersed interface finite element method for elliptic and Maxwell interface problems. *J. Comput. Phys.*, **228**(14), 5000 – 5019, 2009.

[34] Cheng, K. W.; Fries, T. P.: Higher-order XFEM for curved strong and weak discontinuities. *Internat. J. Numer. Methods Engrg.*, **82**, 564 – 590, 2010.

[35] Chessa, J.; Wang, H.; Belytschko, T.: On the construction of blending elements for local partition of unity enriched finite elements. *Internat. J. Numer. Methods Engrg.*, **57**, 1015–1038, 2003.

[36] Chin, E. B; Lasserre, J. B.; Sukumar, N.: Numerical integration of homogeneous functions on convex and nonconvex polygons and polyhedra. *Comput. Mech.*, **56**(6), 967–981, 2015.

[37] Ciarlet, P. G.: *The finite element method for elliptic problems*. Studies in mathematics and its applications. North-Holland, Amsterdam, New-York, 1980.

[38] Ciarlet, P.G.; Raviart, P.-A.: Interpolation theory over curved elements, with applications to finite element methods. *Computer Methods in Applied Mechanics and Engineering*, **1**(2), 217 – 249, 1972.

[39] Clough, R. W.: Early history of the finite element method from the view point of a pioneer. *Internat. J. Numer. Methods Engrg.*, **60**(1), 283–287, 2004.

[40] Codina, R.; Baiges, J.: Weak imposition of essential boundary conditions in the finite element approximation of elliptic problems with non-matching meshes. *Internat. J. Numer. Methods Engrg.*, **104**(7), 624–654, 2015.

[41] Commend, S.; Truty, A.; Zimmermann, T.: Stabilized finite elements applied to elasto-plasticity: I. Mixed displacement-pressure formulation. *Comp. Methods Appl. Mech. Engrg.*, **193**(33), 3559–3586, 2004.

[42] Coons, S. A.: Surfaces for Computer-aided designt of space forms. Technical report, Cambridge, MA, USA, 1967.

[43] Cosimo, A.; Fachinotti, V.; Cardona, A.: An enrichment scheme for solidification problems. *Comput. Mech.*, **52**(1), 17–35, 2013.

[44] Cottrell, J. A.; Hughes, T. J. R.; Bazilevs, Y.: *Isogeometric Analysis*. John Wiley & Sons, Chichester, 2009.

[45] Courant, R.: Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc*, 1943.

[46] Cowper, G. R.: Gaussian quadrature formulas for triangles. *Internat. J. Numer. Methods Engrg.*, **7**(3), 405–408, 1973.

[47] Dauge, M.; Düster, A.; Rank, E.: Theoretical and numerical investigation of the Finite Cell Method. **65**(3), 1039–1064, 2015.

[48] Daux, C.; Moës, N.; Dolbow, J.; Sukumar, N.; Belytschko, T.: Arbitrary branched and intersecting cracks with the extended finite element method. *Internat. J. Numer. Methods Engrg.*, **48**, 1741 – 1760, 2000.

[49] Davis, P. J.; Rabinowitz, P.: *Methods of Numerical Integration*. Dover Publications, second edition edition, 10 2007.

[50] de Souza Neto, E. A.; Perić, D.; Owen, D. R. J.: *Computational Methods for Plasticity: Theory and Applications*. Wiley, 1 edition, 12 2008.

[51] Deuflhard, P.: *Numerische Mathematik: Eine algorithmisch orientierte Einführung*. Walter de Gruyter & Co., 4 edition, 7 2008.

[52] Deuflhard, P.; Weiser, M.: *Adaptive Numerical Solution of PDEs*. Walter de Gruyter & Co., Hawthorne, NJ, USA, 2012.

[53] Dolbow, J.; Moës, N.; Belytschko, T.: Discontinuous enrichment in finite elements with a partition of unity method. *Finite Elem. Anal. Des.*, **36**, 235–260, 2000.

[54] Dréau, K.; Chevaugeon, N.; Moës, N.: Studied X-FEM enrichment to handle material interfaces with higher order finite element. *Comp. Methods Appl. Mech. Engrg.*, **199**, 1922–1936, 2010.

[55] Duczek, S.; Liefold, S.; Gabbert, U.: The finite and spectral cell methods for smart structure applications: transient analysis. *Acta Mechanica*, **226**(3), 845–869, 2015.

[56] Düster, A.; Niggl, A.; Nübel, V.; Rank, E.: A numerical investigation of high-order finite elements for problems of elastoplasticity. **17**(1-4), 397–404, 2002.

[57] Düster, A.; Rank, E.: The p-version of the finite element method compared to an adaptive h-version for the deformation theory of plasticity. *Comp. Methods Appl. Mech. Engrg.*, **190**(15-17), 1925–1935, 2001.

[58] Ebobisse, F.; Reddy, B. D.: Some mathematical problems in perfect plasticity. *Comp. Methods Appl. Mech. Engrg.*, **193**(48-51), 5071 – 5094, 2004.

[59] Elgeti, S.; Sauerland, H.: Deforming fluid domains within the finite element method: five mesh-based tracking methods in comparison. *Archive Comp. Mech. Engrg.*, **23**(2), 323–361, 2016.

[60] Elgeti, S.; Sauerland, H.; Pauli, L.; Behr, M.: On the usage of NURBS as interface representation in free-surface flows. *Int. J. Numer. Methods Fluids*, **69**(1), 73–87, 2012.

[61] Ern, A.; Guermond, J.-L.: *Theory and Practice of Finite Elements*. Springer, Berlin, 2004.

[62] Fernández-Méndez, S.; Huerta, A.: Imposing essential boundary conditions in mesh-free methods. *Comp. Methods Appl. Mech. Engrg.*, **193**, 1257 – 1275, 2004.

[63] Fidkowski, K. J.; Darmofal, D. L.: An adaptive simplex cut-cell method for discontinuous Galerkin discretizations of the Navier-Stokes equations. *Collection of Technical Papers - 18th AIAA Computational Fluid Dynamics Conference*, 2007–3941, 2007.

[64] Fidkowski, K. J.; Darmofal, D. L.: A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations. *J. Comput. Phys.*, **225**(2), 1653–1672, 2007.

[65] Fleming, M.; Chu, Y. A.; Moran, B.; Belytschko, T.: Enriched element-free Galerkin methods for crack tip fields. *Internat. J. Numer. Methods Engrg.*, **40**, 1483 – 1504, 1997.

[66] Frei, S.; Richter, T.: A locally modified parametric finite element method for interface problems. *SIAM J. Numer. Anal.*, **52**(5), 2315–2334, 2014.

[67] Frey, P. J.; George, P.-L.: *Mesh Generation: Application to Finite Elements*. ISTE, 2007.

[68] Fried, I.: Condition of finite element matrices generated from nonuniform meshes. *AIAA Journal*, **10**, 219 – 221, 1972.

[69] Fries, T.-P.: A corrected XFEM approximation without problems in blending elements. *Internat. J. Numer. Methods Engrg.*, **75**, 503 – 532, 2008.

[70] Fries, T. P.: Higher-order accurate integration for cut elements with Chen-Babuška nodes. In *Advances in Discretization Methods: Discontinuities, virtual elements, fictitious domain methods*. (Ventura, G.; Benvenuti, E., Eds.), Vol. 12, *SEMA SIMAI Springer Series*, Springer, Berlin, 245–269, 2016.

[71] Fries, T.-P.; Belytschko, T.: The intrinsic XFEM: A method for arbitrary discontinuities without additional unknowns. *Internat. J. Numer. Methods Engrg.*, **68**, 1358 – 1385, 2006.

[72] Fries, T.-P.; Belytschko, T.: The extended/generalized finite element method: An overview of the method and its applications. *Internat. J. Numer. Methods Engrg.*, **84**, 253 – 304, 2010.

[73] Fries, T.-P.; Omerović, S.: Higher-order accurate integration of implicit geometries. *Internat. J. Numer. Methods Engrg.*, **106**(5), 323–371, 2015.

[74] Fries, T.-P.; Omerović, S.; Schöllhammer, D.; Steidl, J.: Higher-order meshing of implicit geometries-Part I: Integration and interpolation in cut elements. *Comp. Methods Appl. Mech. Engrg.*, **313**, 759 – 784, 2017.

[75] Fries, T.-P.; Schöllhammer, D.: Higher-order meshing of implicit geometries-Part II: Approximations on manifolds. *Comp. Methods Appl. Mech. Engrg.*, to appear.

[76] Ganesan, S.; Matthies, G.; Tobiska, L.: On spurious velocities in incompressible flow problems with interfaces. *Comp. Methods Appl. Mech. Engrg.*, **196**, 1193 – 1202, 2007.

[77] Gekeler, E. W.: *Mathematical Methods for Mechanics: A Handbook with MATLAB Experiments*. Springer, 12 2009.

[78] Gerstenberger, A.; Wall, W. A.: An eXtended finite element method/Lagrange multiplier based approach for fluid-structure interaction. *Comp. Methods Appl. Mech. Engrg.*, **197**, 1699 – 1714, 2008.

[79] Gerstenberger, A.; Wall, W. A.: An embedded Dirichlet formulation for 3D continua. *Internat. J. Numer. Methods Engrg.*, **82**, 537 – 563, 2010.

[80] Glowinski, R.: Finite element methods for incompressible viscous flow. *Numerical Methods for Fluids (Part 3)*, Vol. 9, *Handbook of Numerical Analysis*, Elsevier, 3 – 1176, 2003.

[81] Glowinski, R.; Pan, T. W.; Périaux, J.: A Fictitious Domain Method for Dirichlet Problems and Applications. *Comp. Methods Appl. Mech. Engrg.*, **111**, 283 – 303, 1994.

[82] Gockenbach, M. S.: *Understanding and implementing the finite element method*. SIAM, 2006.

[83] Gong, Y.; Li, B.; Li, Z.: Immersed-interface finite-element methods for elliptic interface problems with nonhomogeneous jump conditions. *SIAM J. Numer. Anal.*, **46**(1), 472–495, 2008.

[84] Goodier, J. N.: Concentration of stress around spherical and cylindrical inclusions and flaws. *J. Appl. Mech., ASME*, **55**, 39–44, 1933.

[85] Gordon, W. J.; Hall, C. A.: Construction of curvilinear co-ordinate systems and applications to mesh generation. *Internat. J. Numer. Methods Engrg.*, **7**, 461 – 477, 1973.

[86] Gordon, W. J.; Hall, C. A.: Transfinite element methods: blending function interpolation over arbitrary curved element domains. *Numer. Math.*, **21**, 109 – 129, 1973.

[87] Groß, S.; Reusken, A.: An extended pressure finite element space for two-phase incompressible flows with surface tension. *J. Comput. Phys.*, **224**, 40 – 58, 2007.

[88] Groß, S.; Reusken, A.: *Numerical methods for two-phase incompressible flows*. Springer Series in Computational Mathematics. Berlin, 2011.

[89] Gruber, P.: *Fast solvers and adaptive high-order FEM in elastoplasticity*. PhD thesis, Johannes Kepler Universität Linz, 2011.

[90] Hammer, P. C.; Marlowe, O. J.; Stroud, A. H.: Numerical integration over simplexes and cones. *Math. Tables Aids Comput.*, **10**, 130–137, 1956.

[91] Hansbo, A.; Hansbo, P.: An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Comp. Methods Appl. Mech. Engrg.*, **191**, 5537 – 5552, 2002.

[92] Hansbo, P.: Nitsche's method for interface problems in computational mechanics. *GAMM-Mitteilungen*, **28**(2), 183–206, 2005.

[93] Haslinger, J.; Maitre, J.-F.; Tomas, L.: Fictitious domain methods with distributed Lagrage multipliers part I: Application to the solution of elliptic state problems. *Math. Models Methods Appl. Sci.*, **11**(03), 521–547, 2001.

[94] Haslinger, J.; Renard, Y.: A new fictitious domain approach inspired by the Extended Finite Element Method. *SIAM J. Numer. Anal.*, **47**(2), 1474–1499, 2009.

[95] Heinze, S.; Joulaian, M.; Düster, A.: Numerical homogenization of hybrid metal foams using the finite cell method. *Computers & Mathematics with Applications*, **70**(7), 1501 – 1517, 2015. High-Order Finite Element and Isogeometric Methods.

[96] Hill, R.: *The Mathematical Theory of Plasticity*. Oxford University Press, 11 1998.

[97] Holdych, D. J.; Noble, D. R.; Secor, R. B.: Quadrature rules for triangular and tetrahedral elements with generalized functions. *Internat. J. Numer. Methods Engrg.*, **73**, 1310 – 1327, 2008.

[98] Höllig, K.; Reif, U.; Wipper, J.: Weighted Extended B-Spline Approximation of Dirichlet Problems. *SIAM J. Numer. Anal.*, **39**(2), 442–462, 2001.

[99] Holzer, S. M.; Yosibash, Z.: The p-version of the finite element method in incremental elasto-plastic analysis. *Internat. J. Numer. Methods Engrg.*, **39**(11), 1859–1878, 1996.

[100] Hughes, T. J. R.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[101] Hurwitz, A.; Courant, R.: *Vorlesungen über Allgemeine Funktionentheorie und Elliptische Funktionen*. Springer, Berlin, 1922. Erste Auflage.

[102] Ilinca, F.; Hétu, J.-F.: A finite element immersed boundary method for fluid flow around rigid objects. *Int. J. Numer. Methods Fluids*, **65**(7), 856–875, 2011.

[103] Irons, B. M.; Zienkiewicz, O. C.: *The isoparametric finite element system: A new concept in finite element analysis*. Royal Aeronautical Society, 1969.

[104] Jeremić, B.; Xenophontos, C.: Application of the p-version of the finite element method to elastoplasticity with localization of deformation. *Commun. Numer. Meth. Engrg.*, **15**(12), 867–876, 1999.

[105] Joulaian, M.; Hubrich, S.; Düster, A.: Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Comput. Mech.*, **57**(6), 979–999, 2016.

[106] Jr, F. Williamson: An historical note on the finite element method. *Internat. J. Numer. Methods Engrg.*, **15**(6), 930–934, 1980.

[107] Kramer, R. M. J.; Noble, D. R.: A conformal decomposition finite element method for arbitrary discontinuities on moving interfaces. *Internat. J. Numer. Methods Engrg.*, **100**(2), 87–110, 2014.

[108] Krivoshapko, S.; Ivanov, V. N.: *Encyclopedia of Analytical Surfaces*. Springer, 2015 edition, 2 2015.

[109] Krommer, A. R.: *Numerical Integration on Advanced Computer Systems*. Lecture Notes in Computer Science 848. Springer-Verlag Berlin Heidelberg, Berlin, 1 edition, 1994.

[110] Kudela, L.; Zander, N.; Kollmannsberger, S.; Rank, E.: Smart octrees: Accurately integrating discontinuous functions in 3D. *Comp. Methods Appl. Mech. Engrg.*, **306**, 406 – 426, 2016.

[111] Kurrer, K.-E.: *Geschichte der Baustatik: Auf der Suche nach dem Gleichgewicht*. Wilhelm Ernst & Sohn, 2016.

[112] Lasserre, J. B.: Integration and homogeneous functions. *Proceedings of the American Mathematical Society*, **127**(3), 813–818, 1999.

[113] Lehrenfeld, C.: High order unfitted finite element methods on level set domains using isoparametric mappings. *Comp. Methods Appl. Mech. Engrg.*, **300**, 716 – 733, 2016.

[114] Lehrenfeld, C.: A higher order isoparametric fictitious domain method for level set domains. *ArXiv e-prints*, 2017.

[115] Lenoir, M.: Optimal Isoparametric Finite Elements and Error Estimates for Domains Involving Curved Boundaries. *Society of Industrial and Applied Mathematics*, **23**, 562 – 580, 1986.

[116] Li, Z.: A Fast Iterative Algorithm for Elliptic Interface Problems. *SIAM J. Numer. Anal.*, **35**(1), 230–254, 1998.

[117] Li, Z.; Lin, T.; Wu, X.: New Cartesian grid methods for interface problems using the finite element formulation. *Numer. Math.*, **96**(1), 61–98, 2003.

[118] Löhner, R.: *Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods*. John Wiley & Sons, Chichester, 2008.

[119] Lorensen, W. W.; Cline, H. E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Comput. Graph.*, **21**, 1987.

[120] Lubliner, J.: On the thermodynamic foundations of non-linear solid mechanics. *Int. J. Non Linear Mech.*, **7**(3), 237 – 254, 1972.

[121] Lubliner, J.: *Plasticity Theory*. Dover Publications Inc., 2008.

[122] Luenberger, D. G.; Ye, Y.: *Linear and Nonlinear Programming*. Springer, 4th ed. 2016 edition, 6 2015.

[123] Massing, A.; Larson, M. G.; Logg, A.: Efficient implementation of finite element methods on nonmatching and overlapping meshes in three dimensions. *SIAMGM J. Comput.*, **35**(1), C23–C47, 2013.

[124] Massjung, R.: An unfitted Discontinuous Galerkin Method applied to elliptic interface problems. *SIAM J. Numer. Anal.*, **50**(6), 3134–3162, 2012.

[125] Melenk, J. M.; Babuška, I.: The partition of unity finite element method: basic theory and applications. *Comp. Methods Appl. Mech. Engrg.*, **139**, 289 – 314, 1996.

[126] Moës, N.; Cloirec, M.; Cartraud, P.; Remacle, J. F.: A computational approach to handle complex microstructure geometries. *Comp. Methods Appl. Mech. Engrg.*, **192**, 3163–3177, 2003.

[127] Moës, N.; Dolbow, J.; Belytschko, T.: A finite element method for crack growth without remeshing. *Internat. J. Numer. Methods Engrg.*, **46**, 131 – 150, 1999.

[128] Mousavi, S. E.; Sukumar, N.: Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. *Comput. Mech.*, **47**, 535 – 554, 2011.

[129] Müller, B.; Kummer, F.; Oberlack, M.: Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *Internat. J. Numer. Methods Engrg.*, **96**, 512 – 528, 2013.

[130] Natarajan, S.; Bordas, S.; Mahapatra, D. R.: Polygonal Finite Element Method using Schwarz-Christoffel mapping. *Internat. J. Numer. Methods Engrg.*, **80**(1), 103–134, 2009.

[131] Natarajan, S.; Mahapatra, D. R.; Bordas, S. P. A: Integrating strong and weak discontinuities without integration subcells and example applications in an XFEM/GFEM framework. *Internat. J. Numer. Methods Engrg.*, **83**(3), 269–294, 2010.

[132] Nitsche, J. A.: Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, Vol. 36, Springer, Berlin, 9 – 15, 1971.

[133] Noble, D. R.; Newren, E. P.; Lechman, J. B.: A conformal decomposition finite element method for modeling stationary fluid interface problems. *Int. J. Numer. Methods Fluids*, **63**(6), 725–742, 2010.

[134] Nübel, V.; Düster, A.; Rank, E.: An *rp*-adaptive finite element method for the deformation theory of plasticity. **39**(5), 557–574, 2007.

[135] Oden, J. T.: Finite elements: An introduction. *Finite Element Methods (Part 1)*, Vol. 2, *Handbook of Numerical Analysis*, Elsevier, 3 – 15, 1991.

[136] Omerović, S.; Fries, T.-P.: Conformal higher-order remeshing schemes for implicitly defined interface problems. *Internat. J. Numer. Methods Engrg.*, **109**(6), 763–789, 2017.

[137] Ortega, J. M.; Rheinboldt, W. C.: *Iterative solution of nonlinear equations in several variables*. Computer Science and Applied Mathematics. Academic Press, New York, NY, 1970.

[138] Ortiz, M.; Quigley, J. J.: Adaptive mesh refinement in strain localization problems. *Comp. Methods Appl. Mech. Engrg.*, **90**(1), 781 – 804, 1991.

[139] Osher, S.; Fedkiw, R. P.: Level set methods: an overview and some recent results. *J. Comput. Phys.*, **169**, 463 – 502, 2001.

[140] Osher, S.; Sethian, J. A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, **79**, 12 – 49, 1988.

[141] Parvizian, J.; Düster; Rank, E.: Topology optimization using the finite cell method. *Optimization and Engineering*, **13**(1), 57–78, 2012.

[142] Parvizian, J.; Düster, A.; Rank, E.: Finite cell method : h- and p-extension for embedded domain problems in solid mechanics. *Comput. Mech.*, **41**, 121 – 133, 2007.

[143] Perić, D.; Hochard, C.; Dutko, M.; Owen, D. R. J.: Transfer operators for evolving meshes in small strain elasto-plasticity. *Comp. Methods Appl. Mech. Engrg.*, **137**(3), 331 – 344, 1996.

[144] Peskin, C. S.: *Flow patterns around heart valves: a digital computer method for solving the equations of motion*. Phd thesis, Albert Einstein College of Medicine, Yeshiva University, 1972.

[145] Peskin, C. S.: Flow patterns around heart valves: A numerical method. *J. Comput. Phys.*, **10**(2), 252 – 271, 1972.

[146] Pinsky, P. M.: A finite element formulation for elastoplasticity based on a three-field variational equation. *Comp. Methods Appl. Mech. Engrg.*, **61**(1), 41–60, 1987.

[147] Rangarajan, R.; Lew, A.: Universal meshes: A method for triangulating planar curved domains immersed in nonconforming meshes. *Internat. J. Numer. Methods Engrg.*, **98**(4), 236 – 264, 2014.

[148] Rüberg, T.; Cirak, F.: Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. *Comp. Methods Appl. Mech. Engrg.*, **209 - 212**, 266 – 283, 2012.

[149] Schillinger, D.; Ruess, M.; Zander, N.; Bazilevs, Y.; Düster, A.; Rank, E.: Small and large deformation analysis with the p- and B-spline versions of the Finite Cell Method. *Comput. Mech.*, **50**, 445 – 478, 2012.

[150] Scott, L. R.: *Finite element techniques for curved boundaries*. PhD thesis, Massachusetts Institute of Technology, 1973.

[151] Sethian, J. A.: *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, 2 edition, 1999.

[152] Sethian, J. A.: Evolution, implementation and application of level set and fast marching methods for advancing fronts. *J. Comput. Phys.*, **169**, 503 – 555, 2001.

[153] Simo, J. C.; Hughes, T. J. R.: *Computational Inelasticity*. Springer, corrected edition, 7 2000.

[154] Simo, J. C.; Kennedy, J. G.; Taylor, R. L.: Complementary mixed finite element formulations for elastoplasticity. *Comp. Methods Appl. Mech. Engrg.*, **74**(2), 177 – 206, 1989.

[155] Simo, J. C.; Oliver, J.; Armero, F.: An analysis of strong discontinuities induced by strain-softening in rate-independent inelastic solids. *Comput. Mech.*, **12**(5), 277–296, 1993.

[156] Śolín, P.: *Partial Differential Equations and the Finite Element Method*. John Wiley & Sons, Inc., 2005.

[157] Šolín, P.; Segeth, K.; Doležel, I.: *Higher-Order Finite Element Methods*. CRC Press, Boca Raton, FL, 2003.

[158] Stein, E.: *The History of Theoretical, Material and Computational Mechanics - Mathematics Meets Mechanics and Engineering*, Chapter History of the Finite Element Method – Mathematics Meets Mechanics – Part I: Engineering Developments, 399–442. Springer, Berlin, Heidelberg, 2014.

[159] Stein, E.; Ramm, E.: *Error-controlled Adaptive Finite Elements in Solid Mechanics*. John Wiley & Sons, 2003.

[160] Stolarska, M.; Chopp, D. L.; Moës, N.; Belytschko, T.: Modelling crack growth by level sets in the extended finite element method. *Internat. J. Numer. Methods Engrg.*, **51**, 943 – 960, 2001.

[161] Strang, G.; Fix, G.: *An Analysis of the Finite Element Method*. Wellesley-Cambridge Press, 2008.

[162] Strouboulis, T.; Babuška, I.; Copps, K.: The design and analysis of the generalized finite element method. *Comp. Methods Appl. Mech. Engrg.*, **181**, 43 – 69, 2000.

[163] Stroud, A. J.: *Approximate Calculation of Multiple Integrals*. Prentice-Hall, Inc., 1971.

[164] Sudhakar, Y.; Wall, W. A.: Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods. *Comp. Methods Appl. Mech. Engrg.*, **258**, 39 – 54, 2013.

[165] Sukumar, N.; Chopp, D. L.; Moës, N.; Belytschko, T.: Modeling holes and inclusions by level sets in the extended finite-element method. *Comp. Methods Appl. Mech. Engrg.*, **190**, 6183 – 6200, 2001.

[166] Suri, M.: On the robustness of the h- and p-versions of the finite-element method. *J. Comput. Appl. Math.*, **35**(1-3), 303 – 310, 1991.

[167] Szabó, B. A.; Actis, R. L.; Holzer, S. M.: *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, Chapter Solution of Elastic-Plastic Stress Analysis Problems by the p-Version of the Finite Element Method, 395–416. Springer, 1995.

[168] Szabó, B. A.; Babuška, I.: *Finite Element Analysis*. John Wiley & Sons, Chichester, 1991.

[169] Szabó, B. A.; Düster, A.; Rank, E.: *The p-Version of the Finite Element Method*, Chapter 5, 119 – 139. John Wiley & Sons, Chichester, 2004.

[170] Thompson, J. F.: The national grid project. *Computing Systems in Engineering*, **3**(1), 393 – 399, 1992.

[171] Trangenstein, J. A.: *Numerical Solution of Elliptic and Parabolic Partial Differential Equations*. Cambridge University Press, 2013. Cambridge Books Online.

[172] Überhuber, C.: *Computer-Numerik 1*. Springer, 1995 edition, 10 2008.

[173] Überhuber, C.: *Computer-Numerik 2*. Springer, 1995 edition, 6 2008.

[174] Varduhn, V.; Hsu, M.-C.; Ruess, M.; Schillinger, D.: The tetrahedral finite cell method: Higher-order immersogeometric analysis on adaptive non-boundary-fitted meshes. *Internat. J. Numer. Methods Engrg.*, **107**(12), 1054–1079, 2016.

[175] Ventura, G.: On the elimination of quadrature subcells for discontinuous functions in the eXtended finite element method. *Internat. J. Numer. Methods Engrg.*, **66**, 761 – 795, 2006.

[176] Wang, F.; Xiao, Y.; Xu, J.: High-order extended finite element methods for solving interface problems. *ArXiv e-prints*, April 2016.

[177] Wriggers, P.: *Nonlinear Finite Element Methods*. Springer, 2008 edition, 11 2008.

[178] Zhang, L.; Gerstenberger, A.; Wang, X.; Liu, W. K.: Immersed finite element method. *Comp. Methods Appl. Mech. Engrg.*, **193**(21-22), 2051–2067, 2004.

[179] Zienkiewicz, O. C.: Origins, milestones and directions of the finite element method - A personal view. *Finite Element Methods (Part 2), Numerical Methods for Solids (Part 2)*, Vol. 4, *Handbook of Numerical Analysis*, Elsevier, 3 – 67, 1996.

[180] Zienkiewicz, O. C.; Taylor, R. L.; Zhu, J. Z.: *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, 7 edition, 2013.

[181] Zilian, A.; Legay, A.: The enriched space-time finite element method (EST) for simultaneous solution of fluid-structure interaction. *Internat. J. Numer. Methods Engrg.*, **75**, 305–334, 2008.

[182] Zulehner, W.: *Numerische Mathematik: Eine Einführung anhand von Differentialgleichungsproblemen; Band 1: Stationäre Probleme*. Birkhäuser, 2008 edition, 12 2007.