

On a Space-Time Extended Finite Element Method for the Solution of a Class of Two-Phase Mass Transport Problems

Von der Fakultät für Mathematik, Informatik und
Naturwissenschaften der RWTH Aachen University zur
Erlangung des akademischen Grades eines Doktors der
Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Ingenieur Christoph Lehrenfeld
aus Viersen

Berichter: Univ.-Prof. Dr. rer.nat. Arnold Reusken
Univ.-Prof. Marek Behr, Ph.D.
Univ.-Prof. Dr. techn. Joachim Schöberl

Tag der mündlichen Prüfung: 4. Februar 2015

Diese Dissertation ist auf den Internetseiten der
Hochschulbibliothek online verfügbar.

Acknowledgements

The present thesis originates from my work at the IGPM (Institut für Geometrie und Praktische Mathematik) at the RWTH Aachen University and was financially supported by the DFG (German Research Foundation) through the Priority Program SPP 1506 “Transport Processes at Fluidic Interfaces”.

I would like to express my gratitude to my advisor Prof. Dr. Arnold Reusken for supervising my work and his support throughout my doctoral studies. His advice and feedback has always been very helpful and is greatly appreciated. I also want to express my gratitude to my co-advisor, Prof. Marek Behr, Ph.D., for his efforts related to this thesis.

I’m very grateful to my colleagues at the LNM (Lehrstuhl für Numerische Mathematik) for nice lunch breaks, discussions on and off topic and legendary social events. Many thanks to Jens Berger, Patrick Esser, Jörg Grande, Sven Groß, Eva Loch, Igor Voulis, Yuanjun Zhang and especially to the best office-mate I can imagine, Liang Zhang. I also appreciate the lunch and dinner meetings with Markus Bachmayr, Patrick Esser, Angela Klewinghaus and Marcel Makowski. It has always been fun spending time with them.

I would also like to thank Prof. Dr. Joachim Schöberl for introducing me to the exciting field of numerical mathematics. I enjoyed the years as student, student worker, diploma student and the months as assistant in Vienna very much. I am also grateful that he declared himself willing to be a third reporter for this thesis.

The collaboration and scientific exchange within the interdisciplinary group of the DFG Priority Program SPP 1506 was very fruitful and I want to thank my colleagues for a nice atmosphere and good collaboration, especially Christoph Meyer, Stephan Weller, Carlos Falconi, Matthias Waidmann, Holger Marschall and Sebastian Aland.

I want to thank my friends and my family for offering a nice compensation for math or programming problems. At last, I wish to sincerely thank my wife Anke for her continuous support and open ears.

Abstract

In the present thesis a new numerical method for the simulation of mass transport in an incompressible immiscible two-phase flow system is presented. The mathematical model consists of convection diffusion equations on moving domains which are coupled through interface conditions. One of those conditions, the Henry interface condition, prescribes a jump discontinuity of the solution across the moving interface. For the description of the interface position and its evolution we consider interface capturing methods, for instance the level set method. In those methods the mesh is not aligned to the evolving interface such that the interface intersects mesh elements. Hence, the moving discontinuity is located within individual elements which makes the numerical treatment challenging.

The discretization presented in this thesis is based on essentially three core components. The first component is an enrichment with an extended finite element (XFEM) space which provides the possibility to approximate discontinuous quantities accurately without the need for aligned meshes. This enrichment, however, does not respect the Henry interface condition. The second component cures this issue by imposing the interface condition in a weak sense into the discrete variational formulation of the finite element method. To this end a variant of the Nitsche technique is applied. For a stationary interface the combination of both techniques offers a good way to provide a reliable method for the simulation of mass transport in two-phase flows. However, the most difficult aspect of the problem is the fact that the interface is typically not stationary, but moving in time. The numerical treatment of the moving discontinuity requires special care. For this purpose a space-time variational formulation, the third core component of this thesis, is introduced and combined with the first two components: the XFEM enrichment and the Nitsche technique. In this thesis we present the components and the resulting methods one after another, for stationary and non-stationary interfaces. We analyze the methods with respect to accuracy and stability and discuss important properties.

For the case of a stationary interface the combination of an XFEM enrichment and the Nitsche technique, the Nitsche-XFEM method, has been introduced by other authors. Their method, however, lacks stability in case of dominating convection. We combine the Nitsche-XFEM method with the Streamline Diffusion technique to provide a stable method also for convection dominated problems. We further discuss the conditioning of

the linear systems arising from Nitsche-XFEM discretizations which can be extremely ill-conditioned.

For the case of a moving interface we propose a space-time Galerkin formulation with trial and test functions which are discontinuous in time and combine this approach with an XFEM enrichment and a Nitsche technique resulting in the Space-Time-DG Nitsche-XFEM method. This method is new. We present an error analysis and discuss implementation aspects like the numerical integration on arising space-time geometries.

The aforementioned methods have been implemented in the software packages **DROPS** for the spatially three-dimensional case. The correctness of the implementation and the accuracy of the method is analyzed for test cases. Finally, we consider the coupled simulation of mass transport and fluid dynamics for realistic scenarios.

Zusammenfassung

In der vorliegenden Arbeit wird eine neue numerische Methode für die Simulation von Stofftransportprozessen in inkompressiblen unvermischbaren Zweiphasen-Strömungen vorgestellt. Das mathematische Modell für den Stofftransport besteht hierbei aus Konvektions-Diffusions-Gleichungen auf sich bewegenden Gebieten, welche durch Übergangsbedingungen an der Phasengrenze miteinander gekoppelt sind. Eine dieser Bedingungen, die Henry-Bedingung, schreibt eine Sprung-Unstetigkeit der Lösung über die Phasengrenze vor. Zur Beschreibung der Phasengrenze und ihrer Entwicklung in der Zeit werden so genannte “interface capturing” Methoden betrachtet, zum Beispiel die Levelset-Methode. Bei diesen Methoden ist das Rechengitter nicht an die sich bewegende Phasengrenze angepasst, sodass die Phasengrenze einzelne Elemente schneidet. Unstetigkeiten bewegen sich also innerhalb einzelner Elemente, was die numerische Behandlung anspruchsvoll macht.

Die in dieser Arbeit vorgestellte Methode basiert auf drei Kernkomponenten. Die erste Komponente ist ein erweiterter Finite-Elemente-Raum (XFEM). Dieser ermöglicht es Unstetigkeiten genau darzustellen ohne auf Gitter angewiesen zu sein, welche an die Phasengrenze angepasst sind. Die hierfür verwendete Anreicherung berücksichtigt jedoch nicht die Sprung-Bedingung an der Phasengrenze. Die zweite Komponente der Methode, eine Variante der Nitsche-Methode, löst dieses Problem. Die Bedingung an der Phasengrenze wird hierbei im Rahmen der diskreten Variationsformulierung der Finite-Elemente-Methode in einem schwachen Sinn gefordert. Für eine stationäre Phasengrenze bietet die Kombination beider Techniken eine gute Herangehensweise, um eine zuverlässige Methode zur Simulation des Stofftransports in Zweiphasenströmungen zu erhalten. Der schwierigste Aspekt an dem betrachteten Problem ist jedoch die Tatsache, dass die Phasengrenze in der Regel nicht stationär ist, sondern sich mit der Zeit bewegt. Die numerische Behandlung von sich bewegenden Unstetigkeiten erfordert besondere Sorgfalt. Zu diesem Zweck wird eine Variationsformulierung in Raum-Zeit eingeführt, welche die dritte Kernkomponente unserer Methode darstellt. Die Kombination mit den ersten beiden Komponenten führt zu einer robusten Methode mit hoher Genauigkeit.

Für den Fall einer stationären Phasengrenze wurde die Kombination aus XFEM-Anreicherung und Nitsche-Methode, die Nitsche-XFEM-Methode, bereits von anderen Autoren eingeführt. Diese Methode ist jedoch nicht stabil im konvektions-dominierten Fall. Um auch im Fall dominierender Konvektion eine stabile Methode zu erhalten, wird

die Nitsche-XFEM-Methode mit der Streamline-Diffusion-Technik kombiniert. Darüber hinaus wird die Kondition linearer Gleichungssysteme diskutiert, die aus Diskretisierungen mit der Nitsche-XFEM-Methode entstehen und extrem schlecht konditioniert sein können. Für den Fall einer sich bewegenden Phasengrenze wird eine Galerkin-Formulierung in Raum-Zeit eingeführt, bei der Ansatz- und Testfunktionen verwendet werden, die unstetig in der Zeit sind. Diese Formulierung wird mit einer XFEM-Anreicherung und der Nitsche-Methode kombiniert und resultiert in der Space-Time-DG-Nitsche-XFEM-Methode. Die Methode wird mitsamt einer zugehörigen Fehleranalyse vorgestellt und Implementierungsaspekte werden diskutiert.

Die behandelten Methoden wurden in dem Softwarepaket **DROPS** für den dreidimensionalen Fall implementiert. Die Korrektheit der Implementierung und die Genauigkeit des Verfahrens wird anhand von Testproblemen analysiert. Abschließend wird die gekoppelte Simulation des Stofftransports und der Fluidodynamik für ein realistisches Szenario betrachtet.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Mass transport model	3
1.2.1	Balance laws	3
1.2.2	Mathematical model	4
1.2.3	A reformulation	6
1.2.4	Eulerian description	6
1.3	Numerical challenges	7
1.4	Outline of the thesis	8
2	Mass transport through a stationary interface	9
2.1	Problem description	9
2.1.1	Simplified problems	10
2.1.2	Weak formulation	11
2.2	Discretization with Nitsche-XFEM	13
2.2.1	Approximation of discontinuous quantities (XFEM)	14
2.2.2	Imposing interface conditions (Nitsche)	18
2.2.3	Variants of and alternatives to the Nitsche formulation	22
2.2.4	The Nitsche-XFEM method with small convection	30
2.2.5	The Nitsche-XFEM method with dominating convection	31
2.2.6	Time discretization for a stationary interface	33
2.2.7	Conservation properties of the Nitsche-XFEM formulation	35
2.3	Error analysis	36
2.3.1	Error analysis for Nitsche-XFEM (diffusion dominates)	36
2.3.2	Error analysis for SD-Nitsche-XFEM (convection dominates)	47
2.4	Preconditioning of linear systems	55
2.4.1	Basis transformation	57
2.4.2	Preliminaries	58
2.4.3	Stable subspace splittings of V_h^F	60
2.4.4	Optimal preconditioners based on approximate subspace corrections	67
2.4.5	Diagonal preconditioner on the XFEM subspace	69
2.4.6	Extension of results	70

2.5 Numerical examples	72
2.5.1 Elliptic interface problem: The disk problem	73
2.5.2 Elliptic interface problem: The starfish problem	76
2.5.3 Elliptic interface problem: Conditioning	78
2.5.4 Stationary, convection-dominated problem	86
2.5.5 Transient convection-dominated problem	89
3 Mass transport through a moving interface	95
3.1 Problem description	96
3.1.1 Weak formulation	96
3.1.2 Solution strategies	98
3.2 Discretization with the Space-Time-DG Nitsche-XFEM method	102
3.2.1 Space-time notation	102
3.2.2 Space-time DG formulation for a parabolic model problem	103
3.2.3 Space-time extended finite elements	105
3.2.4 Nitsche formulation for interface conditions in space-time	107
3.3 Error analysis of the Space-Time-DG Nitsche-XFEM method	114
3.3.1 Regularity statements and assumptions	114
3.3.2 Interpolation in space-time	116
3.3.3 Error analysis for the mass transport problem with moving interface	124
3.4 Numerical examples	130
3.4.1 Moving plane	130
3.4.2 Moving sphere	135
3.4.3 Deforming bubble in a vortex	136
3.5 Preconditioning of the Space-Time-DG Nitsche-XFEM method	138
3.5.1 Preliminaries	139
3.5.2 Diagonal preconditioning	140
3.5.3 Block preconditioning	142
3.5.4 Diagonal preconditioning for XFEM block	144
3.5.5 Preconditioning for space-time finite element block	144
3.5.6 A new preconditioner for the Space-Time-DG Nitsche-XFEM method	146
3.5.7 Discussion of results	147
4 Numerical integration on implicitly defined domains	149
4.1 Approximation of implicitly defined domains	150
4.1.1 Approximation of implicit space domains	151
4.1.2 Approximation of implicit space-time domains	151
4.1.3 Remarks on piecewise planar approximation	152
4.2 Integral types	153
4.2.1 Stationary interface	153
4.2.2 Space-time interface	154
4.2.3 Summary of cases	155
4.3 A strategy to decompose intersected 3-simplices or 3-prisms into simplices	156
4.3.1 2D stationary case	156

4.3.2	3D stationary case	156
4.3.3	(2+1)D space-time case	158
4.4	A strategy to decompose intersected 4-prisms into pentatopes	159
4.4.1	Definition of simple geometries in four dimensions	160
4.4.2	Decomposition of a 4-prism into four pentatopes	161
4.4.3	Decomposing the reference hypertriangle	162
4.4.4	Decomposition of a pentatope intersected by the space-time interface into uncut pentatopes	164
4.5	Details of the numerical integration for Space-Time-DG Nitsche-XFEM . .	166
4.5.1	Quadrature on 4D simplices (pentatopes)	166
4.5.2	Computation of ν	168
5	Two-phase flow simulations with mass transport	171
5.1	Model for fluid dynamics in two-phase flows	172
5.1.1	Two-phase Navier-Stokes model	172
5.1.2	Model for the evolution of the interface	173
5.1.3	Two-phase flows model	175
5.2	Numerical methods for solving two-phase flow problems (DROPS)	176
5.2.1	Discretization of the level set equation	176
5.2.2	XFEM discretization for two-phase Navier-Stokes problems	177
5.3	Benchmark problem with complex two-phase fluid dynamics	179
5.3.1	Physics of Taylor flows	180
5.3.2	Description of the benchmark problem	180
5.3.3	Methods compared in the benchmark problem	181
5.3.4	Case setup in DROPS	183
5.3.5	Simulation results	185
5.4	Two-phase flow problem with mass transport	186
5.4.1	Physical setting	187
5.4.2	Case setup in DROPS	190
5.4.3	Simulation results	192
6	Summary and Outlook	197
6.1	Summary	197
6.2	Open problems and outlook	200
	Bibliography	203
	List of Figures	217
	List of Tables	219

CHAPTER 1

Introduction

1.1 Motivation

In many industrial applications mass transfer from one fluid into another is an important process. In operations like extraction, gas scrubbing and waste water treatment the transfer of a certain species from one fluid into another is desired as efficient as possible. The design of technical installations and reactors requires detailed knowledge of fluid properties such as the shape of the interface between the fluids, the interfacial forces, main flow patterns, distribution of the phases and many more. Further, to optimize mass transfer units a profound knowledge of the mass transport processes, especially close to the interface, is imperative.

Direct numerical simulations are useful tools to evaluate and optimize the design of multiphase units. However, the development of reliable and accurate numerical methods is still challenging and is the topic of ongoing research. In the past decades various methods for the simulation of the fluid dynamics in such two-phase flow systems have been developed, e.g. the level set method [OS88, SSO94, Set99] or the Volume of Fluid (VoF) method [NW76, HN81].

In this thesis we focus on the discussion of numerical methods for a mass transport model in incompressible immiscible two-phase flows based on an interface description with interface capturing methods as the level set or Volume of Fluid method. This leads to an *implicit* description of the interface with a computational mesh which is not aligned to the fluid interface. This is in contrast to interface tracking methods, such as the Arbitrary Lagrangian-Eulerian description [Beh01, DHPRF04], where an explicit description of the interface is used.

In this thesis we describe new numerical methods for the simulation of mass transport problems within two-phase flows for stationary and non-stationary interfaces. The demanding aspect of the mass transport problem in two-phase flows results from the fact that the equations within the separate phases are coupled through interface conditions

which prescribe the conservation of mass and *Henry's law*. The latter leads to a jump discontinuity of the solution across the interface.

For *unfitted* interface problems with only weak discontinuities, i.e. problems where the solution is continuous but can have kinks across the interface, a finite element method based on the extended finite element (XFEM) method and the Nitsche technique, the Nitsche-XFEM method, has been proposed in the original paper [HH02]. In this paper a *stationary* problem without convection is considered. In [Ngu09, RN09] this method has been extended to unsteady problems including convection and solutions with a jump-discontinuity. However, in these publications the interface is assumed to be stationary and diffusion is assumed to be dominating.

The main achievements of this thesis are the extension of existing methods and the development of new methods for this class of *unfitted interface problems*:

- For the case of dominating convection and a *stationary* interface we derive a convection stabilized formulation of the Nitsche-XFEM method utilizing the Streamline Diffusion method [HB79, HB82, DH03]. We discuss the interplay between the Nitsche-XFEM method and the Streamline Diffusion method and prove quasi-optimal error bounds. The theoretical predictions are confirmed by numerical experiments which are discussed.
- In the literature for the Nitsche-XFEM method the problem of conditioning of the arising linear systems is rarely discussed. We investigate the performance of simple preconditioning techniques and develop a new, more sophisticated preconditioner for elliptic interface problems which is *optimal* in the following sense: The application of the preconditioner has only linear complexity and we can prove condition number bounds which are independent of the mesh size h and the position of the interface.
- A major contribution of the work is related to the *moving* interface case. We propose a space-time Galerkin formulation with trial and test functions which are discontinuous in time and combine this approach with an XFEM enrichment and a Nitsche technique. The resulting method is new. We present an error analysis which results in a proven second order error estimate in space and time which is confirmed by numerical examples.
- An implementation of the space-time method requires the numerical integration on four-dimensional geometries which are possibly cut by the implicitly described interface. The treatment of implicit domain descriptions for the numerical integration in space-time for the spatially three-dimensional case is not discussed in the literature. We propose a solution strategy based on an approximation of the (space-time) interface which allows for an explicit representation. The strategy contains new decomposition rules for four-dimensional geometries.

In this chapter we introduce the model for the mass transport in two-phase flows (section 1.2) and explain the key challenges for the numerical discretization arising from it (section 1.3). In section 1.4 we give an outline of the remainder of the thesis.

1.2 Mass transport model

In this section we derive a mathematical model for the transport of solute species within an incompressible immiscible two-phase flow. The remainder of this thesis discusses the numerical treatment of this model. We describe the physical balance laws for the species that are considered within the fluids and across the interface. We formulate a mathematical model and discuss a reformulation of the problem.

At this point, we do not discuss the fluid dynamics of incompressible immiscible two-phase flows but focus on the model for the species transport. In chapter 5, in the context of complex flows, the fluid dynamics and a suitable numerical discretization is discussed.

1.2.1 Balance laws

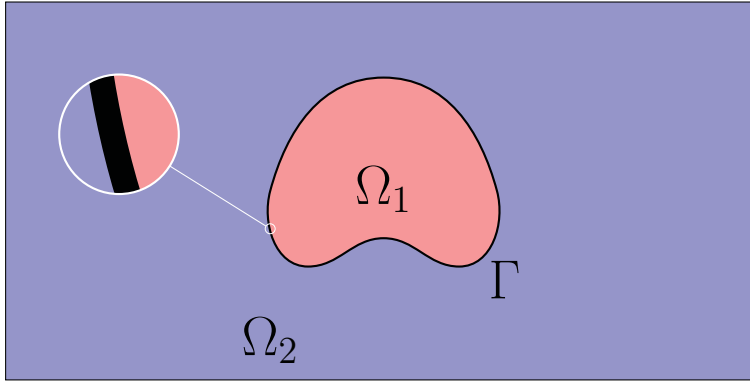


Figure 1.2.1: Sketch of two phases.

Consider the concentration u of a soluble species inside two immiscible incompressible fluids. The fluids are immiscible, contained in the domain Ω and separated by an interface. In reality, there is a transition layer from one phase into the other. In this layer a mixture of both species exists. However, the thickness of such a layer is in the order of several nanometers whereas the domain of interest is typically in the order of millimeters or larger. The resolution of the transition layer is most often circumvented by considering a *sharp interface* model where the transition layer thickness is assumed to be zero and the interface Γ is a lower dimensional manifold. The interface Γ divides Ω into two disjoint parts, Ω_1 and Ω_2 . One fluid is contained in Ω_1 whereas the other is contained in Ω_2 . There obviously holds $\Gamma = \overline{\Omega_1} \cap \overline{\Omega_2}$.

1 Introduction

In this thesis we make the following assumption.

Assumption 1.2.1 (Species conservation across the interface). *We assume that the species does not adhere to the interface and no chemical reactions take place at the interface.*

Inside each of the domains the concentration is transported via convection and molecular diffusion. At the (possibly) moving interface we pose two conditions that we discuss next.

The amount of a species u has to be conserved through the interface. As the two fluids are immiscible and we do not consider phase transition, the fluid velocities normal to the interface of the bulk phases coincide at the interface and determine the interface motion. The relative velocity of the fluid w.r.t. the interface velocity is zero in the normal direction $\mathbf{n}$ of the interface. Hence, the transport of species through the interface is only driven by diffusion. We apply Fick's law to model the flux due to molecular diffusion and together with the conservation of mass arrive at the first interface condition

$$\alpha_1 \nabla u_1 \cdot \mathbf{n} = \alpha_2 \nabla u_2 \cdot \mathbf{n}$$

with the diffusivity constants α_1 and α_2 of the two fluids. Here $\mathbf{n} = \mathbf{n}_1$ is the outer normal on Γ pointing from Ω_1 to Ω_2 . In general we have $\alpha_1 \neq \alpha_2$. The second interface condition is the *Henry interface condition*, that results from a constitutive law known as *Henry's law*. Henry's law assumes that chemical potentials from both sites are in balance, i.e. an instantaneous thermodynamical equilibrium is assumed. This assumption is reasonable as long as kinetic processes at the interface are sufficiently fast. Then, Henry's law states that the concentrations at the interface are proportional to the partial pressure of the species in the fluids $p = \beta_i u_i$ with constants β_i which depend on the solute, the solvent and the temperature. Using these constants Henry's law reads as

$$\beta_1 u_1 = \beta_2 u_2.$$

For further details on the modeling we refer to [Ish75, SAC97, SSO07]. The Henry interface condition leads to a discontinuity of the quantity u across the (evolving) interface as we typically have $\beta_1 \neq \beta_2$. Inside the fluid phases we use the same model for the diffusion as before (Fick's law) and end up with a linear convection-diffusion model:

$$\partial_t u - \operatorname{div}(\alpha \nabla u) + \mathbf{w} \cdot \nabla u = f$$

with f a source term which is typically zero in most applications and $\mathbf{w}$ the fluid velocity. At the boundary of the domain we prescribe the concentration (Dirichlet boundary conditions) or linear conditions on the flux (Neumann-type boundary conditions).

1.2.2 Mathematical model

All together we arrive at the following model posed on a domain Ω in the time interval $(0, T]$. In the remainder of this thesis we assume that Ω is a simple domain, for instance

a convex polygon. Note that due to the motion of the interface the domains Ω_i , $i = 1, 2$ and the interface Γ depend on time. We sum up the previous balances and conditions:

- bulk equations:

$$\partial_t u + \mathbf{w} \cdot \nabla u - \operatorname{div}(\alpha_i \nabla u) = f \quad \text{in } \Omega_i(t), \quad i = 1, 2, \quad t \in (0, T], \quad (1.2.1a)$$

- interface conditions:

$$[\![\alpha \nabla u \cdot \mathbf{n}]\!] = 0 \quad \text{on } \Gamma(t), \quad t \in (0, T], \quad (1.2.1b)$$

$$[\![\beta u]\!] = 0 \quad \text{on } \Gamma(t), \quad t \in (0, T], \quad (1.2.1c)$$

- initial conditions:

$$u(\cdot, 0) = u_0 \quad \text{in } \Omega_i(0), \quad i = 1, 2, \quad (1.2.1d)$$

- boundary conditions:

$$u(\cdot, t) = u_D \quad \text{on } \partial\Omega_D, \quad t \in (0, T], \quad (1.2.1e)$$

$$\alpha \nabla u(\cdot, t) \cdot \mathbf{n} = g_N \quad \text{on } \partial\Omega_N, \quad t \in (0, T], \quad (1.2.1f)$$

with the jump operator $[\![\cdot]\!]$ at the interface Γ defined as

$$[\![v(\mathbf{x})]\!] = \lim_{s \rightarrow 0^+} v(\mathbf{x} + s \cdot \mathbf{n}) - \lim_{s \rightarrow 0^+} v(\mathbf{x} - s \cdot \mathbf{n}), \quad \mathbf{x} \in \Gamma(t), \quad t \in (0, T]. \quad (1.2.2)$$

The concentration u is double-valued at the interface. To distinguish between those values we introduce the notation $u_i := u|_{\Omega_i}$ such that

$$[\![\beta u]\!] = \beta_1 u_1 - \beta_2 u_2 \quad \text{and} \quad [\![\alpha \nabla u \cdot \mathbf{n}]\!] = \sum_{i=1}^2 \alpha_i \nabla u_i \cdot \mathbf{n}_i.$$

Here $\mathbf{n}_i$ denote the outer normal of Ω_i .

Although the discretization methods discussed in this work allow for the boundary conditions (1.2.1e), (1.2.1f) and linear combinations (Robin-type boundary conditions), we will mostly restrict ourselves to Dirichlet boundary conditions and set $\partial\Omega = \partial\Omega_D$.

Assumption 1.2.2 ($\beta_i \geq 1$). *Scaling the solubilities β_i with the same constant c in both domains does not change the solution. In the following we set $\beta_{\max} = \max\{\beta_1, \beta_2\}$ and $\beta_{\min} = \min\{\beta_1, \beta_2\}$ and assume $\beta_{\min} = 1$.*

Assumption 1.2.3 (moderate ratios of β). *If not addressed otherwise we further assume that the solubilities in the domains are in the same order of magnitude, i.e. we assume $\beta_{\max}/\beta_{\min} = \mathcal{O}(1)$.*

We assumed that mass transport through the interface is only caused by diffusive transfer. Therefore we make the following assumption on the velocity field $\mathbf{w}$:

Assumption 1.2.4 (compatible velocity). *The velocity field is assumed to originate from an incompressible flow. Further the interface motion in normal direction, denoted by $\mathcal{V} \cdot \mathbf{n}$, has to coincide with the convective velocity $\mathbf{w} \in H(\text{div}, \Omega)$:*

$$-\text{div } \mathbf{w} = 0 \text{ in } \Omega_i, i = 1, 2, \quad \mathbf{w} \cdot \mathbf{n} = \mathcal{V} \cdot \mathbf{n} \text{ at } \Gamma \quad (1.2.3)$$

Further we assume that the velocity is bounded in the L^∞ -norm:

$$|\mathbf{w}|_\infty := \|\mathbf{w}\|_{L^\infty(\Omega)} \leq c < \infty \quad (1.2.4)$$

1.2.3 A reformulation

To get rid of the discontinuity at the interface one can consider an equivalent problem by substituting $\tilde{u} = \beta u$. This formulation is sometimes used in the literature.

Problem 1.2.1.

$$\beta^{-1} \partial_t \tilde{u} + \tilde{\mathbf{w}} \cdot \nabla \tilde{u} - \text{div}(\tilde{\alpha} \nabla \tilde{u}) = f \quad \text{in } \Omega_i, i = 1, 2, t \in (0, T], \quad (1.2.5a)$$

$$[[\tilde{\alpha} \nabla \tilde{u} \cdot \mathbf{n}]]_\Gamma = 0, \quad (1.2.5b)$$

$$[[\tilde{u}]]_\Gamma = 0, \quad (1.2.5c)$$

$$\tilde{u}(\cdot, 0) = \tilde{u}_0 \quad \text{in } \Omega_i(0), i = 1, 2, \quad (1.2.5d)$$

$$\tilde{u}(\cdot, t) = \tilde{u}_D \quad \text{on } \partial\Omega, t \in (0, T]. \quad (1.2.5e)$$

with $\tilde{\mathbf{w}} = \beta^{-1} \mathbf{w}$, $\tilde{\alpha} = \beta^{-1} \alpha$, $\tilde{u}_0 = \beta^{-1} u_0$ and $\tilde{u}_D = \beta^{-1} u_D$. The quantity $\tilde{u}$ is obviously continuous in Ω but can still have a kink (discontinuity in the derivative) across the interface. The reformulation comes at the price of having a β^{-1} -scaled time derivative and a discontinuous velocity $\tilde{\mathbf{w}}$. For the stationary case where $\tilde{\mathbf{w}} \cdot \mathbf{n} = 0$ on Γ and $\partial_t u = 0$ in Ω this reformulation can help simplifying the problem. The reformulation reduces the problem of having a discontinuity and a kink to a problem with a continuous solution and a kink across the interface.

1.2.4 Eulerian description

In fluid dynamics a specification of the balance laws which is based on specific fixed locations in space is called *Eulerian*. The counterpart of an Eulerian description is the *Lagrangian* specification where the balances are formulated relative to a fluid parcel which moves through space and time following the flow field.

Both (and mixed) formulations are used to derive different discretization methods for flow problems. Discretizations based on a Lagrangian description offer a natural treatment for problems with moving boundaries or interfaces. However, Lagrangian methods have

significant drawbacks if deformations get large or topologies change. These issues can be overcome by Eulerian methods. However, the discretization of problems with moving boundaries or interfaces in an Eulerian frame is difficult. A major component of the numerical difficulties discussed in this work arises from the fact that we consider the problem in an Eulerian framework with an implicit description of the interface. In contrast to methods which are based on a Lagrangian formulation at the interface, e.g. a full Lagrangian method or an Arbitrary-Lagrangian-Eulerian formulation (cf. section 3.1.2), the computational mesh is not adapted to fit the interface. As a consequence the interface and thus the discontinuity of the concentration lies or even moves inside a computational element rather than coinciding with element facets.

1.3 Numerical challenges

We briefly summarize the key issues for the numerical solution of mass transport problems in two-phase flows with an implicit description of the interface.

Discontinuous solutions across an unfitted interface. Due to the Henry interface condition the solution of the considered mass transport problem is discontinuous across the interface. Further, the interface is described only implicitly. Hence, the solution has a discontinuity the position of which lies within discretization elements. Standard piecewise polynomial ansatz functions have only a very poor approximation quality in such a situation. For the approximation of functions which are discontinuous across the interface we use an extended finite element (XFEM) space.

The interface is moving in time through the mesh and thus the discontinuity is also moving through the mesh. The application of standard time integration techniques such as the method of lines rely on solutions which are continuous in time and hence the method of lines is not applicable. We introduce a space-time formulation to solve this problem.

Integration on implicitly defined geometries. Finite element discretizations defined on unfitted meshes utilizing an implicit description of the interface at some point define integrals on the separated sub-domains and the interface. The numerical approximation of these integrals needs special solution strategies. Especially the case of intersected four-dimensional prisms stemming from a space-time formulation requires new strategies.

Convection is dominating in many applications. In many applications diffusion is small compared to convection. This can lead to very thin boundary layers close to the interface which can be difficult to resolve numerically. Further, standard finite element discretizations are known to have stability problems if convection dominates. To handle also convection dominated problems stabilization techniques are necessary.

1.4 Outline of the thesis

The thesis is organized as follows.

- In **chapter 2** we discuss the special case of a stationary interface. In that case the interface and the separated sub-domains are independent of time. We introduce a spatial discretization combining two techniques, which we introduce successively: the extended finite element method (XFEM) for the approximation of discontinuous quantities and the Nitsche method for the (weak) imposition of interface conditions. A convection stabilization of the resulting method for the convection dominated case is added using the concept of Streamline Diffusion methods and a corresponding error analysis is carried out. We further discuss preconditioners for this special method and propose a preconditioner the *optimality* of which we prove for elliptic unfitted interface problems. The chapter concludes with the discussion of numerical examples for the presented discretization methods and preconditioners.
- In **chapter 3** we consider the more challenging case of a moving interface. To account for the moving interface in the discretization we combine the discretization techniques applied to the stationary problem with a space-time finite element formulation. The method is derived and an error analysis is carried out which guarantees second order convergence in space and time. We further discuss the problem of preconditioning and evaluate the method on interesting numerical examples.
- In **chapter 4** we discuss the topic of numerical integration. The interface in the setting of this work is typically not given explicitly, but implicitly, for instance as the zero level of a level set function. The finite element formulations for the considered methods, however, require a robust and accurate evaluation of integrals on the interface and the particular sub-domains. An approximation of the interface is constructed which allows for an explicit representation. This explicit representation can then be used to obtain polygonal subdomains and interfaces on which numerical integration is applied. The approximation and the numerical integration is especially challenging for the space-time method introduced in chapter 3 if the spatial domain is three-dimensional. In that case, the arising geometries are four-dimensional and the numerical treatment of the arising polygonal domains is non-standard. We propose a solution strategy for this problem.
- In **chapter 5** we consider realistic two-phase flow problems. Numerical methods for the solution of the fluid dynamics of incompressible immiscible two-phase flows are briefly introduced and simulation results for a two-phase flow problem without mass transport as well as a coupled fluid dynamics problem with mass transport are presented and discussed.
- In **chapter 6** we summarize the main results of this thesis and discuss open questions and future perspectives.

CHAPTER 2

Mass transport through a stationary interface

A special case of the problem in (1.2.1) is the case of a stationary interface where the domains and the interface do not depend on time. In this chapter we discuss the discretization of the mass transport problem in an *unfitted* setting for a stationary interface, that means that the triangulation is not aligned to the *stationary interface*.

Outline of this chapter

In section 2.1 the mathematical model is presented and a well-posed weak formulation of this model is given. Section 2.2 discusses the arising numerical challenges for the discretization and presents an approach to solve the problem numerically. A corresponding a priori error analysis is presented in section 2.3. One interesting aspect of the Nitsche-XFEM method presented in section 2.2 is the fact, that the arising linear systems can become very ill-conditioned. In section 2.4 we will discuss the conditioning of the linear systems and present solution strategies. We conclude the chapter with numerical examples in section 2.5.

2.1 Problem description

We consider the problem in (1.2.1) for a stationary interface $\Gamma(t) = \Gamma$.

Problem 2.1.1.

$$\partial_t u + \mathbf{w} \cdot \nabla u - \operatorname{div}(\alpha \nabla u) = f \quad \text{in } \Omega_i, \quad i = 1, 2, \quad t \in [0, T], \quad (2.1.1a)$$

$$\llbracket \alpha \nabla u \cdot \mathbf{n} \rrbracket_\Gamma = 0 \quad \text{on } \Gamma, \quad t \in [0, T], \quad (2.1.1b)$$

$$\llbracket \beta u \rrbracket_\Gamma = 0 \quad \text{on } \Gamma, \quad t \in [0, T], \quad (2.1.1c)$$

$$u(\cdot, 0) = u_0 \quad \text{in } \Omega_i, \quad i = 1, 2, \quad (2.1.1d)$$

$$u(\cdot, t) = g_D \quad \text{on } \partial\Omega, \quad t \in [0, T]. \quad (2.1.1e)$$

Note that due to assumption 1.2.4 (compatible velocity) we require $\mathbf{w} \cdot \mathbf{n} = 0$ on Γ and $\operatorname{div} \mathbf{w} = 0$ in Ω_i , $i = 1, 2$. In the next subsections we introduce simplified problems which are later used to facilitate the presentation of the discretizations and their key properties in section 2.2. Further we introduce a well-posed weak formulation of the problem 2.1.1 and the simplified versions.

2.1.1 Simplified problems

We introduce two simplified problems which are stationary versions of problem 2.1.1 (with $\partial_t u = 0$). The simplest problem further neglects convection.

2.1.1.1 Two-domain stationary convection-diffusion equation

A stationary solution to problem 2.1.1 solves

Problem 2.1.2.

$$\mathbf{w} \cdot \nabla u - \operatorname{div}(\alpha \nabla u) = f \quad \text{in } \Omega_i, \quad i = 1, 2, \quad (2.1.2a)$$

$$[\![\alpha \nabla u \cdot \mathbf{n}]\!]_{\Gamma} = 0 \quad \text{on } \Gamma, \quad (2.1.2b)$$

$$[\![\beta u]\!]_{\Gamma} = 0 \quad \text{on } \Gamma, \quad (2.1.2c)$$

$$u = g_D \quad \text{on } \partial\Omega. \quad (2.1.2d)$$

This problem, at least with an unfitted interface, is rarely discussed in the literature, especially when convection dominates.

2.1.1.2 Two-domain Poisson equation

The simplest version of problem 2.1.1 is obtained by considering a stationary problem without convection:

Problem 2.1.3.

$$-\operatorname{div}(\alpha \nabla u) = f \quad \text{in } \Omega_i, \quad i = 1, 2, \quad (2.1.3a)$$

$$[\![\alpha \nabla u \cdot \mathbf{n}]\!]_{\Gamma} = 0 \quad \text{on } \Gamma, \quad (2.1.3b)$$

$$[\![\beta u]\!]_{\Gamma} = 0 \quad \text{on } \Gamma, \quad (2.1.3c)$$

$$u = g_D \quad \text{on } \partial\Omega. \quad (2.1.3d)$$

For $\beta_1 = \beta_2$ (or after reformulation as in section 1.2.3) the problem is a standard interface problem in the literature.

Remark 2.1.1 (Interface problems). *In the literature problems with material parameters which are discontinuous across a given interface leading to discontinuities in the derivative (kinks) or the function value itself (jumps) are called interface problems. For the stationary cases $\partial_t u = 0$, i.e. problems (2.1.3) and (2.1.2), we can apply the reformulation from section 1.2.3 to get rid of the discontinuity. Such a reformulation allows to consider many ideas and concepts from the literature which typically consider problems with continuous solutions with discontinuous normal derivatives.*

2.1.2 Weak formulation

In this section we discuss a well-posed weak formulation for problem 2.1.1 under reasonable assumptions on the data. The discussion is kept brief. For a more thorough discussion we refer to [RN09],[GR11, Chapter 10.2] and the references therein.

For simplicity we only consider homogeneous Dirichlet boundary conditions ($g_D = 0$ in problem 2.1.1). Since we restrict to the case of a stationary interface, the discontinuity in the solution is located at a fixed position, independent of time t , which allows for a rather standard weak formulation. In case of an evolving interface a space-time weak formulation is more natural, cf. chapter 3.

We need the broken spaces

$$H^k(\Omega_1 \cup \Omega_2) := \{ v \in L^2(\Omega), v|_{\Omega_i} \in H^k(\Omega_i), i = 1, 2 \}, k \in \mathbb{N} \quad (2.1.4)$$

$$H_0^1(\Omega_1 \cup \Omega_2) := \{ v \in H^1(\Omega_1 \cup \Omega_2), v|_{\partial\Omega} = 0 \}. \quad (2.1.5)$$

To abbreviate notation we also write

$$H^k(\Omega_{1,2}) = H^k(\Omega_1 \cup \Omega_2), \quad H_0^1(\Omega_{1,2}) = H_0^1(\Omega_1 \cup \Omega_2).$$

For $v \in H_0^1(\Omega_{1,2})$ we write $v_i := v|_{\Omega_i}$, $i = 1, 2$. Furthermore we define

$$L_\beta^2(\Omega) := L^2(\Omega), \quad H_{0,\beta}^1(\Omega) := \{ v \in H_0^1(\Omega_{1,2}), \llbracket \beta v \rrbracket = 0 \text{ on } \Gamma \}. \quad (2.1.6)$$

Note that $v \in H_{0,\beta}^1(\Omega)$ iff $\beta v \in H_0^1(\Omega)$. On $L_\beta^2(\Omega)$ we use the scalar product

$$(u, v)_0 := (\beta u, v)_{L^2} = \int_{\Omega} \beta u v \, d\mathbf{x}, \quad (2.1.7)$$

which is equivalent to the standard scalar product on $L^2(\Omega)$. The corresponding norm is denoted by $\| \cdot \|_0$. For $u, v \in H^1(\Omega_i)$ we define $(u, v)_{1,\Omega_i} := \beta_i \int_{\Omega_i} \nabla u_i \cdot \nabla v_i \, d\mathbf{x}$ and furthermore

$$(u, v)_{1,\Omega_{1,2}} := (u, v)_{1,\Omega_1} + (u, v)_{1,\Omega_2}, \quad u, v \in H^1(\Omega_{1,2}).$$

The corresponding semi-norm is denoted by $| \cdot |_{1,\Omega_{1,2}}$ and the norm is

$$\| \cdot \|_{1,\Omega_{1,2}} := \left(\| \cdot \|_0^2 + | \cdot |_{1,\Omega_{1,2}}^2 \right)^{\frac{1}{2}}.$$

We emphasize that the norms $\|\cdot\|_0$ and $\|\cdot\|_{1,\Omega_{1,2}}$ depend on β . We define the bilinear forms

$$a(u, v) := (\alpha u, v)_{1,\Omega_{1,2}}, \quad u, v \in H^1(\Omega_{1,2}), \quad (2.1.8)$$

$$c(u, v) := (\mathbf{w} \cdot \nabla u, v)_0, \quad u, v \in H^1(\Omega_{1,2}). \quad (2.1.9)$$

Note that these are well-defined also for functions which do not fulfill the interface conditions.

2.1.2.1 Weak formulation of stationary problem

We define the following weak formulation of problem 2.1.3. Let $H_\beta^{-1}(\Omega)$ be the dual space to $H_{0,\beta}^1(\Omega)$ and assume $f \in H_\beta^{-1}(\Omega)$. Find $u \in H_{0,\beta}^1(\Omega)$, such that

$$a(u, v) + c(u, v) = \langle f, v \rangle \quad \forall v \in H_{0,\beta}^1(\Omega) \quad (2.1.10)$$

where $\langle \cdot, \cdot \rangle$ denotes the duality pairing between $H_\beta^{-1}(\Omega)$ and $H_{0,\beta}^1(\Omega)$. For smooth data f we assume the following regularity for the unique solution of (2.1.10)

$$\|u\|_{2,\Omega_{1,2}} \leq c\|f\|_0 \quad (2.1.11)$$

for a constant c independent of f .

2.1.2.2 Weak formulation of non-stationary problem

The time derivative $\partial_t u$ is defined in a distributional sense using Bochner spaces, $\partial_t u \in L^2(0, T; H_\beta^{-1}(\Omega))$ while we have $u \in L^2(0, T; H_{0,\beta}^1(\Omega))$. We introduce the following space

$$W^1(0, T; H_{0,\beta}^1(\Omega)) := \{v \in L^2(0, T; H_{0,\beta}^1(\Omega)), \partial_t v \in L^2(0, T; H_\beta^{-1}(\Omega))\}. \quad (2.1.12)$$

There holds $C([0, T]; L_\beta^2(\Omega)) \subset W^1(0, T; H_{0,\beta}^1(\Omega))$ so that initial values $u(\cdot, 0) = u_0$ are well-defined. Consider the following weak formulation of the mass transport problem, problem 2.1.1, for $f \in H_\beta^{-1}(\Omega)$, $u_0 \in H_{0,\beta}^1(\Omega)$:

Determine $u \in W^1(0, T; H_{0,\beta}^1(\Omega))$ such that $u(\cdot, 0) = u_0$ and for almost all $t \in (0, T)$:

$$\langle \partial_t u, v \rangle + a(u, v) + c(u, v) = \langle f, v \rangle \quad \text{for all } v \in H_{0,\beta}^1(\Omega). \quad (2.1.13)$$

The weak formulation also has a unique solution, see [GR11, lemma 10.2.3]. For sufficiently smooth data f and u_0 the unique solution of the weak formulation (2.1.13) has a higher regularity, see [GR11, Theorem 10.2.2].

2.2 Discretization of the stationary problem using Nitsche-XFEM

In this section we present the Nitsche-XFEM method for the discretization of unfitted interface problems. We give a short outline. In section 2.2.1 we discuss the problem of how to approximate *unfitted* discontinuities. We introduce and discuss the ideas of fictitious domain and extended finite element methods. Further, we introduce the extended finite element space V_h^Γ which is used in the remainder of this chapter. Since the presented finite element spaces do not implement the interface conditions as *essential* conditions, in section 2.2.2 we present a way to implement the interface condition via a variational formulation. This is done with a Nitsche technique. The resulting Nitsche-XFEM method is our favored choice in this work. There are, however, other approaches to deal with interface conditions with non-conforming spaces. Those are closely related to a Nitsche discretization. Therefore we discuss some modifications of the Nitsche method and different approaches in section 2.2.3. As the Nitsche-XFEM method is based on a standard Galerkin method for the separate domains it also inherits its problems in the convection dominated case. For standard finite elements in one phase, one typically applies some method of stabilization. In section 2.2.5 we apply the ideas from Streamline Diffusion stabilization and discuss the interaction of Nitsche and Streamline Diffusion method.

Preliminaries

Let $\{\mathcal{T}_h\}_{h>0}$ be a family of shape regular simplex triangulation of Ω . A triangulation $\mathcal{T}_h$ consists of simplices T , with $h_T := \text{diam}(T)$ and $h := \max\{h_T \mid T \in \mathcal{T}_h\}$. In general we have that the interface Γ does not coincide with element boundaries. The triangulation is *unfitted*. We introduce some notation for *cut elements*, i.e. elements T with $\Gamma \cap T \neq \emptyset$. For any simplex $T \in \mathcal{T}_h$, $T_i := T \cap \Omega_i$ denotes the part of T in Ω_i and $\Gamma_T := T \cap \Gamma$ the part of the interface that lies in T . $\mathcal{T}_h^\Gamma$ denotes the set of elements that are “close to the interface”, $\mathcal{T}_h^\Gamma := \{T : T \cap \Gamma \neq \emptyset\}$. The corresponding domain is denoted by $\Omega^\Gamma = \{\mathbf{x} \in T : T \in \mathcal{T}_h^\Gamma\}$. Further, we define the set of elements with nonzero support in one domain: $\mathcal{T}_h^i := \{T : T \cap \Omega_i \neq \emptyset\}$, $i = 1, 2$, the corresponding domain is denoted by $\Omega_i^+ = \{\mathbf{x} \in T : T \in \mathcal{T}_h^i\}$. We also define the domain of uncut elements in domain i as $\Omega_i^- = \Omega_i \setminus \Omega^\Gamma = \Omega_i^+ \setminus \Omega^\Gamma$.

At some places we use the notation with the relations $\preceq$ and $\succeq$.

Definition 2.2.1 (Notation: smaller/greater up to a constant ($\preceq, \succeq$), equivalent ($\simeq$)). For $a, b \in \mathbb{R}$ we use the notation $a \preceq b$ ($a \succeq b$), if there exists a constant $c \in \mathbb{R}$ such that there holds $a \leq cb$ ($a \geq cb$), with c independent of h or the cut position. If we have $a \preceq b$ and $b \preceq a$, we write $a \simeq b$.

Assumption 2.2.1 (Resolution of the interface). We assume that the resolution close to the interface is sufficiently high such that the interface can be resolved by the triangulation,

in the sense that if $\Gamma \cap T =: \Gamma_T \neq \emptyset$ then Γ_T can be represented as the graph of a function on a planar cross-section of T . We refer to [HH02] for precise conditions.

Remark 2.2.1 (Interface approximation). *In implementations of any method with an unfitted triangulation one needs to deal with the interface Γ in terms of subdomain and interface integrals. In practice Γ is often defined implicitly, e.g. as the zero level of a given level set function. As soon as the level set function is not (piecewise) linear the interface Γ is not (piecewise) planar and an explicit construction is (usually) not feasible. Often an approximation Γ_h of Γ is constructed which has an explicit representation and easily allows for implementations of subdomain and interface integrals. In this chapter however we neglect this issue and assume that we can evaluate integrals on subdomains and the interface exactly. In chapter 4 a strategy to construct suitable approximations Γ_h is discussed. This strategy is also used in the numerical examples.*

2.2.1 Approximation of discontinuous quantities (XFEM)

In this section we consider the approximation quality of certain finite element spaces w.r.t. domain-wise smooth functions u with a discontinuity across the interface, i.e. the approximation error of a finite element space V_h

$$\inf_{v_h \in V_h} \|v_h - u\|_{H^k(\Omega_{1,2})}, \quad k = 0, 1.$$

We consider the finite element space V_h of continuous functions which are polynomials of degree k on each element:

$$V_h := \{v \in H^1(\Omega) : v|_T \in \mathcal{P}^k(T), T \in \mathcal{T}_h\}.$$

It is well-known that the approximation of discontinuous functions u (with an *unfitted* discontinuity) with piecewise polynomials only allows for an approximation estimates of the form:

$$\inf_{v_h \in V_h} \|v_h - u\|_{L^2(\Omega)} \leq c\sqrt{h} \|u\|_{H^k(\Omega_{1,2})}, \quad k \geq 1$$

This estimate is sharp, cf. [GR11, Section 7.9.1]) and the numerical example in section 2.5.1.2. This result is independent of the choice of continuity restrictions at element boundaries. Hence, applying standard finite element discretizations (including Discontinuous Galerkin (DG) discretizations) without further adaptations for problems with discontinuous solutions will not lead to satisfying results.

Consider the simpler problems, problem 2.1.3 and problem 2.1.2 which allow for the reformulation in section 1.2.3 to get rid of the discontinuity across the interface. After reformulation the jump discontinuity vanishes but the discontinuity in the derivative (kink discontinuity) due to different (transformed) diffusivities $\tilde{\alpha}$ remains. In this case the approximation quality of standard finite element spaces is better, cf. the numerical results in section 2.5.1.2. Still, the sub-optimal approximation error estimate

$$\inf_{v_h \in V_h} \|v_h - u\|_{L^2(\Omega)} \leq ch^{\frac{3}{2}} \|u\|_{H^k(\Omega_{1,2})}, \quad k \geq 1$$

is sharp, independent of the polynomial degree of the finite element space V_h . In the next sections a remedy to this problem is presented.

2.2.1.1 The fictitious domain approach

To overcome the approximation problem for kinks and jumps that are not fitted to the mesh we introduce special finite element spaces. The main idea is sketched in figure 2.2.1 and is as follows: Consider the problem of approximating a function u_1 in Ω_1 when $\partial\Omega_1$ is not fitted to the discretization elements. If that function u_1 is sufficiently smooth it can be extended smoothly to Ω and a standard finite element space V_h with (element-) piecewise polynomials of degree k can be used to approximate the function with the usual (good) quality of approximation. We denote the corresponding (continuous) extension operator as $\mathcal{E}_1 : H^k(\Omega_1) \rightarrow H^k(\Omega)$. For the L^2 -norm one directly gets

$$\inf_{v_h \in V_h} \|v_h - u\|_{L^2(\Omega_1)} \leq \inf_{v_h \in V_h} \|v_h - \mathcal{E}_1 u\|_{L^2(\Omega)} \leq ch^{k+1} \|\mathcal{E}_1 u\|_{H^{k+1}(\Omega)} \leq ch^{k+1} \|u\|_{H^{k+1}(\Omega_1)}.$$

It is already sufficient to extend the functions to the smallest set of elements that have some part in domain i , Ω_i^+ . This is the basic idea of the fictitious domain approach and it appears in the literature under different names and in different contexts. We briefly discuss the literature on fictitious domain approaches in section 2.2.2.

The same idea that we just applied for Ω_1 can also be applied for the function in Ω_2 . In order to approximate both functions at the same time we have to use twice the degrees of freedom of V_h in the overlap Ω^Γ . We get the finite element space

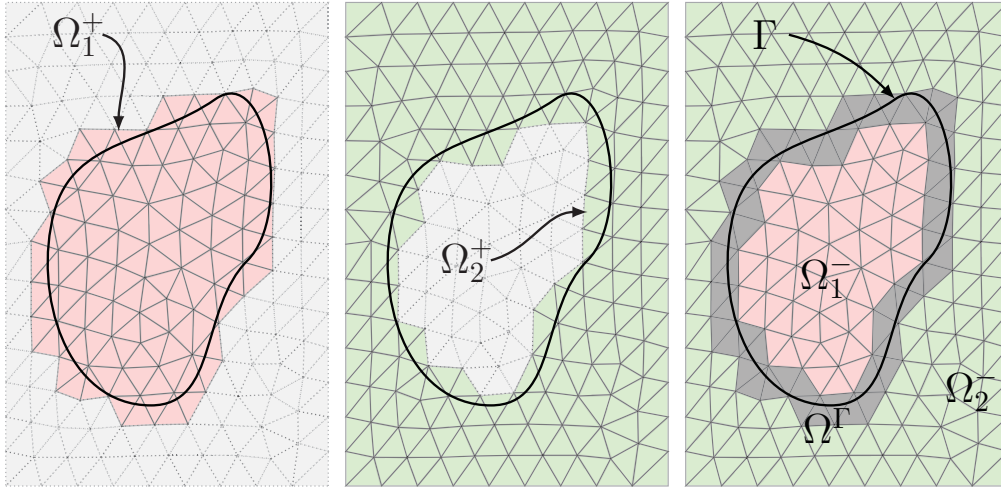


Figure 2.2.1: Fictitious domain approach applied for domain Ω_1 (left) and Ω_2 (center). Combining both results in a finite element space with double-valued representatives in the overlap region Ω^Γ (right).

$$V_h^\Gamma := \{v \in H_0^1(\Omega_{1,2}) \mid v|_{T_i} \in \mathcal{P}^k(T_i) \text{ for all } T \in \mathcal{T}_h, i = 1, 2\}. \quad (2.2.1)$$

which can be characterized as

$$V_h^\Gamma = \mathcal{R}_1 V_h \oplus \mathcal{R}_2 V_h \quad (2.2.2)$$

with $\mathcal{R}_i : L^2(\Omega) \rightarrow L^2(\Omega_i)$ the restriction operator on domain i .

Note that $V_h^\Gamma \subset H_0^1(\Omega_{1,2})$, but $V_h^\Gamma \not\subset H_{0,\beta}^1(\Omega)$, since the Henry interface condition $[[\beta v_h]] = 0$ does not necessarily hold for $v_h \in V_h^\Gamma$. The task of enforcing the interface condition is shifted to the (discrete) variational formulation. This is later discussed in section 2.2.2.

A different characterization can be made which is typically better suited for implementation and discussed in the next section.

2.2.1.2 The extended finite element method (XFEM)

In the literature a finite element discretization based on the space V_h^Γ is often called an *extended finite element method* (XFEM), cf. [MDB99, BMUP01, CB03]. Furthermore, in the (engineering) literature this space is usually characterized in a different way, which we briefly explain for linear finite elements ($k = 1$). Let $V_h \subset H_0^1(\Omega)$ be the standard finite element space of continuous piecewise linear functions, corresponding to the triangulation $\mathcal{T}_h$. Define the index set $\mathcal{J} = \{1, \dots, n\}$, where $n = \dim V_h$, and let $(\varphi_i)_{i \in \mathcal{J}}$ be the nodal basis in V_h . Let $\mathcal{J}_\Gamma := \{j \in \mathcal{J} \mid |\Gamma \cap \text{supp}(\varphi_j)| > 0\}$ be the index set of those basis functions the support of which is intersected by Γ . The Heaviside function H_Γ has the values $H_\Gamma(\mathbf{x}) = 0$ for $\mathbf{x} \in \Omega_1$, $H_\Gamma(\mathbf{x}) = 1$ for $\mathbf{x} \in \Omega_2$. Using this, for $j \in \mathcal{J}_\Gamma$ we introduce a so-called *enrichment function* $\Phi_j(\mathbf{x}) := |H_\Gamma(\mathbf{x}) - H_\Gamma(\mathbf{x}_j)|$, where $\mathbf{x}_j$ is the vertex with index j . We introduce new basis functions $\varphi_j^\Gamma := \varphi_j \Phi_j$, $j \in \mathcal{J}_\Gamma$, and define the space

$$V_h \oplus V_h^x \text{ with } V_h^x := \text{span}\{\varphi_j^\Gamma \mid j \in \mathcal{J}_\Gamma\}. \quad (2.2.3)$$

In figure 2.2.2 a sketch of an added basis function is depicted. The space $V_h \oplus V_h^x$ is the same as V_h^Γ in (2.2.1) and the characterization in (2.2.3) accounts for the name “extended finite element method”. The new basis functions φ_j^Γ have the property $\varphi_j^\Gamma(\mathbf{x}_i) = 0$ for all $i \in \mathcal{J}$. From an implementational point of view this is an important property as it guarantees that $v(\mathbf{x}) = 0$ for $\mathbf{x} \in \Omega \setminus \Omega^\Gamma$ and $v \in V_h^x$, i.e. that only on discretization elements which are cut, (non-zero) enrichment functions exist. An L^2 -stability property of the basis $(\varphi_j)_{j \in \mathcal{J}} \cup (\varphi_j^\Gamma)_{j \in \mathcal{J}_\Gamma}$ of V_h^Γ (for $k = 1$) is given in [Reu08].

2.2.1.3 The fictitious domain approach and the extended finite element method in the literature

The general idea of fictitious domain approaches is to find a solution to a PDE problem on a complicated domain Ω by replacing the problem with a problem on a larger domain $\tilde{\Omega} \supset \Omega$ such that the restriction to Ω of the solution coincides with the solution of

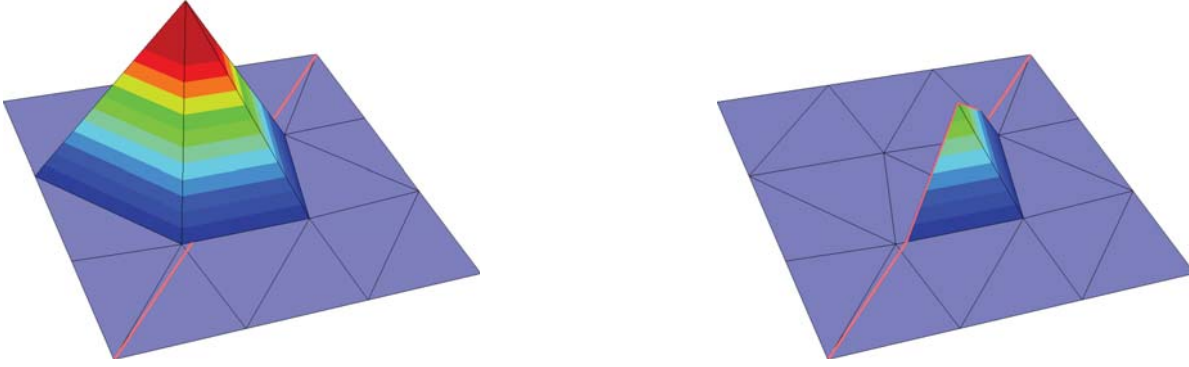


Figure 2.2.2: Example of an XFEM shape functions. On the left a shape function φ_j from the standard finite element space V_h is shown. On the right the restriction $\mathcal{R}_1\varphi_j$ on Ω_1 is shown. The function $\mathcal{R}_1\varphi_j$ is a basis function of V_h^x .

the original problem. Typically, the domain $\tilde{\Omega}$ is chosen as a simple geometry which is easily meshed. The main motivation for this approach is that one can work with a simple background mesh that is independent of a (possibly) complex and time-dependent geometry. This apparent simplification comes at a price. The interface is not aligned to element boundaries of a triangulation, the interface is *unfitted*. Managing data structures pertaining the actual geometry is in general not trivial. Further the imposition of boundary (or interface) conditions that are posed on the boundary (or interface) of the physical domain needs special treatments. The latter aspect will be discussed in detail in section 2.2.2 and as there are several ways to attack this problem many similar methods exist. They are all based on the main idea of fictitious domains, which is the extension of the problem in Ω to $\tilde{\Omega}$.

The first *unfitted* finite element methods were based on penalty formulations for Dirichlet boundary conditions and have been investigated and analyzed in [Bab73a, BE86]. The fictitious domain method which makes use of Lagrangian multipliers to implement Dirichlet boundary conditions is discussed and analyzed in a series of papers by Glowinski et al. [GPP94a, GPP94b, GG95]. We also list other methods which are based on very similar ideas.

In fluid-structure interaction problems, immersed boundary (IB) methods (see, e.g., [PM89]) use non-matching overlapping grids, for example a static mesh for the fluid and a moving mesh for the object which is in contact with the fluid (and its vicinity). Typically, on one of the meshes the equations are formulated in an Eulerian framework, while on the other mesh, which is moving, one uses a Lagrangian (or semi-Lagrangian) formulation. Force balance is then controlled at a number of points in the intersection of both domains. A variant of the IB method is the Immersed Interface (II) method (cf. [LL94]). For problems with perforated domains or domains with single holes, the Fat Boundary method (FBM) introduced in [Mau01] is another method which adapts the fictitious domain idea similar to the IB and II method.

Similar to the FBM the finite cell method (FCM) introduced in [PDR07] is a method to

compute structure problems in solids with randomly shaped voids on regular grids using higher order elements. In [VvLS08] an overview over several fictitious domain approaches which are suitable for higher order discretizations is given. A higher order discretization of an unfitted interface problem (similar to the one discussed in this section) is presented in [Mas12].

In most of those methods the construction of the underlying finite element spaces follows standard ideas. On the background mesh standard basis functions are used and on overlapping domains the basis functions are defined according to the corresponding meshes. We have already seen that in the context of unfitted interface problems the (two-domain) fictitious domain approach coincides with the extended finite element method (XFEM). We briefly discuss the basic idea and original purpose of XFEM methods. The extended finite element method (XFEM) was introduced by Belytschko et al in [MDB99]. The XFEM method has its origin in structural mechanics when dealing with crack phenomena. The core component of the method is the combination of an implicit (mesh-free) geometry representation and an enrichment of a finite element space by singular and discontinuous functions. The choice of those enrichment functions is problem-dependent. For the representation of jumps a Heaviside-enrichment as presented above is suitable. To approximate kinks an enrichment with a distance function can be applied, cf. [MCCR03]. In this work we only consider the jump-enrichment.

We also mention the approach in [FR14], where on an unfitted background mesh an explicit triangulation of the interface is used *only locally* to define finite element functions which allow for kinks in the solution.

2.2.2 Imposing interface conditions in non-conforming finite element spaces (Nitsche)

In the previous section we discussed how to recover the (good) approximation quality of the standard situation (where no kink or jump discontinuity is present) for problems with discontinuous solutions across an *unfitted* interfaces. However, across the interface no conditions are implemented as *essential* conditions on the introduced finite element space V_h^Γ . Especially the Henry interface condition is not considered. Thus, the finite element space is *non-conforming* w.r.t. the interface condition, i.e. we have $V_h^\Gamma \not\subset H_{0,\beta}^1(\Omega)$.

In [MBT06] the (simpler) case of one fictitious domain and the problem of imposing Dirichlet values as *essential* conditions is considered and it is shown for a simple example that a strong imposition of boundary conditions can lead to problems (“boundary locking”). In that case the strong imposition leads to non-physical conditions on the boundary fluxes which results in an over-constrained solution.

In this section we discuss how interface conditions can be enforced in a weak sense by means of an adapted discrete variational formulation.

The Nitsche formulation is one approach to tackle the problem. We derive it for our

problem setting in section 2.2.2.1. The basic components of the approach go back to the original paper [Nit71], in which a (one-phase) Poisson problem with homogeneous Dirichlet boundary conditions on a *fitted* boundary is considered. In order to homogenize the problem, one would need to know a sufficiently smooth function which fulfills the Dirichlet conditions. To avoid this, a variational principle is introduced to enforce the boundary condition in a weak sense.

The imposition of Dirichlet-type boundary or interface conditions on finite element spaces which do not respect the condition automatically is a well-known problem in the literature and several solution approaches exist. For instance, a *fitted* interface between two non-matching meshes across which a continuity condition should be prescribed is a common situation in domain decomposition methods. The mortar method is a popular way to deal with this problem. We mention the paper [HP02] where a Nitsche method is applied and analyzed in this context. In [CH11] such a problem for higher order finite elements is discussed.

The Nitsche approach for *unfitted* interface problems has been introduced in the seminal paper [HH02] for a problem without discontinuity. It has been generalized to the case with a Henry condition and (small) convection in [RN09].

The Nitsche formulation for a fictitious domain problem has been considered in [BBH11]. A nice overview on the Nitsche method for *fitted* and *unfitted* interfaces can be found in [Han05]. Another interesting overview paper (with a focus on high contrast problems) is [BZ12].

In this thesis we almost exclusively consider the use of the *unfitted* Nitsche method. In section 2.2.3 we discuss variants of it. Important alternatives to the Nitsche method are penalty methods and especially the Lagrange multiplier method. Both have a close relation to the Nitsche method. In section 2.2.3.3 and section 2.2.3.4 we briefly discuss the methods and their close relation to the Nitsche method.

2.2.2.1 Derivation of the Nitsche method

The enforcement of interface or boundary conditions on unfitted meshes can be achieved in several ways. One way to implement interface conditions is the Nitsche method which uses a *consistent penalization* to enforce the interface conditions. This is also our method of choice in this work. In this section we derive this method. At some places during the derivation several choices can be made. In this section we always use the “standard” choices made in the literature. Afterwards we discuss several variants resulting from different choices and alternatives to the Nitsche approach which however are (closely) related.

We derive the Nitsche method for the model problem, problem 2.1.3 and assume (for simplicity) homogeneous Dirichlet conditions $g_D = 0$. For now, we assume that a smooth solution to problem 2.1.3 exists and fulfills $u \in H^2(\Omega_{1,2})$, s.t. all appearing differentials

exist at least in a weak sense. As usual in the context of finite element methods, we test equation (2.1.3a) with an arbitrary function from our finite element space $v \in V_h^\Gamma$. As u and v may be discontinuous, integration is done domain-wise. We use the β -weighted scalar product introduced in (2.1.7). We thus start with

$$(-\operatorname{div}(\alpha \nabla u), v)_0 = (f, v)_0 =: \langle f, v \rangle \quad (2.2.4)$$

where we assume $f \in L^2(\Omega)$ such that the duality pairing $\langle \cdot, \cdot \rangle$ between $H^{-1}(\Omega_{1,2})$ and $H_0^1(\Omega_{1,2})$ reduces to the scalar product $(\cdot, \cdot)_0$. Applying partial integration we get using (2.1.8) and $V_h^\Gamma \subset H_0^1(\Omega_{1,2})$

$$a(u, v) - \sum_{i=1,2} \int_{\partial\Omega_i \setminus \partial\Omega} \alpha \nabla u \cdot \mathbf{n} \beta v \, d\mathbf{s} = \langle f, v \rangle. \quad (2.2.5)$$

For integrals on the interface we introduce the scalar products

$$(f, g)_\Gamma := \int_\Gamma f g \, d\mathbf{s}, \quad (f, g)_{\pm \frac{1}{2}, h, \Gamma} := \sum_{T \in \mathcal{T}_h^\Gamma} h_T^{\mp 1} (f, g)_{\Gamma_T} \quad (2.2.6)$$

with correspondingly induced norms $\|\cdot\|_\Gamma$ and $\|\cdot\|_{\pm \frac{1}{2}, h, \Gamma}$. For the boundary terms stemming from partial integration there holds

$$- \sum_{i=1,2} \int_{\partial\Omega_i \setminus \partial\Omega} \alpha \nabla u \cdot \mathbf{n} \beta v \, d\mathbf{x} = \sum_{i=1,2} (-\alpha_i \nabla u_i \cdot \mathbf{n}, \beta_i v_i)_\Gamma \quad (2.2.7)$$

Due to (2.1.3b) we can replace $-\alpha_1 \nabla u_1 \cdot \mathbf{n}$ with $-\alpha_2 \nabla u_2 \cdot \mathbf{n}$ and vice versa or, what we do here, replace both with a unique value, which in the DG community is often called the *numerical flux* $\hat{\sigma}_\mathbf{n}$:

$$\hat{\sigma}_\mathbf{n} = -\{\!\!\{\alpha \nabla u \cdot \mathbf{n}\}\!\!\} = -(\kappa_1 \alpha_1 \nabla u_1 + \kappa_2 \alpha_2 \nabla u_2) \cdot \mathbf{n}_1 \quad (2.2.8)$$

with $\kappa_1 + \kappa_2 = 1$ where κ_i , $i = 1, 2$ is typically defined in an element-wise fashion. The choice of κ_i is an important issue w.r.t. the stability of the formulation and is discussed in section 2.2.2.2. We define

$$N_c : H^2(\mathcal{T}_h^{1,2}) \times H^1(\Omega_{1,2}) \rightarrow \mathbb{R}, \quad N_c(u, v) := -(\{\!\!\{\alpha \nabla u \cdot \mathbf{n}\}\!\!\}, \llbracket \beta v \rrbracket)_\Gamma \quad (2.2.9)$$

where $H^k(\mathcal{T}_h^{1,2}) := \bigcup_{i=1,2} \bigcup_{T \in \mathcal{T}_h^i} H^k(T \cap \Omega_i)$ and arrive at

$$a(u, v) + N_c(u, v) = \langle f, v \rangle$$

In contrast to the continuous formulation this formulation is no longer symmetric. In order to retain the symmetry of the continuous problem we add the symmetrical counterpart of $N_c(\cdot, \cdot)$ and have

$$a(u, v) + N_c(u, v) + N_c(v, u) = \langle f, v \rangle \quad (2.2.10)$$

Note that this is also consistent as due to $[\![\beta u]\!] = 0$ in (2.1.3c) for the solution u we have $N_c(v, u) = 0$. The bilinear form corresponding to the left hand side is now consistent and symmetric. To make the corresponding bilinear form also coercive we need to add another integral term, a stabilization term

$$N_s : H^1(\Omega_{1,2}) \times H^1(\Omega_{1,2}) \rightarrow \mathbb{R}. \quad (2.2.11)$$

This is added in order to control the interface jump $[\![\beta u]\!]$. There are several variants on how to choose $N_s(\cdot, \cdot)$. The most common stabilization in the case of an unfitted interface is the one proposed in [HH02], obtained by adding the mesh-dependent bilinear form

$$N_s^H(u, v) := \left(\frac{\lambda}{h} \bar{\alpha} [\![\beta u]\!], [\![\beta v]\!]\right)_\Gamma. \quad (2.2.12)$$

If not addressed otherwise we set $N_s(\cdot, \cdot) = N_s^H(\cdot, \cdot)$. This additional term is again consistent due to $[\![\beta u]\!] = 0$ in (2.1.3c). Note that $N_s(\cdot, \cdot)$ is not scaled with β which is not a problem due to assumption 1.2.2 ($\beta_i \geq 1$).

Here, λ is the stabilization parameter which has to be chosen larger than a constant depending on the shape regularity and the polynomial degree. This is due to an inverse trace inequality that is applied to bound the normal derivatives in $N_c(\cdot, \cdot)$ by the stabilization form and the domain-wise H^1 -norm. For details see section 2.2.2.2.

Putting all terms together we define the mesh-dependent bilinear form

$$\begin{aligned} a_h : H^2(\mathcal{T}_h^{1,2}) \cap H^1(\Omega_{1,2}) \times H^2(\mathcal{T}_h^{1,2}) \cap H^1(\Omega_{1,2}) &\rightarrow \mathbb{R} \\ a_h(u, v) &:= a(u, v) + N_c(u, v) + N_c(v, u) + N_s(u, v) \end{aligned} \quad (2.2.13)$$

and have $a_h(u, v) = \langle f, v \rangle$ for every $v \in V_h^\Gamma$. Accordingly we denote the following discrete problem as the Nitsche-XFEM discretization of problem 2.1.3:

Find $u_h \in V_h^\Gamma$, s.t.

$$a_h(u_h, v_h) = \langle f, v_h \rangle \quad \forall v_h \in V_h^\Gamma \quad (2.2.14)$$

Remark 2.2.2 (Stabilized numerical flux). *Adding $N_s(\cdot, \cdot)$ to (2.2.10) can also be viewed as a change in the numerical flux $\hat{\sigma}_\mathbf{n}$ in (2.2.8) to the stabilized numerical flux*

$$\hat{\sigma}_\mathbf{n} = -\{\!\!\{\alpha \nabla u \cdot \mathbf{n}\}\!\!\} + \lambda \frac{\bar{\alpha}}{h} [\![\beta u]\!]. \quad (2.2.15)$$

This choice is equivalent to the numerical flux of the interior penalty method [DD76] in the context of Discontinuous Galerkin methods. See [ABCM02] for a nice overview on choices for the numerical flux. This choice is also important for the stabilized Lagrange multiplier formulation, cf. section 2.2.3.4. In section 2.2.7 we show a conservation property of the Nitsche-XFEM discretization w.r.t. the flux $\hat{\sigma}_\mathbf{n}$.

2.2.2.2 Weighted average and the choice of λ

In the derivation of the Nitsche formulation we introduced the weighted average $\{\!\!\{\cdot\}\!\!\}$ with weights κ_i , $i = 1, 2$. For the consistency of the method any convex combination can be

applied. Nevertheless the choice of the weights influences how well the non-symmetric term $N_c(u, v)$ for $u = v$ can be bounded by $a(u, u)$ and $N_s(u, u)$. This is important for the stability of the method. The crucial point is that the following inverse inequality for discrete functions u_h with $u_h|_{T_i} \in \mathcal{P}^k(T_i)$, $T \in \mathcal{T}_h$, $i = 1, 2$, holds:

$$\kappa_i^2 \int_{\Gamma_T} h \|\nabla u_h\|^2 ds \leq c_{\text{tr}} \int_{T_i} \|\nabla u_h\|^2 d\mathbf{x}, \quad \forall T \in \mathcal{T}_h, i = 1, 2, \quad (2.2.16)$$

with c_{tr} a constant that only depends on the shape regularity of T (not on the shape regularity of T_i !). The validity of the inequality, however, depends on the choice of κ_i . A typical choice for κ_i for the case of piecewise linear functions for which the inequality holds (see section 2.3.1.3 for details) has been introduced in [HH02].

Definition 2.2.2. We denote the averaging operator $\{\!\!\{v\}\!\!\}^H := \kappa_1^H v_1 + \kappa_2^H v_2$ with $\kappa_i^H = \frac{|T_i|}{|T|}$ as the HANSBO-averaging.

The HANSBO-averaging will be our standard choice and if averaging is not addressed specifically we set $\{\!\!\{v\}\!\!\} = \{\!\!\{v\}\!\!\}^H$.

If the HANSBO-averaging is applied, the constant c_{tr} in (2.2.16) depends only on the shape regularity of T . The stabilization parameter λ has to be chosen larger than a constant only depending on c_{tr} (see section 2.3.1.3 for details). It is thus relevant to know the range in which c_{tr} lies. For simple geometries an explicit description of c_{tr} can be given. In practice however, λ is typically chosen on the safe side. The benefit of an increasing λ is two-fold. First, for a sufficiently large λ stability of the discretization can be ensured. Second, the error in the interface condition is essentially determined by λ and the mesh resolution. Hence, a large λ leads to a small error in the interface condition. The only drawback of a large λ is an increase in the condition number (see section 2.4 and section 2.5.3.5 for details). A compromise is typically to choose λ one order of magnitude larger than necessary for stability. In section 2.2.3.2 we present a modification of the Nitsche discretization which is stable and has no such parameter as λ .

2.2.3 Variants of and alternatives to the Nitsche formulation

In this section we want to put the presented Nitsche formulation in context to other related methods in the literature.

A careful look at the derivation of the Nitsche formulation displays that the Nitsche formulation allows for several modifications. E.g. the choice of the numerical flux $\hat{\sigma}_{\mathbf{n}}$, the averaging operator $\{\!\!\{ \}$ or the choice to aim for a symmetric formulation are, although justified, neither necessary nor essential. In section 2.2.3.1 and section 2.2.3.2 we will discuss two modifications of the Nitsche formulation which are parameter-free, that means that they do not depend on a parameter like λ which has to be chosen “sufficiently large” in order to guarantee stability.

In section 2.2.3.4 we briefly present the method of Lagrange multipliers as an alternative to the Nitsche formulation. We also highlight its close relation to the Nitsche formulation.

In [BZ12] it was pointed out that the stability of the Nitsche formulation as presented in the last section relies on the HANSBO-averaging and thereby contradicts with other weighted averages as they are relevant for high contrast problem. A way to overcome stability problems (and conditioning problems) is to add another consistent stabilization which ensures control on the gradient of u independent on the cut position. This is done with the so called “Ghost penalty” method introduced in a series of paper by Burman et al. [Bur10, BH10, BH12, BZ12]. This and a similar approach are briefly discussed in section 2.2.3.5.

2.2.3.1 Non-symmetric formulations

In applications, especially when simulating coupled problems, it is desirable to reduce the number of free parameters. The Nitsche method presented above however has the stabilization parameter λ .

In view of stability the simplest modification of the Nitsche-XFEM method presented before is to replace $N_c(v, u)$ by $-N_c(v, u)$. Then one has (with $N_s(\cdot, \cdot) = N_s^H(\cdot, \cdot)$)

$$a_h(u, u) = a(u, u) + N_c(u, u) - N_c(u, u) + N_s(u, u) = a(u, u) + N_s(u, u)$$

which already implies coercivity of $a_h(u, u)$ with respect to the norm $(a(u, u) + N_s(u, u))^{\frac{1}{2}}$. This approach has already been discussed (for one-domain problems with matched boundaries) in [FS95]. Note that the statement is true independent of the choice of λ such that we can fix $\lambda = 1$ independent of the shape regularity of the triangulation $\mathcal{T}_h$.

This modification renders the bilinear form $a_h(\cdot, \cdot)$ non-symmetric. That again results in the fact that the bilinear form is not adjoint consistent which means that the adjoint of $a_h(\cdot, \cdot)$ does not correspond to a consistent discretization of the continuous adjoint problem which coincides with the original problem (as the problem is self-adjoint). The lack of adjoint consistency results in a sub-optimality in the L^2 -norm for the a priori error analysis *and* for the practical results.

In cases however where the adjoint problem does not possess high regularity estimates w.r.t. the data, the lack of adjoint consistency does not weight so much. This is especially the case if convection is present and dominant.

It turns out that the penalty term $N_s(u, v)$ can be dropped completely ($\lambda = 0$) which can also be favorable in special applications. A detailed discussion of the method and its error analysis can be found in [Bur12].

Another approach which allows to remove the free parameter λ while keeping the formulation symmetric is presented in the next section. The approach is inspired by Discontinuous Galerkin (DG) methods.

2.2.3.2 Minimal stabilization

One disadvantage of the Nitsche formulation as presented above is the fact that the penalty parameter has to be chosen “sufficiently large”. Although the condition on an upper bound for a minimal λ can be derived, the parameter is typically chosen to be “on the safe side”. Especially if the mesh is less regular and jumps in the parameters get larger or the polynomial degree of the discretization is higher, the minimal choice for λ might be less obvious. The modification presented in this section gets rid of the stabilization parameter λ by adding an “indirect” stabilization. This approach is inspired by a method for Discontinuous Galerkin (DG) discretizations. Discontinuous Galerkin discretizations for elliptic problems need to weakly impose continuity. This is similar to introducing the interface condition for the XFEM space V_h^Γ . In this context the Nitsche method derived in the preceding sections for our problem is an analogue to the symmetric interior penalty method for DG discretizations. For DG discretizations a number of other methods exist to enforce continuity in a weak sense, see [ABCM02] for a nice overview of different methods. The subsequent method is based on the DG method presented in [BR97, BRM⁺97] and analyzed in [BMM⁺99].

First, we extend the previous definition of the bilinear form $a(\cdot, \cdot)$ from (2.1.8) to functions from element-wise broken Sobolev spaces.

$$a : H^1(\mathcal{T}_h^{1,2}) \times H^1(\mathcal{T}_h^{1,2}) \rightarrow \mathbb{R}, \quad a(u, v) := \sum_{T \in \mathcal{T}_h} (\alpha u, v)_{1, T_{1,2}} \quad (2.2.17)$$

On a cut element T we further introduce the element-wise lifting operator $\mathcal{L}$.

Definition 2.2.3 (Lifting $\mathcal{L}$). *We define the lifting*

$$\mathcal{L} : H^1(\mathcal{T}_h^{1,2}) \rightarrow W_h^* := \{u|_{T_i} \in \mathcal{P}^k \cap (\mathcal{P}^0)^\perp, T \in \mathcal{T}_{h,i}, i = 1, 2\}$$

by its element contributions. Let T be a cut element, $T \in \mathcal{T}_h^\Gamma$. We define $\mathcal{L}^T : H^1(T_{1,2}) \rightarrow \{u|_{T_i} \in \mathcal{P}^k \cap (\mathcal{P}^0)^\perp, i = 1, 2\} =: W_T^*$, such that $w := \mathcal{L}^T(u)$ is the unique solution of

$$a^T(w, v_h) := (\alpha u, v)_{1, T_{1,2}} = N_c^T(v_h, u) := -(\llbracket \beta u \rrbracket, \llbracket \alpha \nabla v_h \cdot \mathbf{n} \rrbracket)_{\Gamma_T} \quad \forall v_h \in W_h^*. \quad (2.2.18)$$

On uncut elements we set $\mathcal{L}^T(u) = 0$ and thus have for every $u \in H^1(\mathcal{T}_h^{1,2})$

$$a(\mathcal{L}(u), v_h) = N_c(v_h, u), \quad \forall v_h \in V_h^\Gamma.$$

Using this lifting operator for $u_h, v_h \in V_h^\Gamma$ yields

$$a(u_h, v_h) + N_c(u_h, v_h) + N_c(v_h, u_h) + a(\mathcal{L}(u_h), \mathcal{L}(v_h)) = a(u_h + \mathcal{L}(u_h), v + \mathcal{L}(v_h)). \quad (2.2.19)$$

We immediately get for $u_h \in V_h^\Gamma$

$$2N_c(u_h, u_h) = 2a(\mathcal{L}(u_h), u_h) \leq 2a(\mathcal{L}(u_h), \mathcal{L}(u_h)) + \frac{1}{2}a(u_h, u_h). \quad (2.2.20)$$

This motivates the following choice for the stabilizing bilinear form $N_s(\cdot, \cdot)$

$$N_s^\mathcal{L}(u, v) = 2a(\mathcal{L}(u), \mathcal{L}(v)) + \bar{\alpha} \|\llbracket \beta u \rrbracket\|_{\frac{1}{2}, h, \Gamma}^2 \quad (2.2.21)$$

The first term is introduced to guarantee non-negativeness of the bilinear form on V_h^Γ (using (2.2.20)) whereas the second term is introduced in order to add explicit control on the jump $\llbracket \beta u \rrbracket$. Note that no generic constants or tuning parameters (e.g. λ) appear which is an advantage of the method.

Remark 2.2.3. *The functional $\mathcal{F} : H^1(\Omega_{1,2}) \rightarrow \mathbb{R}$, $\mathcal{F}(u) := \sqrt{a(u, u) + a(\mathcal{L}(u), \mathcal{L}(u))}$ has the kernel $\{u|_{\Omega_i} = \text{const}\}$ and thus does not define a norm.*

We define

$$a_h^\mathcal{L}(u, v) := a(u, v) + N_c(u, v) + N_c(v, u) + N_s^\mathcal{L}(u, v) \quad (2.2.22)$$

and the norm

$$\|u\|_\mathcal{L}^2 := a(u, u) + a(\mathcal{L}(u), \mathcal{L}(u)) + \bar{\alpha} \|\llbracket \beta u \rrbracket\|_{\frac{1}{2}, h, \Gamma}^2. \quad (2.2.23)$$

Using the relations from above you can show

$$\|u\|_\mathcal{L}^2 \preceq a_h^\mathcal{L}(u, u) \preceq \|u\|_\mathcal{L}^2 \quad \forall u \in V_h^\Gamma$$

where the constants for the left and right inequality in this case are bounded by the factor three.

Controlling the lifting norm. One can bound $a(\mathcal{L}(u), \mathcal{L}(u))$ by a constant times $\|\llbracket \beta u \rrbracket\|_{\frac{1}{2}, h, \Gamma}$ for $u \in H^1(\mathcal{T}_h^{1,2})$, if the HANSBO-averaging is applied. Then, we get, using the inverse estimate in (2.2.16), and standard estimates

$$\begin{aligned} a(\mathcal{L}(u), \mathcal{L}(u)) &= N_c(\mathcal{L}(u), u) \leq \bar{\alpha} c_{\text{tr}} \|\llbracket \beta u \rrbracket\|_{\frac{1}{2}, h, \Gamma}^2 + \frac{1}{2}a(\mathcal{L}(u), \mathcal{L}(u)) \quad \forall u \in H^1(\mathcal{T}_h^{1,2}) \\ \text{and thus } a(\mathcal{L}(u), \mathcal{L}(u)) &\leq 2c_{\text{tr}} \bar{\alpha} \|\llbracket \beta u \rrbracket\|_{\frac{1}{2}, h, \Gamma}^2 \quad \forall u \in H^1(\mathcal{T}_h^{1,2}). \end{aligned}$$

It follows

$$a_h^\mathcal{L}(u, u) \simeq \|u\|_\mathcal{L}^2 \simeq a(u, u) + \bar{\alpha} \|\llbracket \beta u \rrbracket\|_{\frac{1}{2}, h, \Gamma}^2, \quad u \in V_h^\Gamma \quad (2.2.24)$$

with constants only depending on c_{tr} . Note that due to the normal derivative in $N_c(\cdot, \cdot)$ (2.2.24) does not hold for $u \in H^1(\Omega_{1,2})$.

Implementation aspects. To implement the element-local lifting $w = \mathcal{L}^T(u)$ of a local finite element function u , we solve for w with $u, w \in \{u|_{T_i} \in \mathcal{P}^k\}$, such that

$$a^T(w, v) + \sum_{i=1,2} k_i^T(w, v) = N_c^T(v, u), \quad \forall v \in \{u|_{T_i} \in \mathcal{P}^k\} \quad (2.2.25)$$

with the bilinear form

$$k_i^T(w, v) := h^{-(d+2)} (w, 1)_{T_i} (v, 1)_{T_i}$$

which is tailored to eliminate the kernel $\{u|_{T_i} = \text{const}\}$. We need to compute the element matrix $\mathbf{K}$ corresponding to $\sum_{i=1,2} k_i^T(\cdot, \cdot)$. Note that the local element matrices $\mathbf{A}$ and $\mathbf{N}_c$ corresponding to the bilinear form $a^T(\cdot, \cdot)$ and $N_c^T(\cdot, \cdot)$ have to be computed anyway. We thus get the coefficients $\mathbf{w}$ of the local lifting ($\mathbf{w}_i = \mathcal{L}^T(\varphi_i)$) as

$$\mathbf{w} = \mathbf{L} \cdot \mathbf{u} \quad \text{with } \mathbf{L} = (\mathbf{A} + \mathbf{K})^{-1} \mathbf{N}_c^T.$$

The overall element contribution to the bilinear form $a_h(\cdot, \cdot)$ in matrix notation is:

$$\mathbf{A} + \mathbf{N}_c + \mathbf{N}_c^T + 2 \cdot \mathbf{L}^T \mathbf{A} \mathbf{L} + \mathbf{N}_s$$

where $\mathbf{N}_s$ is the element matrix corresponding to $N_s^T(\cdot, \cdot) = (\frac{\bar{\alpha}}{h} \llbracket \beta \cdot \rrbracket, \llbracket \beta \cdot \rrbracket)_{\Gamma_T}$.

2.2.3.3 Penalty methods

A very early approach to enforce Dirichlet boundary conditions in a weak sense is to replace the boundary conditions with similar ones which allow a simple integration into a weak form. In our context such a boundary condition would be

$$-\alpha_i \nabla u_i \cdot \mathbf{n}_1 = h^{-\rho} \bar{\alpha} \llbracket \beta u \rrbracket \quad \text{on } \Gamma$$

for $\rho > 0$. The corresponding discrete weak formulation would then be:
Find $u \in V_h^\Gamma$ so that

$$a(u, v) + \int_{\Gamma} h^{-\rho} \bar{\alpha} \llbracket \beta u \rrbracket \llbracket \beta v \rrbracket d\mathbf{s} = \langle f, v \rangle \quad v \in V_h^\Gamma$$

Due to the change in the interface condition, this formulation introduces a consistency error. However, for different values of ρ the consistency error vanishes fast enough to obtain optimal error bounds at least in some norms (cf. [Bab73a, BE86]). However a choice for ρ which gives optimal error estimates in all norms comes at the price of ill-conditioned system matrices. The Nitsche method can be seen as a *consistent* variant of the penalty method with $\rho = 1$.

2.2.3.4 The method of Lagrange multipliers

The method of Lagrange multipliers to implement Dirichlet boundary conditions has originally been introduced in [Bab73b]. In the context of fictitious domain methods the Lagrange multiplier method has been applied (among others) in [GPP94a, GPP94b, GG95, BH10]. We briefly introduce the method in our two-domain context. We again assume $u \in H^2(\Omega_{1,2})$, $v \in V_h^\Gamma$, and start from (2.2.5)

$$(\alpha u, v)_{1,\Omega_{1,2}} - \sum_{i=1,2} \int_{\partial\Omega_i \setminus \partial\Omega} \alpha \nabla u \cdot \mathbf{n} \beta v \, ds = \langle f, v \rangle \quad (2.2.5)$$

Now we introduce a new variable, the flux $\sigma_{\mathbf{n}} := -\alpha \nabla u \cdot \mathbf{n}$ and get

$$(u, v)_{1,\Omega_{1,2}} + \int_{\Gamma} \sigma_{\mathbf{n}} \llbracket \beta v \rrbracket \, ds = \langle f, v \rangle \quad (2.2.26)$$

To impose the interface condition (2.1.3c) we multiply $\llbracket \beta u \rrbracket = 0$ by sufficiently many test functions μ and integrate over Γ :

$$(\llbracket \beta u \rrbracket, \mu)_{\Gamma} = 0 \quad \forall \mu \in Q \quad (2.2.27)$$

with Q to be determined later. Combining both, we define the discrete problem as: Find $(u, \sigma_{\mathbf{n}}) \in V_h^\Gamma \times Q$, such that

$$(\alpha u, v)_{1,\Omega_{1,2}} + (\sigma_{\mathbf{n}}, \llbracket \beta v \rrbracket)_{\Gamma} = \langle f, v \rangle \quad \forall u \in V_h^\Gamma \quad (2.2.28a)$$

$$(\llbracket \beta u \rrbracket, \mu)_{\Gamma} = 0 \quad \forall \mu \in Q. \quad (2.2.28b)$$

This is a saddle point problem which can also be written as: Find $(u, \sigma_{\mathbf{n}}) \in V_h^\Gamma \times Q$, such that

$$K((u, \sigma_{\mathbf{n}}), (v, \mu)) = a(u, v) + b(u, \mu) + b(v, \sigma_{\mathbf{n}}) = \langle f, v \rangle \quad \forall (v, \mu) \in V_h^\Gamma \times Q$$

where

$$b : H^1(\Omega_{1,2}) \times L^2(\Gamma) \rightarrow \mathbb{R}, \quad b(v, \mu) = (\llbracket \beta v \rrbracket, \mu)_{\Gamma}.$$

Note that $a(\cdot, \cdot)$ is elliptic on the kernel of $b(\cdot, \cdot)$. A crucial condition for a stable discretization is the discrete “inf-sup”-condition:

$$\sup_{v \in V_h^\Gamma} \frac{b(v, \mu)}{\|v\|_{1,h}} \geq c \|\mu\|_{-\frac{1}{2},h,\Gamma} \quad \forall \mu \in Q \quad (2.2.29)$$

for a $c > 0$ independent on h where $\|v\|_{1,h}^2 := |v|_{1,\Omega_{1,2}}^2 + \|\llbracket \beta v \rrbracket\|_{\frac{1}{2},h,\Gamma}^2$. In a series of papers by Pitkäranta [Pit79, Pit80, Pit81] this problem (with only one phase) has been studied in detail and it was shown that in order to achieve optimal order of convergence of the method the space Q has to be chosen very carefully. For example choosing piecewise linear functions for V_h^Γ and piecewise linears on the interface for Q leads to an unstable

discretization. It turns out that constructing a suitable space Q is an involved procedure which raises the question of the practical use of the method. To overcome this problem suitable modifications of the method have been proposed in the literature. In [Ver91] and [Ste95] the close connection between a modified (“stabilized”) Lagrange multiplier method and the Nitsche method has been pointed out in the context of fitted one-domain problems. These ideas have a natural extension to our two-domain problem. In the next section we present a “stabilized” Lagrange multiplier method which reveals a close relation to the formulation derived before. In the recent publication [Bur14] a more general approach based on similar ideas is discussed.

A Stabilized Lagrange multiplier formulation and the relation to the Nitsche method.

The discrete “inf-sup”-condition in (2.2.29) is in general hard to fulfill. Further, already the saddle-point structure of the Lagrange multiplier formulation is, from a computational point of view, a drawback of the method. In order to circumvent both, one can introduce another consistent term which couples $\sigma_{\mathbf{n}}$ and μ and allows to eliminate the unknown $\sigma_{\mathbf{n}}$.

The coupling between μ and $\sigma_{\mathbf{n}}$ is introduced by adding the symmetric bilinear form

$$\begin{aligned} d : (H^2(\mathcal{T}_h^{1,2}), L^2(\Gamma)) \times (H^2(\mathcal{T}_h^{1,2}), L^2(\Gamma)) &\rightarrow \mathbb{R}, \\ d((u, \sigma_{\mathbf{n}}), (v, \mu)) &:= -\frac{\delta}{\bar{\alpha}}(h(\sigma_{\mathbf{n}} - \hat{\sigma}_{\mathbf{n}}(u)), \mu - \hat{\sigma}_{\mathbf{n}}(v))_{\Gamma} \end{aligned} \quad (2.2.30)$$

with a small stabilization parameter $\delta = \text{const}$ and $\hat{\sigma}_{\mathbf{n}}(w) = -\{\!\!\{\alpha \nabla w \cdot \mathbf{n}\}\!\!\}$ the (unstabilized) *numerical flux* as in (2.2.8). In this discretization $\frac{1}{\delta}$ takes the role of λ in the Nitsche formulation. By construction $\sigma_{\mathbf{n}} - \hat{\sigma}_{\mathbf{n}}(u)$ vanishes for the true solution.

We can now solve the modified version of (2.2.28b) for $\sigma_{\mathbf{n}}$. We have

$$b(\llbracket \beta u \rrbracket, \mu) + d((u, \sigma_{\mathbf{n}}), (0, \mu)) = 0 \quad \forall \mu \in Q$$

and can thus express $\sigma_{\mathbf{n}}$ in terms of u :

$$\sigma_{\mathbf{n}} = \Pi_Q(-\{\!\!\{\alpha \nabla u \cdot \mathbf{n}\}\!\!\} + \frac{\bar{\alpha}}{\delta h} \llbracket \beta u \rrbracket) \quad (2.2.31)$$

where Π_Q is the $L^2(\Gamma)$ -projection into the space Q . If Q is element-wise discontinuous this projector is element-local. Substituting $\sigma_{\mathbf{n}}$ into (2.2.28a) we get the discrete problem: Find $u \in V_h^\Gamma$ such that

$$(\alpha u, v)_{1, \Omega_{1,2}} - (\{\!\!\{\alpha \nabla u \cdot \mathbf{n}\}\!\!\}, \Pi_Q(\llbracket \beta v \rrbracket))_{\Gamma} - (\{\!\!\{\alpha \nabla v \cdot \mathbf{n}\}\!\!\}, \Pi_Q(\llbracket \beta u \rrbracket))_{\Gamma} \quad (2.2.32)$$

$$+ (\frac{\bar{\alpha}}{\delta h} \Pi_Q \llbracket \beta u \rrbracket, \Pi_Q \llbracket \beta v \rrbracket)_{\Gamma} + (\frac{\delta h}{\bar{\alpha}} \Pi_Q^\perp \{\!\!\{\alpha \nabla u \cdot \mathbf{n}\}\!\!\}, \Pi_Q^\perp \{\!\!\{\alpha \nabla v \cdot \mathbf{n}\}\!\!\})_{\Gamma} \quad (2.2.33)$$

$$= \langle f, v \rangle \quad \forall v \in V_h^\Gamma \quad (2.2.34)$$

with $\Pi_Q^\perp = \Pi_Q - I$.

Note that we no longer need the pair (V_h^Γ, Q) to fulfill an inf-sup-condition and we can choose $Q = \text{tr}_{|\Gamma} V_h$. Hence, we can replace Π_Q with the identity and $\Pi_Q^\perp$ with zero. Additionally choosing $\delta = \frac{1}{\lambda}$ and $\{\!\!\{\!\!\} = \{\!\!\{\!\!\}^H$ results in

$$a(u, v) + N_c(u, v) + N_c(v, u) + N_s(u, v) = \langle f, v \rangle \quad \forall v \in V_h^\Gamma \quad (2.2.35)$$

which is exactly the Nitsche discretization derived before.

Remark 2.2.4 (Characterization of the flux for Nitsche-XFEM). *An advantage of the Lagrange multiplier method is that the flux $\sigma_{\mathbf{n}}$ is an explicit unknown. The above derivation of the Nitsche-XFEM discretization however reveals that we can use (2.2.31) to reconstruct a conservative flux also for the Nitsche discretization. See also section 2.2.7 for a discussion on the conservation properties of the Nitsche-XFEM discretization.*

2.2.3.5 Ghost penalty

In a series of papers [Bur10, BH10, BH12, BZ12] Burman et al. suggested an additional stabilization mechanism which enhances the robustness of the Nitsche formulation w.r.t. the interface cut position. The stability of the method derived before relies on the choice of the averaging operator $\{\!\!\{\!\!\}$ where we considered the HANSBO-choice as a good choice. For this discretization the condition number of the system matrix is *not* independent on the cut position and can get arbitrarily bad. In section 2.4 we discuss this issue and demonstrate that this issue can easily be solved with diagonal preconditioning. This result however also depends on the choice of the averaging-operator.

The “ghost penalty” stabilization (cf. [Bur10]) was originally designed for implementing Dirichlet boundary conditions in the fictitious domain method. Note that for the fictitious domain method a stable imposition of Dirichlet boundary conditions is even more difficult as there is no averaging operator which helps to ensure stability. In the two-domain context the “ghost penalty” stabilization is interesting in cases where the weights of the averaging operator should be significantly different from the HANSBO-choice, for instance for large contrast problems (see [BZ12]). In this case the Nitsche-XFEM discretization lacks stability (and suffers from arising ill-conditioned linear systems).

By introducing an additional term, the “ghost penalty” stabilization releases the averaging operator from a constraint that has been necessary to ensure stability (essentially (2.2.16)). We briefly present the stabilization with the “ghost penalty” method for piecewise linear functions ($k = 1$).

We introduce the set of faces within the band of cut elements

$$\mathcal{F}_i^\Gamma := \{F = T_a \cap T_b, T_a \neq T_b, T_a, T_b \in \Omega_i^+, T_a \text{ or } T_b \in \mathcal{T}_h^\Gamma\}. \quad (2.2.36)$$

On this set we add the stabilization bilinear form

$$J(u_h, v_h) := \sum_{i=1,2} \sum_{F \in \mathcal{F}_i^\Gamma} \alpha_i \beta_i \gamma_J h_F (\llbracket \nabla \mathcal{E}_{i,h} u_h \cdot \mathbf{n} \rrbracket, \llbracket \nabla \mathcal{E}_{i,h} v_h \cdot \mathbf{n} \rrbracket)_F, \quad u_h, v_h \in V_h^\Gamma \quad (2.2.37)$$

with $\mathcal{E}_{i,h}$ the canonical extension from Ω_i to Ω_i^+ of discrete functions from V_h^Γ , $h_F = \max\{h_{T_a}, h_{T_b}\}$ where $F = T_a \cap T_b$ and γ_J the stabilization parameter. This additional term penalizes discontinuities in the derivative within the band of cut elements $\mathcal{T}_h^\Gamma$. Note that the penalty is imposed not only within the domains Ω_i , $i = 1, 2$ but also on the extension of the functions into Ω_i^+ . Therefore this stabilization term is independent on the cut position within the elements which gives the robustness of the method. The crucial point of the method is the following estimate.

Consider an element $T \in \mathcal{T}_h^\Gamma$ and assume that a neighbor $T_N \in \mathcal{T}_h \setminus \mathcal{T}_h^\Gamma$ with $T \cap T_N = F \neq \emptyset$ exists. We have (with c a generic constant and $i = 1, 2$) under the assumption of shape regularity ($|T| \leq c|T_N|$, $|T| \leq ch_F|F|$) for $u \in H^2(\mathcal{T}_{i,h})$

$$\|\nabla u_i \cdot \mathbf{n}\|_{-\frac{1}{2},h,\Gamma_T}^2 \leq |\Gamma_T| h_T \|\nabla u_i|_T\|_2^2 \quad (2.2.38a)$$

$$\leq c|T|(\|\nabla u_i|_{T_N}\|_2^2 + \|\llbracket \nabla u_i \rrbracket\|_2^2) \quad (2.2.38b)$$

$$\leq c(\|\nabla u_i\|_{L^2(T_N)}^2 + h_F \|\llbracket \nabla \mathcal{E}_{i,h} u_i \cdot \mathbf{n} \rrbracket\|_{L^2(F)}^2). \quad (2.2.38c)$$

The result can be generalized to arbitrary elements in $\mathcal{T}_h^\Gamma$ under reasonable (milder) assumptions, see [BH12] for details. In consequence this estimate states that the normal derivative on the interface can be controlled by the $|\cdot|_1$ -semi-norm and the stabilization term *independent of the cut position* and *independent of the averaging operator*.

Alternative stabilization. Another approach to improve the robustness of the Nitsche formulation w.r.t. the dependency on the cut position is discussed in [HR09]. In that paper a stabilized Lagrange multiplier approach (cf. section 2.2.3.4) for a fictitious domain problem is considered. In the consistent stabilization term that is added (cf. (2.2.30)) the normal derivative is replaced with a (weakly) consistent representative of the normal derivative. This is chosen such that forming the gradient on elements with small cuts is avoided.

2.2.4 The Nitsche-XFEM method with small convection

In the previous section the discretization of the simplified problem, problem 2.1.3 has been discussed. In this section we reintroduce convection and recall the stationary problem, problem 2.1.2:

$$\mathbf{w} \cdot \nabla u - \operatorname{div}(\alpha \nabla u) = f \quad \text{in } \Omega_i, \quad i = 1, 2, \quad (2.1.2a)$$

$$\llbracket \alpha \nabla u \cdot \mathbf{n} \rrbracket_\Gamma = 0 \quad \text{on } \Gamma, \quad (2.1.2b)$$

$$\llbracket \beta u \rrbracket_\Gamma = 0 \quad \text{on } \Gamma, \quad (2.1.2c)$$

$$u = g_D \quad \text{on } \partial\Omega. \quad (2.1.2d)$$

We remind that due to assumption 1.2.4 (compatible velocity) we have $\mathbf{w} \cdot \mathbf{n} = 0$ at the interface Γ .

Applying the Nitsche discretization for the diffusive part and adding the convection bilinear form

$$c(u, v) := (\mathbf{w} \cdot \nabla u, v)_{\Omega_{1,2}}, \quad u, v \in H^1(\Omega_{1,2}) \quad (2.2.40)$$

we get the following discrete problem as the Nitsche-XFEM discretization of problem 2.1.2: Find $u_h \in V_h^\Gamma$, s.t.

$$B_h(u_h, v_h) := a_h(u_h, v_h) + c(u_h, v_h) = \langle f, v_h \rangle \quad \forall v_h \in V_h^\Gamma. \quad (2.2.41)$$

An a priori error analysis of this discretization is presented in section 2.3.1.

2.2.5 The Nitsche-XFEM method with dominating convection

For large convection velocities $\|\mathbf{w}\|$ or small diffusion parameters α the approach introduced in the last section becomes unstable. This is not related to Nitsche or XFEM, but is already a problem of the Galerkin discretization for a one phase problem. We will show that a possible solution to this problem can be achieved by applying the Streamline Diffusion (SD) stabilization to the two-phase situation. In the next section we recall the main idea of the Streamline Diffusion stabilization for a one phase problem and afterwards extend it to the two-domain case.

In the convection dominated case the diffusion parameter is (at least after rescaling, s.t. $|\mathbf{w}|_\infty \leq \mathcal{O}(1)$) a small number. To emphasize this fact, in the literature of convection dominated problems the diffusion parameter is often denoted as ε and diffusion is seen as a singular perturbation to a (linear) hyperbolic equation. Hence, we identify $\varepsilon_i = \alpha_i$, $i = 1, 2$.

2.2.5.1 The Streamline Diffusion stabilization for a one phase problem

In one phase the stationary convection-diffusion problem is

Problem 2.2.1.

$$\mathbf{w} \cdot \nabla u - \operatorname{div}(\varepsilon \nabla u) = f \quad \text{in } \Omega \quad (2.2.42a)$$

$$u = 0 \quad \text{on } \partial\Omega. \quad (2.2.42b)$$

The Galerkin discretization of this problem is to find $u_h \in V_h$, s.t.

$$(\varepsilon u_h, v_h)_{1,\Omega} + (\mathbf{w} \cdot \nabla u_h, v_h)_{0,\Omega} = \langle f, v_h \rangle \quad \forall v_h \in V_h \quad (2.2.43)$$

For $\varepsilon \rightarrow 0$ this discretization lacks control on ∇u . Thus, if convection is dominant, the control that is obtained due to the symmetric part $(\varepsilon u_h, v_h)_{1,\Omega}$ degenerates. This results in stability problems. In many textbooks the problem is discussed, see for instance in [DH03, ESW05, RST08]. The Streamline Diffusion (SD) method stabilizes the Galerkin formulation to add additional control.

At the beginning of the eighties in [HB79, HB82] the Streamline-Upwind-Petrov-Galerkin (SUPG) method was introduced which has a similar stabilizing effect as “upwinding” schemes in finite volume and finite difference methods. The SD-method has a very close relation to the SUPG method and both methods can be identified with each other in some cases.

The essential idea of the SD-method is to add diffusion to the numerical scheme that scales with the dominating effect which is the convection. This additional diffusion however is, in contrast to artificial diffusion methods, added only in streamline direction and in a *consistent* way.

One adds a residual term of the form

$$\sum_{T \in \mathcal{T}_h} \gamma_T (\mathbf{w} \cdot \nabla u_h - \operatorname{div}(\varepsilon \nabla u_h) - f, \mathbf{w} \cdot \nabla v_h)_{0,T} \quad (2.2.44)$$

where γ_T is an element-wise defined stabilization parameter. Typical choices for γ_T can be found in (a.o.) [RST08, ESW05]. We take γ_T as follows:

$$\gamma_T = \begin{cases} \frac{2h_T}{|\mathbf{w}|_{\infty,T}} & \text{if } P_h^T > 1 \\ h_T^2/\varepsilon & \text{if } P_h^T \leq 1. \end{cases} \quad (2.2.45)$$

where we use the local Péclet number $P_h^T := \frac{1}{2}|\mathbf{w}|_{\infty,T}h_T/\varepsilon$. The motivation for this choice is as follows. For $P_h^T \leq 1$ no stabilization is necessary and the additional term should become small very rapidly. For $P_h^T > 1$ the stabilization term should scale as $c(\cdot, \cdot)$ (w.r.t. $\mathbf{w}$ and h), thus we set $\gamma_T \sim \frac{h_T}{|\mathbf{w}|_{\infty,T}}$. In practice several variants are used, e.g. if for the case $P_h^T \leq 1$ one sets $\gamma_T = 0$ very similar results (both in the theoretical analysis and in the experiments) are obtained.

2.2.5.2 Application of the SD stabilization for the two phase problem

Consider the basic (HANSBO) discretization of the diffusive part which led to the bilinear form

$$\begin{aligned} a_h(u, v) := & a(u, v) - ([\beta u], \{\{\varepsilon \nabla v \cdot \mathbf{n}\}\}_{\Gamma})_{\Gamma} \\ & - ([\beta v], \{\{\varepsilon \nabla u \cdot \mathbf{n}\}\}_{\Gamma})_{\Gamma} + \lambda \bar{\varepsilon} ([\beta u], [\beta v])_{\frac{1}{2}, h, \Gamma} \end{aligned} \quad (2.2.46)$$

where we recall $\bar{\varepsilon} = \frac{1}{2}(\varepsilon_1 + \varepsilon_2)$.

This discretization inherits the stability problems of the one phase Galerkin method from the last section. We thus add the Streamline Diffusion stabilization to the discretization. For the stabilization of the Nitsche-XFEM method we make obvious modifications related to the fact that in the XFEM space, close to the interface we have contributions on elements $T_i \neq T$, $i = 1, 2$. For the stabilization we introduce a locally weighted discrete variant of $(\cdot, \cdot)_0$:

$$(u, v)_{0,h} := \sum_{i=1}^2 \sum_{T \in \mathcal{T}_h} \beta_i \gamma_T \int_{T_i} uv \, d\mathbf{x} = \sum_{T \in \mathcal{T}_h} \gamma_T (u, v)_{0,T} \quad (2.2.47)$$

where we take γ_T as in (2.2.45) but replace ε with $\bar{\varepsilon}$. Note that the stabilization parameter γ_T does not depend on the position of the interface within the element.

We introduce the following Nitsche-XFEM discretization method with SD stabilization which will also be denoted as the SD-Nitsche-XFEM discretization:

Find $u_h \in V_h^\Gamma$ such that

$$\begin{aligned} a_h(u_h, v_h) + s^{\text{SD}}(u_h, v_h) + c(u_h, v_h) \\ = (f, v_h)_0 + (f, \mathbf{w} \cdot \nabla v_h)_{0,h} \quad \text{for all } v_h \in V_h^\Gamma. \end{aligned} \quad (2.2.48)$$

with

$$s^{\text{SD}}(u, v) := (-\text{div}(\varepsilon \nabla u) + \mathbf{w} \cdot \nabla u, \mathbf{w} \cdot \nabla v)_{0,h}. \quad (2.2.49)$$

In this discretization, λ is chosen as a “sufficiently large” constant. For stability considerations this constant only depends on the shape regularity of the mesh. The interface stabilization scales with the diffusion parameter. That means on a fixed spatial mesh for vanishing diffusion $\varepsilon \rightarrow 0$ that the enforcement of the interface conditions vanishes.

Assume convection dominates diffusion in the sense that the mesh Péclet number $P_h^T := \frac{1}{2}|\mathbf{w}|_{\infty,T}h_T/\bar{\varepsilon}$ is larger than one. The enforcement of the interface condition would get a small weight compared to the dominating effects in the subdomains.

The Streamline Diffusion stabilization essentially adds numerical diffusion (although in a consistent way) to the subdomains. This motivates to scale the interface stabilization with the convection rather than the diffusion parameter. Both, the scaling with diffusion as well as the scaling with convection can be justified. Furthermore the analysis in section 2.3.2 shows that the complete range between these scales allow for an “optimal” order (w.r.t. one-phase results in the literature) error analysis when the error inside the fluid domains is considered. Thus we formulate the following restriction on the Nitsche stabilization parameter λ_T which we allow to vary between elements.

$$\lambda^d \leq \lambda_T \leq \lambda_T^c := \lambda^d \max(P_h^T, 1) \quad (2.2.50)$$

with λ^d an $\mathcal{O}(1)$ constant only depending on shape regularity. The choice $\lambda_T = \lambda_T^c$ will be denoted as the *convective scaling* as in that case the stabilization term $([\![\beta u]\!], [\![\beta v]\!])_{\frac{1}{2},h,\Gamma}$ in (2.2.46) scales with $|\mathbf{w}|_\infty$ in the convection-dominated case $P_h \geq 1$. The other case $\lambda_T = \lambda^d$ will be denoted as the *diffusive scaling* for obvious reasons.

2.2.6 Time discretization for a stationary interface

We recall that the interface is *stationary*. Hence, from the discretization of the stationary problem 2.1.2 a corresponding semi-discretizations for problem 2.1.1 directly follows. We briefly present full discretizations obtained by applying the method of lines.

2.2.6.1 Diffusion dominates

Using the bilinear form $a_h(\cdot, \cdot)$ we define a method of lines discretization of (2.1.13). Let $\hat{u}_0 \in V_h^\Gamma$ be an approximation of u_0 . For $t \in [0, T]$ let $u_h(t) \in V_h^\Gamma$ be such that $u_h(0) = \hat{u}_0$ and

$$\left(\frac{du_h}{dt}, v_h\right)_0 + a_h(u_h, v_h) + c(u_h, v_h) = (f, v_h)_0 \quad \text{for all } v_h \in V_h^\Gamma. \quad (2.2.51)$$

As V_h^Γ does not depend on time, the semi-discretization in (2.2.51) is a system of ODEs. For simplicity we assume that $\mathbf{w}$ is constant in time (see also remark 2.2.5). We write this in matrix-vector notation and to this end introduce the following notation for matrices and vectors and the operator which relates coefficient vectors to finite element functions in a finite element space W .

Definition 2.2.4 (Galerkin isomorphism). *We define the Galerkin isomorphism with $\mathcal{G} : \mathbb{R}^n \rightarrow W$, $\mathcal{G}\mathbf{u} = \sum_{j=1}^n \mathbf{u}_j \varphi_j$, where φ_j are the basis functions of W so that $W = \text{span}\{\varphi_j\}_{j=1, \dots, n}$.*

The Galerkin isomorphism will be used (depending on the context) w.r.t. different finite element spaces, e.g. V_h , V_h^x or V_h^Γ without change of notation.

We define $\mathbf{M}, \mathbf{A}, \mathbf{C} \in \mathbb{R}^{N \times N}$, with $N = \dim(V_h^\Gamma)$ such that

$$\mathbf{v}^T \mathbf{M} \mathbf{u} := (\mathcal{G}\mathbf{u}, \mathcal{G}\mathbf{v})_0, \quad \mathbf{v}^T \mathbf{A} \mathbf{u} := a_h(\mathcal{G}\mathbf{u}, \mathcal{G}\mathbf{v}) \quad \text{and} \quad \mathbf{v}^T \mathbf{C} \mathbf{u} := c(\mathcal{G}\mathbf{u}, \mathcal{G}\mathbf{v}).$$

Further we define $\mathbf{f}$ so that $\mathbf{v}^T \mathbf{f} = (f, \mathcal{G}\mathbf{v})_0$. We thus have with $\mathbf{u}(0) = \mathcal{G}^{-1} \hat{u}_0$

$$\partial_t \mathbf{M} \mathbf{u} + \mathbf{A} \mathbf{u} + \mathbf{C} \mathbf{u} = \mathbf{f}, \quad t \in [0, T]$$

or

$$\partial_t \mathbf{u} = \mathbf{F}(\mathbf{u}, t) = -\mathbf{M}^{-1}(\mathbf{A} + \mathbf{C})\mathbf{u} + \mathbf{M}^{-1}\mathbf{f}, \quad t \in [0, T].$$

The ODE system $\partial_t \mathbf{u} = \mathbf{F}(\mathbf{u}, t)$ is stiff and allows for the application of standard time integration techniques such as Runge-Kutta methods, multi-step methods or the simple θ -method. The simplest discretization is obtained by applying the θ -scheme and results in (for a constant in time $\mathbf{f}$)

$$(\mathbf{M} + \theta \Delta t (\mathbf{A} + \mathbf{C})) \mathbf{u}^n = \mathbf{M} \mathbf{u}^{n-1} - (1 - \theta) \Delta t (\mathbf{A} + \mathbf{C}) \mathbf{u}^{n-1} + \Delta t \mathbf{f}, \quad t \in [0, T] \quad (2.2.52)$$

with the time step size $\Delta t = t_n - t_{n-1}$ where the time levels within on time steps are denoted as t_{n-1} , t_n . The discrete solutions are marked with corresponding superscripts.

2.2.6.2 Convection dominates

If we add the Streamline Diffusion stabilization in a non-stationary context, the stabilization term changes to

$$(\partial_t u_h + \mathbf{w} \cdot \nabla u_h - \text{div}(\varepsilon \nabla u_h) - f, w \cdot \nabla v_h)_{0,h} \quad (2.2.53)$$

and the corresponding semi-discretization is

$$\begin{aligned} & (\partial_t u_h, v_h)_0 + (\partial_t u_h + \mathbf{w} \cdot \nabla u_h - \operatorname{div}(\varepsilon \nabla u_h) - f, w \cdot \nabla v_h)_{0,h} + a_h(u_h, v_h) \\ & = (f, v_h)_0 + (f, \mathbf{w} \cdot \nabla v_h)_{0,h} \quad \text{for all } v_h \in V_h^\Gamma, t \in [0, T]. \end{aligned} \quad (2.2.54)$$

Clearly, this semi-discretization can also be combined with standard methods for time discretization to obtain a fully discrete problem. We again consider the simple θ -scheme and the case with piecewise linear functions ($k = 1$) and introduce notation for matrices and vectors corresponding to the Streamline Diffusion stabilization. We define $\mathbf{M}_{\text{SD}}, \mathbf{A}_{\text{SD}} \in \mathbb{R}^{N \times N}$, $\mathbf{f}_{\text{SD}} \in \mathbb{R}^N$ and $\tilde{\mathbf{M}}, \tilde{\mathbf{B}} \in \mathbb{R}^{N \times N}$, $\tilde{\mathbf{f}}_{\text{SD}} \in \mathbb{R}^N$ so that

$$\begin{aligned} \mathbf{v}^T \mathbf{M}_{\text{SD}} \mathbf{u} &= (\mathcal{G} \mathbf{u}, \mathbf{w} \cdot \nabla \mathcal{G} \mathbf{v})_{0,h}, & \tilde{\mathbf{M}} &= \mathbf{M} + \mathbf{M}_{\text{SD}} \\ \mathbf{v}^T \mathbf{A}_{\text{SD}} \mathbf{u} &:= (\mathbf{w} \cdot \nabla \mathcal{G} \mathbf{u}, \mathbf{w} \cdot \nabla \mathcal{G} \mathbf{v})_{0,h}, & \tilde{\mathbf{B}} &= \mathbf{A} + \mathbf{C} + \mathbf{A}_{\text{SD}} \\ \mathbf{v}^T \mathbf{f}_{\text{SD}} &:= (f, \mathbf{w} \cdot \nabla \mathcal{G} \mathbf{v})_{0,h}. & \tilde{\mathbf{f}} &= \mathbf{f} + \mathbf{f}_{\text{SD}} \end{aligned}$$

With the time step size $\Delta t = t_n - t_{n-1}$, where time levels within one time step are denoted as t_{n-1} , t_n and the discrete solutions are marked with corresponding superscripts, we get

$$(\tilde{\mathbf{M}} + \theta \Delta t \tilde{\mathbf{B}}) \mathbf{u}^n = \tilde{\mathbf{M}} \mathbf{u}^{n-1} - (1 - \theta) \Delta t \tilde{\mathbf{B}} \mathbf{u}^{n-1} + \Delta t \tilde{\mathbf{f}}, \quad t \in [0, T]. \quad (2.2.55)$$

In the numerical experiments in section 2.5.5 we used this method with $\theta = 1$.

Remark 2.2.5 (Time dependent velocity field). *In general $\mathbf{w}$ depends on time which in addition to a non-stationary matrix $\mathbf{C}$ also renders the Streamline Diffusion matrices $\mathbf{M}_{\text{SD}}, \mathbf{A}_{\text{SD}}$ time dependent. That case has to be treated with special care. A generalization of the θ -scheme should be used. We refer the interested reader to [GR11, Chapter 8].*

2.2.7 Conservation properties of the Nitsche-XFEM formulation

We discuss a mass conservation property of the Nitsche-XFEM discretization. On the one hand we have that global mass is conserved in Ω by the Nitsche-XFEM method. On the other hand we have a mass conservation property across Γ w.r.t. a discrete flux, the so-called *numerical flux*. For the conservation of global mass property we refer to [LR12, Remark 5]. The conservation property across the interface Γ is briefly discussed here. To this end we take the strong formulation (2.1.1a) and integrate over Ω_1 (recall $\mathbf{w} \cdot \mathbf{n} = 0$ on Γ):

$$\frac{d}{dt} \int_{\Omega_1} u_1 d\mathbf{x} = - \int_{\Gamma} \underbrace{-\alpha_1 \nabla u_1 \cdot \mathbf{n}}_{\sigma_{\mathbf{n}}} ds + \int_{\Omega_1} f d\mathbf{x}$$

which describes the balance of mass: The rate of change of the quantity u within Ω_1 is determined by the source term f in Ω_1 and the flux $\sigma_{\mathbf{n}}$ through the interface. Testing the discrete formulations (2.2.41) or (2.2.48) with $v = \beta_1^{-1}$ we get

$$\frac{d}{dt} \int_{\Omega_1} u_1 d\mathbf{x} = - \int_{\Gamma} \underbrace{-\{\alpha \nabla u \cdot \mathbf{n}\} + \lambda \frac{\bar{\alpha}}{h} \llbracket \beta u \rrbracket}_{=:\hat{\sigma}_{\mathbf{n}}} ds + \int_{\Omega_1} f d\mathbf{x}.$$

Hence, a mass balance for the discrete formulation is only achieved with respect to the numerical flux $\hat{\sigma}_{\mathbf{n}}$.

A similar derivation can be done for Ω_2 and yields the same flux reconstruction. Note that $\sigma_{\mathbf{n}}$ in the continuous formulation as well as $\hat{\sigma}_{\mathbf{n}}$ in the discrete formulation are uni-valued. A different way of deriving the conserved flux $\hat{\sigma}_{\mathbf{n}}$ is discussed in section 2.2.3.4, see especially remark 2.2.4.

Remark 2.2.6 (Minimal stabilization variant). *Note that also for the variant presented in section 2.2.3.2 this result is true (with $\lambda = 1$) as $a(\cdot, \mathcal{L}(\beta^{-1})) = 0$.*

2.3 Error analysis

In this section we present an error analysis of the Nitsche-XFEM discretization. This is done in several steps. First we consider the stationary diffusion dominated problem and derive a priori error estimates in section 2.3.1. In section 2.3.2 we consider the convection dominated regime and the discretization with a Streamline Diffusion stabilization as presented in section 2.2.5. The non-stationary problem with a stationary interface is only briefly discussed in remark 2.3.2 and remark 2.3.4

2.3.1 A priori error analysis for Nitsche-XFEM (diffusion dominates)

In this section we consider the problem 2.1.2 and its discretization with (2.2.41). A similar error analysis of the unsteady (diffusion dominated) case has been presented in [RN09]. The a priori error analysis is divided into several sections. In section 2.3.1.2, section 2.3.1.3, section 2.3.1.4 and section 2.3.1.5 we show consistency, coercivity and continuity of $B_h(\cdot, \cdot)$ (on suitable spaces, in suitable norms) and interpolation bounds for V_h^Γ . Based on those properties we apply standard ideas to proof an error bound in a natural mesh-dependent norm in section 2.3.1.6. With the help of duality arguments we further proof error bounds in the norm $\|\cdot\|_{0,\Omega}$.

Note that the analysis in this section applies to (extended) finite elements of arbitrarily order k , cf. remark 2.3.1.

2.3.1.1 Preliminaries

We recall the definition of the mean diffusion coefficient $\bar{\alpha} := \frac{1}{2}(\alpha_1 + \alpha_2)$ and assume that the ratio between α_1 and α_2 is moderate:

Assumption 2.3.1 (Moderate ratios). *The ratio between α_1 and α_2 is bounded, i.e. for $i = 1, 2$, we have $\bar{\alpha}/\alpha_i \leq c_\alpha$ with a moderate constant c_α .*

We also recall the definition of the Péclet number, the ratio between convection and diffusion w.r.t. the local mesh size h_T , $P_h^T := \frac{1}{2}|\mathbf{w}|_{\infty,T}h_T/\bar{\alpha}$. We assume the following.

Assumption 2.3.2 (Diffusion dominates). *We assume that the mesh Péclet number $P_h := \frac{1}{2}|\mathbf{w}|_\infty h_{\max}/\bar{\alpha}$ is smaller than one, $P_h \leq 1$, with $h_{\max} = \max_{T \in \mathcal{T}_h} h_T$.*

Note that in the diffusion dominated case the “convective” and “diffusive” scaling (introduced in section 2.2.5) coincide again. We thus have $\lambda \approx \mathcal{O}(1)$. We especially set $\lambda > 1$.

The constants denoted with c used in the results derived below are all independent of λ , $\bar{\alpha}$, h , $|\mathbf{w}|_\infty$, and of how the interface Γ intersects the triangulation $\mathcal{T}_h$ (i.e. of the shape regularity of T_i).

We define the space of smooth functions

$$V_{\text{reg}} := H_0^1(\Omega_{1,2}) \cap H^2(\Omega_{1,2}) \text{ and } W_{\text{reg}} := H_{0,\beta}^1(\Omega) \cap H^2(\Omega_{1,2}) \subset V_{\text{reg}}$$

and recall the bilinear form

$$B_h(u, v) = a(u, v) + N_c(u, v) + N_c(v, u) + N_s(u, v) + c(u, v), \quad u, v \in V_h^\Gamma + V_{\text{reg}}.$$

We further summarize the bilinear forms for the interface integrals to

$$N(u, v) := N_c(u, v) + N_c(v, u) + N_s(u, v), \quad u, v \in V_h^\Gamma + V_{\text{reg}}.$$

The inner products $(\cdot, \cdot)_0$ and $(\cdot, \cdot)_{1,\Omega_{1,2}}$ (with corresponding norms $\|\cdot\|_0$ and $|\cdot|_{1,\Omega_{1,2}}$) have been defined above in section 2.1.2. The inner products depend on a weighting with β , but this causes no problem since β is assumed to be of order one (see assumption 1.2.3 (moderate ratios of β)).

For the error analysis we introduce two norms. One in which we show coercivity and continuity w.r.t. the discrete space V_h^Γ and a stronger norm which also allows to show continuity for all functions in V_{reg} which specifically means that the norm is able to control normal derivatives on Γ . The norms allow for continuity and coercivity estimates with constants independent of $\bar{\alpha}$, h , λ , β , $\mathbf{w}$ and independent of how the interface cuts through the elements. In the following error analysis the corresponding parameter dependencies, which exist for the a priori error bounds, essentially only appear in the interpolation error estimates in these norms. The approximation quality of V_h^Γ for functions in V_{reg} has to be analyzed in the second, stronger, norm. The norms are

$$\|u\|_{\text{N}}^2 := \bar{\alpha}|u|_{1,\Omega_{1,2}}^2 + \bar{\alpha}\lambda \|\llbracket \beta u \rrbracket\|_{\frac{1}{2},h,\Gamma}^2, \quad u \in V_{\text{reg}} + V_h^\Gamma \quad (2.3.1)$$

$$\|u\|_{\text{N}}^2 := \|u\|_{\text{N}}^2 + \bar{\alpha}^{-1} \|\llbracket \alpha \nabla u \cdot \mathbf{n} \rrbracket\|_{-\frac{1}{2},h,\Gamma}^2, \quad u \in V_{\text{reg}} + V_h^\Gamma \quad (2.3.2)$$

with

$$\|u\|_{\frac{1}{2},h,\Gamma}^2 := \sum_{i=1,2} \sum_{T \in \mathcal{T}_h^i} h_T^{-1} \int_{\Gamma_T} u^2 ds \quad \text{and} \quad \|u\|_{-\frac{1}{2},h,\Gamma}^2 := \sum_{i=1,2} \sum_{T \in \mathcal{T}_h^i} h_T \int_{\Gamma_T} u^2 ds.$$

Note that both norms ($\|\cdot\|_{\text{N}}$ and $\|\cdot\|_{\text{N}}$) depend on β , $\bar{\alpha}$ and λ .

2.3.1.2 Consistency

As motivated in the derivation of the Nitsche method we have that for a smooth solution u of the continuous problem, problem 2.1.2, most parts of $N(u, \cdot)$ vanish.

Lemma 2.3.1 (Consistency of Nitsche bilinear form).

Let u be the solution of problem 2.1.2 with $u \in W_{\text{reg}}$. Then there holds

$$N(u, v_h) = - \sum_{i=1}^2 (\alpha_i \nabla u_i \cdot \mathbf{n}_i, \beta_i v_{h,i})_{\Gamma} \quad \forall v_h \in V_h^{\Gamma}. \quad (2.3.3)$$

Proof. As $u \in H_{0,\beta}^1(\Omega)$ we have $[\![\beta u]\!] = 0$ and hence $N_c(v_h, u) = N_s(u, v_h) = 0$. This gives

$$N(u, v_h) = N_c(u, v_h) = -(\{\!\!\{\alpha \nabla u \cdot \mathbf{n}\}\!\!\}, [\![\beta v_h]\!])_{\Gamma} = -(\alpha \nabla u \cdot \mathbf{n}, [\![\beta v_h]\!])_{\Gamma}$$

where we exploited that $\alpha \nabla u \cdot \mathbf{n}$ is well-defined and single valued on Γ . Separating the jump term into the contributions from domain 1 and 2 gives the result. $\square$

Theorem 2.3.2 (Galerkin orthogonality).

Let $u_h \in V_h^{\Gamma}$ be the solution of (2.2.41) and u be the solution of problem 2.1.2 with $u \in W_{\text{reg}}$. Then there holds

$$B_h(u_h - u, v_h) = 0 \quad \forall v_h \in V_h^{\Gamma}. \quad (2.3.4)$$

Proof. For the purpose of showing consistency we consider the smooth solution $u \in W_{\text{reg}}$ of the problem and plug it into the bilinear form $B_h(\cdot, \cdot)$ and show $B_h(u, v_h) = f(v_h)$. Note that due to $u \in H^2(\Omega_{1,2})$ there holds $f \in L^2(\Omega)$. Plugging u into $B_h(\cdot, v_h)$ gives (using lemma 2.3.1, partial integration and the boundary conditions $v_h|_{\partial\Omega} = 0$):

$$\begin{aligned} B_h(u, v_h) &= a(u, v_h) + N(u, v_h) + c(u, v_h) = a(u, v_h) + N_c(u, v_h) + c(u, v_h) \\ &= (\alpha u, v_h)_{1, \Omega_{1,2}} - \sum_{i=1}^2 (\alpha_i \nabla u_i \cdot \mathbf{n}_i, \beta_i v_{h,i})_{\Gamma} + (\mathbf{w} \cdot \nabla u, v_h)_{\Omega_{1,2}} \\ &= \underbrace{(-\text{div}(\alpha \nabla u) + \mathbf{w} \cdot \nabla u)}_{=f} v_h)_{\Omega_{1,2}} = (f, v_h)_{\Omega_{1,2}} \end{aligned}$$

By definition of u_h we further have $B_h(u_h, v_h) = f(v_h)$. Subtracting both equations gives the claim. $\square$

For an error estimate in the $L_{\beta}^2(\Omega)$ -norm we later in section 2.3.1.6 apply the *Aubin-Nitsche* trick. For this purpose we pose the homogeneous adjoint problem. For clarity we denote the solution of the adjoint problem by w . The adjoint problem reads

Problem 2.3.1.

$$-\mathbf{w} \cdot \nabla w - \operatorname{div}(\alpha \nabla w) = f \text{ in } \Omega_i, \quad i = 1, 2, \quad (2.3.6a)$$

$$\llbracket \alpha \nabla w \cdot \mathbf{n} \rrbracket_\Gamma = 0 \text{ on } \Gamma, \quad (2.3.6b)$$

$$\llbracket \beta w \rrbracket_\Gamma = 0 \text{ on } \Gamma, \quad (2.3.6c)$$

$$w = 0 \text{ on } \partial\Omega. \quad (2.3.6d)$$

There holds the following stability statement which has been formulated as an assumption in (2.1.11).

Lemma 2.3.3 (H^2 -Regularity of adjoint problem).

Let $|\mathbf{w}|_\infty < \infty$, $\Omega \subset \mathbb{R}^2$ be a convex polygon and Γ be a C^2 -smooth interface. For $f \in L^2_\beta(\Omega)$ the solution $w \in H^2_\beta(\Omega)$ of problem 2.3.1 fulfills

$$\|w\|_{2,\Omega_{1,2}} \leq c_{\text{adj}} \|f\|_0 \quad (2.3.7)$$

with a constant c_{adj} depending on $|\mathbf{w}|_\infty$.

Proof. See [CZ98]. □

Applying the adjoint of bilinear form $B_h(\cdot, \cdot)$ gives a consistent discretization of the adjoint problem. This is shown in the next lemma.

Lemma 2.3.4 (Adjoint consistency).

Let w be the solution of problem 2.3.1 with $f \in L^2_\beta(\Omega)$. Then there holds

$$B_h(v, w) = (f, v)_0 \quad \forall v \in V_{\text{reg}} + V_h^\Gamma \quad (2.3.8)$$

Proof. First note that due to $f \in L^2_\beta(\Omega)$ we have $w \in W_{\text{reg}}$. Applying partial integration for $c(\cdot, \cdot)$ and using the homogeneous boundary conditions for v_h as well as the interface condition for the velocity $\mathbf{w} \cdot \mathbf{n} = 0$ on Γ we get

$$c(v, w) = (-\mathbf{w} \cdot \nabla w, v)_{\Omega_{1,2}}$$

The bilinear forms $a(\cdot, \cdot)$ and $N(\cdot, \cdot)$ are symmetric and we thus arrive at

$$B_h(v, w) = a(v, w) + N(v, w) + c(v, w) = \underbrace{(-\operatorname{div}(\alpha \nabla w) - \mathbf{w} \cdot \nabla w, v)_{\Omega_{1,2}}}_f = (f, v)_{\Omega_{1,2}}. \quad \square$$

2.3.1.3 Stability

The crucial component in order to show stability of the Nitsche-XFEM formulation is the control on the (weighted) normal derivative. In section 2.2.2.2 we briefly addressed this point to motivate the HANSBO-weighted averaging proposed in [HH02].

Lemma 2.3.5.

If piecewise linear functions ($k = 1$) are used and the weights κ_i in the averaging $\{\!\!\{\cdot\}\!\!\}$ satisfy the estimate

$$\kappa_i^2 \leq c_\kappa \frac{|T_i|}{|T|} \quad (2.3.9)$$

for a fixed constant c_κ , then there exists a constant c_{tr} independent of α , β , h or the cut position such that

$$\bar{\alpha}^{-\frac{1}{2}} \|\{\!\!\{\alpha \nabla u_h \cdot \mathbf{n}\}\!\!\}\|_{-\frac{1}{2}, h, \Gamma} \leq \sqrt{c_{\text{tr}}} |\sqrt{\alpha} u_h|_{1, \Omega_{1,2}} \quad \text{for all } u_h \in V_h^\Gamma. \quad (2.3.10)$$

Proof. The proof is based on lemma 4 in [HH02]. There exists a fixed number c_Γ independent on h or the cut position so that $|\Gamma_T| h \leq c_\Gamma |T|$. As $\nabla u_h = \text{const}$ we can thus deduce

$$\begin{aligned} \bar{\alpha}^{-1} \|\{\!\!\{\alpha \nabla u_h \cdot \mathbf{n}\}\!\!\}\|_{-\frac{1}{2}, h, \Gamma}^2 &\leq 2\bar{\alpha}^{-1} \sum_{T \in \mathcal{T}_h^\Gamma} \sum_{i=1,2} \int_{\Gamma_T} h \alpha_i^2 \kappa_i^2 \|\nabla u_h\|_2^2 ds \\ &\leq 4 \sum_{T \in \mathcal{T}_h^\Gamma} \sum_{i=1,2} c_\Gamma |T| \alpha_i \kappa_i^2 \frac{1}{|\Gamma_T|} \int_{\Gamma_T} \|\nabla u_h\|_2^2 ds \\ &\leq 4 \sum_{T \in \mathcal{T}_h^\Gamma} \sum_{i=1,2} c_\Gamma \alpha_i \kappa_i^2 \frac{|T|}{|T_i|} \int_{T_i} \|\nabla u_h\|_2^2 d\mathbf{x} \\ &\leq 4c_\Gamma c_\kappa \sum_{i=1,2} \alpha_i |\nabla u_h|_{1, \Omega_i}^2. \end{aligned}$$

The claim follows with $c_{\text{tr}} = 4c_\Gamma c_\kappa$. $\square$

The HANSBO-weighting fulfills (2.3.9) with $c_\kappa = 1$.

Remark 2.3.1 (Higher order discretizations). *To generalize the statement in lemma 2.3.5 w.r.t. the polynomial degree k one has to either adjust the weighting κ_i or add additional stabilization terms. In [Mas12] a choice for κ_i in two dimensions is derived which allows for such a generalization. However, the proof of the result as well as the construction of κ_i is very technical. A simpler choice is $\kappa_i = 1$ if $|T_i| > \frac{1}{2}|T|$ and 0 otherwise. We expect that in that case there holds for all polynomials p and $|T_i| > \frac{1}{2}|T|$*

$$\frac{1}{|\Gamma_T|} \int_{\Gamma_T} p(\mathbf{s})^2 ds \leq \max_{\mathbf{s} \in \Gamma_T} p(\mathbf{s})^2 \leq \max_{\mathbf{x} \in T_i} p(\mathbf{x})^2 \leq c_p \frac{1}{|T_i|} \int_{T_i} p(\mathbf{x})^2 d\mathbf{x}$$

with a constant c_p only depending on the polynomial degree k . This estimate suffices to generalize the result in lemma 2.3.5. For the two-dimensional case this claim also follows directly from the analysis in [Mas12] (cf. lemma 3.5 and the prior discussion in that paper). Another possibility is a higher order version of the ghost penalty stabilization (see section 2.2.3.5) which allows to control the normal derivative at the interface independent on the cut position. We did not investigate this further.

In what follows we assume that κ_i is chosen such that (2.3.10) also holds for higher order discretizations. This implies the following

Lemma 2.3.6.

On V_h^Γ the norms $\|\cdot\|_N$ and $\|\!\!\|\cdot\|\!\!\|_N$ are equivalent, that means for every $u_h \in V_h^\Gamma$ there holds

$$\|u_h\|_N \leq \|\!\!\|u_h\|\!\!\|_N \leq c_e \|u_h\|_N. \quad (2.3.12)$$

with $c_e = \sqrt{1 + c_{tr}}$.

Proof. The left inequality is trivial. For the right inequality consider that the part of the norm $\|\!\!\|\cdot\|\!\!\|_N$ that involves normal derivatives can be bounded with the $\|\cdot\|_{1,\Omega_{1,2}}$ semi norm using lemma 2.3.5. $\square$

We derive an ellipticity result for $a_h(\cdot, \cdot)$:

Lemma 2.3.7.

For $\lambda > c_\lambda := \max\{4c_{tr}, 1\}$ there holds

$$a_h(u_h, u_h) \geq g_a \|u_h\|_N^2 \quad \text{for all } u_h \in V_h^\Gamma.$$

with $g_a = \frac{\alpha_{\min}}{2\bar{\alpha}}$.

Proof. There holds

$$\begin{aligned} N_c(u_h, u_h) &= - \int_\Gamma \{\!\!\{ \alpha \nabla \cdot \mathbf{n} \}\!\!\} \llbracket \beta u \rrbracket ds \leq \bar{\alpha}^{-\frac{1}{2}} \|\{\!\!\{ \alpha \nabla u \cdot \mathbf{n} \}\!\!\|\|_{-\frac{1}{2},h,\Gamma} \bar{\alpha}^{\frac{1}{2}} \|\llbracket \beta u \rrbracket\|_{\frac{1}{2},h,\Gamma} \\ &\leq \frac{c_{tr}}{2\gamma} |\sqrt{\alpha} u|_{1,\Omega_{1,2}}^2 + \bar{\alpha} \frac{\gamma}{2} \|\llbracket \beta u \rrbracket\|_{\frac{1}{2},h,\Gamma}^2 \end{aligned}$$

for any $\gamma > 0$ where we used lemma 2.3.5. Now setting $\gamma = 2c_{tr}$ we get

$$2N_c(u_h, u_h) \leq \frac{1}{2}a(u, u) + 2c_{tr}\bar{\alpha} \|\llbracket \beta v \rrbracket\|_{\frac{1}{2},h,\Gamma}^2 \quad (2.3.14)$$

Let $c_\lambda := \max\{4c_{tr}, 1\}$. Then for $\lambda \geq c_\lambda$ we have

$$2N_c(u_h, u_h) \leq \frac{1}{2}a(u_h, u_h) + \frac{1}{2}N_s(u_h, u_h).$$

It easily follows

$$\begin{aligned} a_h(u_h, u_h) &\geq a(u_h, u_h) - 2N_c(u_h, u_h) + N_s(u_h, u_h) \\ &\geq \frac{1}{2}(a(u_h, u_h) + N_s(u_h, u_h)) \geq \frac{\alpha_{\min}}{2\bar{\alpha}} \|u_h\|_N^2. \end{aligned}$$

$\square$

Lemma 2.3.8.

Under assumption 1.2.4 (compatible velocity) there holds

$$c(u_h, u_h) \geq 0 \quad \text{for all } u_h \in V_h^\Gamma.$$

Proof. Partial integration gives

$$\begin{aligned} (\mathbf{w} \cdot \nabla u_h, u_h)_0 &= (u_h, -\mathbf{w} \cdot \nabla u_h)_0 - (\operatorname{div}(\mathbf{w}), u_h^2)_0 \\ &\quad + (\mathbf{w} \cdot \mathbf{n}, \beta u_h^2)_{\partial\Omega} + (\mathbf{w} \cdot \mathbf{n}, \llbracket \beta u_h^2 \rrbracket)_\Gamma \end{aligned}$$

Assumption 1.2.4 (compatible velocity) includes $\operatorname{div}(\mathbf{w}) = 0$ and $\mathbf{w} \cdot \mathbf{n} = 0$ on Γ . Together with the boundary condition $u_h = 0$ for every $u_h \in V_h^\Gamma$ the claim follows. $\square$

Summing up the results from the from the previous lemmas yields

Theorem 2.3.9.

For $\lambda > c_\lambda = \max\{4c_{\text{tr}}, 1\}$ there holds

$$B_h(u_h, u_h) \geq g_B \|u_h\|_N^2, \quad \forall u_h \in V_h^\Gamma$$

with $g_B = g_A = \frac{\alpha_{\min}}{2\bar{\alpha}}$.

Proof. Combine lemma 2.3.7 and lemma 2.3.8. $\square$

This implies that the discrete problem has a unique solution.

2.3.1.4 Boundedness

Lemma 2.3.10.

For $\lambda > c_\lambda = \max\{4c_{\text{tr}}, 1\}$, there holds

$$a_h(u, v_h) \leq G_a \|u\|_N \|v_h\|_N \quad \text{for all } u \in V_{\text{reg}} + V_h^\Gamma, v_h \in V_h^\Gamma$$

for a constant $G_a = \sqrt{3}$.

Proof. Applying Cauchy-Schwarz for all bilinear forms one after another we get

$$\begin{aligned} a(u, v) &\leq \alpha_{\max} |u|_{1, \Omega_{1,2}} |v_h|_{1, \Omega_{1,2}} \\ N_c(u, v_h) &\leq \bar{\alpha}^{-\frac{1}{2}} \|\llbracket \alpha \nabla u \cdot \mathbf{n} \rrbracket\|_{-\frac{1}{2}, h, \Gamma} \bar{\alpha}^{\frac{1}{2}} \|\llbracket \beta v_h \rrbracket\|_{\frac{1}{2}, h, \Gamma} \\ N_c(v_h, u) &\leq \bar{\alpha}^{-\frac{1}{2}} \|\llbracket \alpha \nabla v_h \cdot \mathbf{n} \rrbracket\|_{-\frac{1}{2}, h, \Gamma} \bar{\alpha}^{\frac{1}{2}} \|\llbracket \beta u \rrbracket\|_{\frac{1}{2}, h, \Gamma} \\ &\leq \sqrt{c_{\text{tr}}} \sqrt{\alpha} |v_h|_{1, \Omega_{1,2}} \bar{\alpha}^{\frac{1}{2}} \|\llbracket \beta u \rrbracket\|_{\frac{1}{2}, h, \Gamma} \\ N_s(u, v_h) &\leq \lambda^{\frac{1}{2}} \bar{\alpha}^{\frac{1}{2}} \|\llbracket \beta u \rrbracket\|_{\frac{1}{2}, h, \Gamma} \lambda^{\frac{1}{2}} \bar{\alpha}^{\frac{1}{2}} \|\llbracket \beta v_h \rrbracket\|_{\frac{1}{2}, h, \Gamma} \end{aligned}$$

With $\lambda > \max\{4c_{\text{tr}}, 1\}$ we get

$$a_h(u, v_h) \leq \sqrt{3} \|u\|_N \|v_h\|_N$$

which implies the claim. $\square$

Lemma 2.3.11.

There holds

$$c(u, v) \leq \frac{|\mathbf{w}|_\infty}{\sqrt{\alpha}} \|u\|_0 \|v\|_N \quad \text{for all } u, v \in V_{\text{reg}} + V_h^\Gamma.$$

Proof. As in lemma 2.3.8 we apply partial integration and make use of assumption 1.2.4 (compatible velocity) and boundary conditions. Then applying Cauchy-Schwarz inequality gives the result. $\square$

Theorem 2.3.12.

There holds

$$B_h(u, v_h) \leq G_B \|u\|_N \|v_h\|_N + \frac{|\mathbf{w}|_\infty}{\sqrt{\alpha}} \|u\|_0 \|v_h\|_N \quad \forall u \in V_{\text{reg}} + V_h^\Gamma, v_h \in V_h^\Gamma \quad (2.3.16)$$

with $G_B = \sqrt{3}$.

Proof. Combine lemma 2.3.10 and lemma 2.3.11. $\square$

2.3.1.5 Interpolation error

In the analysis of the Nitsche-XFEM method an interpolation operator $I_h^\Gamma : V_{\text{reg}} \rightarrow V_h^\Gamma$ plays an important role. We recall the extension operator $\mathcal{E}_i$ and the restriction operator $\mathcal{R}_i$ introduced in section 2.2.1.1. Let $I_h : H^2(\Omega) \cap H_0^1(\Omega) \rightarrow V_h$ be the standard nodal interpolation operator corresponding to the space V_h of continuous finite elements of degree k . The XFEM interpolation operator is given by

$$I_h^\Gamma = \mathcal{R}_1 I_h \mathcal{E}_1 \mathcal{R}_1 + \mathcal{R}_2 I_h \mathcal{E}_2 \mathcal{R}_2. \quad (2.3.17)$$

For the interpolation operator I_h^Γ optimal (local) interpolation error bounds can easily be derived

Lemma 2.3.13.

Let $u \in V_{\text{reg}} \cap H^m(\Omega_{1,2})$, $m \geq 2$ and I_h^Γ be the interpolation operator into V_h^Γ the XFEM finite element space of (continuous) piecewise polynomials of degree $k \geq 1$ as in (2.3.17). With $e_I := u - I_h^\Gamma u$, $\tilde{u}_i = \mathcal{E}_i \mathcal{R}_i u$, $l := \min\{k, m - 1\}$ and $i = 1, 2$ there hold the following local interpolation error bounds for $T \in \mathcal{T}_h$.

$$\|e_I\|_{0,T_i}^2 \leq ch_T^{2l+2} \|\tilde{u}_i\|_{l+1,T}^2 \quad (2.3.18a) \quad \|\mathcal{R}_i e_I\|_{\frac{1}{2},h,\Gamma_T}^2 \leq ch_T^{2l} \|\tilde{u}_i\|_{l+1,T}^2 \quad (2.3.18c)$$

$$|e_I|_{1,T_i}^2 \leq ch_T^{2l} \|\tilde{u}_i\|_{l+1,T}^2 \quad (2.3.18b) \quad \|\nabla \mathcal{R}_i e_I \cdot \mathbf{n}\|_{-\frac{1}{2},h,\Gamma_T}^2 \leq ch_T^{2l} \|\tilde{u}_i\|_{l+1,T}^2 \quad (2.3.18d)$$

Proof. The results in (2.3.18a) and (2.3.18b) are known in the literature, see for instance [HH02, Reu08]. The results in (2.3.18c), (2.3.18d), are derived in [HH02]. The essential ingredient is the following result:

$$\|w\|_{L^2(\Gamma_T)}^2 \leq c(h_T^{-1} \|w\|_{L^2(T)}^2 + h_T |w|_{1,T}^2) \quad \text{for all } w \in H^1(T),$$

which holds for all $T \in \mathcal{T}_h^\Gamma$ and with a constant c that is independent of the shape regularity of T_i , cf. [HH02, GR11]. We give the proof of (2.3.18c) and (2.3.18d):

$$\begin{aligned} \|\mathcal{R}_i e_I\|_{\frac{1}{2},h,\Gamma_T}^2 &= h_T^{-1} \|\mathcal{E}_i R_i e_I\|_{L^2(\Gamma_T)}^2 \leq c \left(h_T^{-2} \|\mathcal{E}_i R_i e_I\|_{L^2(T)}^2 + |\mathcal{E}_i R_i e_I|_{1,T}^2 \right) \\ &\leq ch_T^{2l} \|\tilde{u}_i\|_{2,T}^2 \Rightarrow (2.3.18c), \end{aligned}$$

$$\begin{aligned} \|\nabla \mathcal{R}_i e_I \cdot \mathbf{n}\|_{-\frac{1}{2},h,\Gamma_T}^2 &\leq h_T \|\nabla \mathcal{E}_i R_i e_I\|_{L^2(\Gamma_T)}^2 \leq c \left(\|\nabla \mathcal{E}_i R_i e_I\|_{L^2(T)}^2 + h_T^2 |\nabla \mathcal{E}_i R_i e_I|_{1,T}^2 \right) \\ &\leq ch_T^{2l} \|\tilde{u}_i\|_{2,T}^2 \Rightarrow (2.3.18d). \end{aligned}$$

□

Lemma 2.3.14.

Let $u \in V_{\text{reg}} \cap H^m(\Omega_{1,2})$ and I_h^Γ as in lemma 2.3.13. Assume the family of triangulations is quasi-uniform so that $h_T \leq ch$. With $e_I := u - I_h^\Gamma u$ and $l := \min\{k, m - 1\}$ there holds

$$\|e_I\|_{0,\Omega_{1,2}} \leq ch^{l+1} \|u\|_{l+1,\Omega_{1,2}} \quad (2.3.19a) \quad \|\mathcal{R}_i e_I\|_{\frac{1}{2},h,\Gamma} \leq ch^l \|u\|_{l+1,\Omega_i} \quad (2.3.19c)$$

$$|e_I|_{1,\Omega_{1,2}} \leq ch^l \|u\|_{l+1,\Omega_{1,2}} \quad (2.3.19b) \quad \|\nabla \mathcal{R}_i e_I \cdot \mathbf{n}\|_{-\frac{1}{2},h,\Gamma} \leq ch^l \|u\|_{l+1,\Omega_i} \quad (2.3.19d)$$

Proof. Using quasi-uniformity for the estimates in lemma 2.3.13 and in addition making use of the continuity of the extension operator $\|\mathcal{E}_i \mathcal{R}_i u\|_{l+1,\Omega} \leq c \|u\|_{l+1,\Omega_i}$ directly gives the estimates. □

For the norms $\|\cdot\|_N$ and $\|\cdot\|_N$ we can directly conclude interpolation bounds using the above estimates.

Lemma 2.3.15.

For $u \in V_{\text{reg}} \cap H^m(\Omega_{1,2})$, $m \geq 2$ and $l := \min\{k, m-1\}$ the following interpolation error bound holds on a quasi-uniform family of triangulations with mesh size h :

$$\|u - I_h^\Gamma u\|_N \leq \|u - I_h^\Gamma u\|_N \leq c\sqrt{\alpha}\lambda h^l \|u\|_{2,\Omega_{1,2}} \quad (2.3.20)$$

2.3.1.6 A priori error estimates**Lemma 2.3.16** (Modified Cea's lemma).

Let $u_h \in V_h^\Gamma$ be the solution of (2.2.41) with $\lambda > c_\lambda$ and u be the solution of problem 2.1.2 with $u \in W_{\text{reg}}$. There holds, with g_B , G_B as in theorem 2.3.9, theorem 2.3.12

$$\|u - u_h\|_N \leq \inf_{v_h \in V_h^\Gamma} \left\{ \left(1 + \frac{G_B}{g_B}\right) \|u - v_h\|_N + \frac{1}{g_B} P_h \frac{\sqrt{\alpha}}{h_{\max}} \|u - v_h\|_0 \right\} \quad (2.3.21a)$$

and

$$\|u - u_h\|_N \leq \inf_{v_h \in V_h^\Gamma} \left\{ \left(1 + \frac{G_B}{g_B} c_e\right) \|u - v_h\|_N + \frac{c_e}{g_B} P_h \frac{\sqrt{\alpha}}{h_{\max}} \|u - v_h\|_0 \right\}. \quad (2.3.21b)$$

Proof. With the triangle inequality we split the error into two parts. We introduce a new discrete function v_h which is arbitrary at this point but will be the best approximation to u in V_h^Γ later. We define $e_a := u - v_h$ what is going to represent the approximation error and $e_d := v_h - u_h \in V_h^\Gamma$ what is denoted as the discrete error:

$$\|u - u_h\|_N^2 \leq \|e_d\|_N^2 + \|e_a\|_N^2, \quad \|u - u_h\|_N^2 \leq \|e_d\|_N^2 + \|e_a\|_N^2,$$

For the discrete error we can apply (in order) lemma 2.3.6, theorem 2.3.9, lemma 2.3.1 and theorem 2.3.12 to bound the discrete error by the approximation error:

$$\begin{aligned} c_e^{-1} \|e_d\|_N \|e_d\|_N &\leq \|v_h - u_h\|_N^2 \leq \frac{1}{g_B} B_h(v_h - u_h, v_h - u_h) \\ &= \frac{1}{g_B} B_h(v_h - u, v_h - u_h) \\ &\leq \frac{G_B}{g_B} \|e_a\|_N \|e_d\|_N + \frac{1}{g_B} \frac{|\mathbf{w}|_\infty}{\sqrt{\alpha}} \|e_a\|_0 \|e_d\|_N \\ \Rightarrow \|e_d\|_N &\leq \frac{G_B}{g_B} \|e_a\|_N + \frac{1}{g_B} \frac{|\mathbf{w}|_\infty}{\sqrt{\alpha}} \|e_a\|_0, \\ \|e_d\|_N &\leq \frac{G_B}{g_B} c_e \|e_a\|_N + \frac{c_e}{g_B} \frac{|\mathbf{w}|_\infty}{\sqrt{\alpha}} \|e_a\|_0 \end{aligned}$$

Putting all together concludes the proof. $\square$

Lemma 2.3.17 (Error bound in $\|\cdot\|_N$ -norm).

Let $u_h \in V_h^\Gamma$ be the solution of (2.2.41) and u be the solution of problem 2.1.2 with $u \in W_{\text{reg}} \cap H^m(\Omega_{1,2})$. Define $l := \min\{k, m-1\}$ and assume $\mathcal{T}_h$ is quasi-uniform with mesh size h . Then there holds

$$\|u - u_h\|_N \leq \|u - u_h\|_N \leq c(1 + P_h)\sqrt{\bar{\alpha}\lambda}h^l \|u\|_{2,\Omega_{1,2}}. \quad (2.3.23)$$

Proof. Combine lemma 2.3.16 and the approximation results in lemma 2.3.15. $\square$

With $P_h \leq 1$ this implies for sufficiently smooth functions u

$$|u - u_h|_{1,\Omega_{1,2}} \leq c\sqrt{\lambda}h^k \|u\|_{2,\Omega_{1,2}}, \quad (2.3.24a)$$

$$\|[\beta u_h]\|_\Gamma \leq ch^{k+\frac{1}{2}} \|u\|_{2,\Omega_{1,2}}, \quad (2.3.24b)$$

$$\|\llbracket \alpha \nabla u \cdot \mathbf{n} \rrbracket\|_\Gamma \leq c\bar{\alpha}\sqrt{\lambda}h^{k-\frac{1}{2}} \|u\|_{2,\Omega_{1,2}}. \quad (2.3.24c)$$

Duality arguments.
Lemma 2.3.18 (Error bound in $\|\cdot\|_0$).

Let $u_h \in V_h^\Gamma$ be the solution of (2.2.41) and u be the solution of problem 2.1.2 with $u \in V_{\text{reg}} \cap H^m(\Omega_{1,2})$. Assuming a quasi-uniform mesh with mesh size h and $P_h \leq 1$ there holds (with $l := \min\{k, m-1\}$)

$$\|u - u_h\|_0 \leq c\sqrt{\bar{\alpha}\lambda}h \|u - u_h\|_N \quad (2.3.25)$$

$$\leq c\bar{\alpha}\lambda h^{l+1} \|u\|_{2,\Omega_{1,2}} \quad (2.3.26)$$

Proof. We consider the adjoint problem, problem 2.3.1, with data $f = u - u_h \in L^2(\Omega)$ and denote the corresponding solution as w . We exploit adjoint consistency (lemma 2.3.4), Galerkin orthogonality of the primal problem (lemma 2.3.1) and continuity (theorem 2.3.12) to get (for any $w_h \in V_h^\Gamma$)

$$\|u - u_h\|_0^2 = (u - u_h, u - u_h)_0 = B_h(u - u_h, w) \quad (2.3.27a)$$

$$= B_h(u - u_h, w - w_h) \leq K \|u - u_h\|_N \|w - w_h\|_* \quad (2.3.27b)$$

where in the last step we used

$$B_h(u, v) \leq K \|u\|_N \|v\|_* \quad \forall u, v \in V_{\text{reg}} + V_h^\Gamma$$

which is a generalization of theorem 2.3.12 with

$$\|v\|_* := \|v\|_N + \frac{|\mathbf{w}|_\infty}{\sqrt{\bar{\alpha}}} \|v\|_0.$$

Now choose $w_h = I_h^\Gamma w$ and apply the previously used approximation results (with $P_h \leq 1$) to get

$$\|u - u_h\|_0^2 \leq c\sqrt{\bar{\alpha}\lambda}hK \|u - u_h\|_N \|w\|_{2,\Omega_{1,2}}.$$

Applying the stability estimate of the adjoint problem (see lemma 2.3.3) to bound

$$\|w\|_{2,\Omega_{1,2}} \leq c_{\text{adj}} \|f\|_0 = c_{\text{adj}} \|u - u_h\|_0$$

finally yields

$$\|u - u_h\|_0 \leq c c_{\text{adj}} \sqrt{\bar{\alpha}\lambda}hK \|u - u_h\|_N.$$

Now plugging in the result in lemma 2.3.17 concludes the proof. $\square$

Remark 2.3.2 (Non-stationary problem). *For the semi-discretization in (2.2.51) optimal order error bounds are derived in [RN09] for polynomial degree $k = 1$. In the analysis in that paper it is also assumed that the transport problem is diffusion-dominated. Further, in that paper the full discretization (2.2.52) with $\theta = 1$ is analyzed and error bounds are derived.*

2.3.2 A priori error analysis for Nitsche-XFEM with Streamline Diffusion stabilization (convection dominates)

In this section we present an error analysis of the Nitsche-XFEM method with Streamline Diffusion stabilization. The error analysis is based on the one in [LR12].

We are particularly interested in the convection-dominated case, and therefore allow $\bar{\alpha} = \frac{1}{2}(\alpha_1 + \alpha_2) \downarrow 0$. To emphasize that α is small we introduced the notation for the diffusion coefficient with ε such that $\varepsilon_i = \alpha_i$ and $\bar{\varepsilon} = \bar{\alpha}$.

We recall that in the convection dominated case the Nitsche stabilization parameter λ can be chosen within a given range bounded by the limit cases that we called “diffusive” and “convective” scaling. Further note that λ can be defined element-local which allows it to differ on different elements $T \in \mathcal{T}_h$:

$$c_\lambda = \lambda^d \leq \lambda_T \leq \lambda_T^c := c_\lambda \max\{P_h^T, 1\}$$

For ease of presentation we make the following assumption which can always be achieved by rescaling.

Assumption 2.3.3 ($|\mathbf{w}|_\infty \leq \mathcal{O}(1)$). *We assume that $|\mathbf{w}|_\infty \leq 1$, s.t. the amount of convection domination only depends on the size of the diffusion parameters ε_i , $i = 1, 2$.*

We restrict to piecewise linear functions in this section. For $k = 1$ we have $\text{div}(\nabla u_h) = 0$ for all $u_h \in V_h^\Gamma$. This is exploited in lemma 2.3.21. We comment on higher order extensions in remark 2.3.3.

2.3.2.1 Preliminaries

We will need several norms related to the Nitsche stabilization (see previous section) and the Streamline Diffusion stabilization. For the Streamline Diffusion stabilization we recall the inner product

$$(u, v)_{0,h} = \sum_{T \in \mathcal{T}_h} \gamma_T (u, v)_{0,T}$$

with corresponding norm denoted by $\|\cdot\|_{0,h}$.

In the analysis we consider the solution of the following problem.

Problem 2.3.2.

$$\xi u + \mathbf{w} \cdot \nabla u - \operatorname{div}(\alpha \nabla u) = f \quad \text{in } \Omega_i, \quad i = 1, 2, \quad (2.3.28a)$$

$$[\![\alpha \nabla u \cdot \mathbf{n}]\!]_{\Gamma} = 0 \quad \text{on } \Gamma, \quad (2.3.28b)$$

$$[\![\beta u]\!]_{\Gamma} = 0 \quad \text{on } \Gamma, \quad (2.3.28c)$$

$$u = 0 \quad \text{on } \partial\Omega. \quad (2.3.28d)$$

Compared to the transport problem considered above (see problem 2.1.2) we introduced an additional zero order term ξu , with a given constant $\xi \geq 0$. This is standard in the analysis of convection-dominated problems (cf. [RST08]), since only if this zero order term is present ($\xi > 0$) one can derive uniform error bounds in the $L^2_{\beta}(\Omega)$ -norm.

To obtain estimates that are uniform with respect to the parameter ξ , we have to generalize the choice of the stabilization parameter γ_T and choose

$$\gamma_T = \min \left\{ \frac{2h_T}{|\mathbf{w}|_{\infty,T}}, \frac{h_T^2}{\bar{\varepsilon}}, \frac{1}{\xi} \right\}. \quad (2.3.29)$$

This parameter choice is essentially the same as in [RST08] and allows for the following estimates.

Lemma 2.3.19.

With γ_T as in (2.3.29) the following estimates can be derived:

$$\gamma_T \xi \leq 1, \quad (2.3.30a) \quad \gamma_T \bar{\varepsilon}^2 \leq \bar{\varepsilon} h_T^2, \quad (2.3.30c)$$

$$\gamma_T |\mathbf{w}|_{\infty,T} \leq 2h_T, \quad (2.3.30b) \quad \gamma_T^{-1} h_T^2 \leq \xi h_T^2 + \frac{1}{2} |\mathbf{w}|_{\infty,T} h_T + \bar{\varepsilon}. \quad (2.3.30d)$$

We derive an error bound for the discretization error cf. theorem 2.3.28 below. To this end we investigate the bilinear form

$$C_h(u, v) := a_h(u, v) + c(u, v) + b(u, v) + d(u, v), \quad u, v \in V_{\text{reg}} + V_h^{\Gamma} \quad (2.3.31)$$

with $d(u, v) := (-\operatorname{div}(\varepsilon \nabla u) + \xi u + \mathbf{w} \cdot \nabla u, \mathbf{w} \cdot \nabla v)_{0,h}$

and $b(u, v) := \xi(u, v)_0$

and the linear form

$$f(v) := (f, v)_0 + (f, \mathbf{w} \cdot \nabla v)_{0,h}, \quad u, v \in V_{\text{reg}} + V_h^\Gamma. \quad (2.3.32)$$

The discrete problem is

$$\text{Find } u_h \in V_h^\Gamma, \text{ so that } C_h(u_h, v_h) = f(v_h) \text{ for all } v_h \in V_h^\Gamma. \quad (2.3.33)$$

As we will see below, we can derive an ellipticity and continuity result for the bilinear form $C_h(\cdot, \cdot)$ with respect to suitable norms. As expected, these norms involve terms that result from the Nitsche stabilization and from the Streamline Diffusion stabilization. To simplify the presentation we split the bilinear form in two parts (corresponding to Nitsche and Streamline Diffusion stabilization) and first consider these two parts separately where we can recycle results from section 2.3.1. Afterwards the results for these two parts can easily be glued together.

We use the splitting

$$\begin{aligned} C_h(u, v) &:= a_h^N(u, v) + a_h^{\text{SD}}(u, v) \\ a_h^N(u, v) &:= \frac{1}{2}a(u, v) + N(u, v) \\ a_h^{\text{SD}}(u, v) &:= \frac{1}{2}a(u, v) + c(u, v) + b(u, v) + d(u, v). \end{aligned}$$

In the analysis of the method the following norms are used:

$$\|v\|_N^2 := \frac{1}{2}\bar{\varepsilon}|v|_{1,\Omega_{1,2}}^2 + \lambda\bar{\varepsilon}\|\llbracket \beta v \rrbracket\|_{\frac{1}{2},h,\Gamma}^2, \quad (2.3.34a)$$

$$\|v\|_{\text{SD}}^2 := \frac{1}{2}\bar{\varepsilon}|v|_{1,\Omega_{1,2}}^2 + \xi\|v\|_0^2 + \|\mathbf{w} \cdot \nabla v\|_{0,h}^2, \quad (2.3.34b)$$

$$\|v\|_C^2 := \|v\|_{\text{SD}}^2 + \|v\|_N^2. \quad (2.3.34c)$$

Further we use the stronger norms

$$\|v\|_N^2 := \|v\|_N^2 + \bar{\varepsilon}^{-1}\|\llbracket \varepsilon \nabla u \cdot \mathbf{n} \rrbracket\|_{-\frac{1}{2},h,\Gamma}^2, \quad (2.3.35a)$$

$$\|v\|_{\text{SD}}^2 := \|v\|_{\text{SD}}^2 + \sum_{T \in \mathcal{T}_h} \gamma_T^{-1}\|v\|_{0,T}^2 + \bar{\varepsilon}^2\|\Delta v\|_{0,h}^2, \quad (2.3.35b)$$

$$\|v\|_C^2 := \|v\|_{\text{SD}}^2 + \|v\|_N^2. \quad (2.3.35c)$$

Note that the two terms $\|\mathbf{w} \cdot \nabla v\|_{0,h}^2$ and $\lambda\bar{\varepsilon}\|\llbracket \beta v \rrbracket\|_{\frac{1}{2},h,\Gamma}^2$ originate from the stabilization terms in the Streamline Diffusion and the Nitsche method, respectively.

Note also that the scaling of the (β -weighted) H^1 -semi-norm is different from the one in the last section by a factor $\frac{1}{2}$.

The constants used in the results derived below are all independent of $\lambda, \xi, \bar{\varepsilon}, h, |\mathbf{w}|_\infty$, and of how the interface Γ intersects the triangulation $\mathcal{T}_h$ (i.e. of the shape regularity of T_i).

2.3.2.2 Consistency

The discrete formulation (2.3.33) is consistent.

Lemma 2.3.20.

Let $u_h \in V_h^\Gamma$ be the solution of (2.3.33) and $u \in W_{\text{reg}}$ be the solution of problem 2.3.2, then there holds

$$C_h(u - u_h, v_h) = 0 \quad \text{for all } v_h \in V_h^\Gamma.$$

Proof. Note that $C_h(\cdot, \cdot) = B_h(\cdot, \cdot) + b(\cdot, \cdot) + d(\cdot, \cdot)$ and with the consistency result from the previous section we have $B_h(u, v_h) + b(u, v_h) = (f, v_h)_0$. Further by construction we have $-\text{div}(\varepsilon \nabla u) + \xi u - \mathbf{w} \cdot \nabla u = f$ and thus $d(u, v_h) = (f, \mathbf{w} \cdot \nabla v_h)_{0,h}$. Together the claim holds. $\square$

2.3.2.3 Stability

Recalling lemma 2.3.7 we already have

$$a_h^N(u_h, u_h) \geq \frac{1}{2} g_a \|u_h\|_N^2 \quad \text{for all } u_h \in V_h^\Gamma.$$

We now show a similar statement for the Streamline Diffusion part $a_h^{\text{SD}}(\cdot, \cdot)$ of the bilinear form.

Lemma 2.3.21.

There holds

$$a_h^{\text{SD}}(u_h, u_h) \geq c_\alpha \|u_h\|_{\text{SD}}^2 \quad \text{for all } u_h \in V_h^\Gamma,$$

with $c_\alpha = \frac{\varepsilon_{\min}}{2\bar{\varepsilon}} = g_a$.

Proof. Using lemma 2.3.8 we have $c(u_h, u_h) \geq 0$. Furthermore, using (2.3.30a) we get

$$\begin{aligned} \xi(u_h, \mathbf{w} \cdot \nabla u_h)_{0,h} &= \xi \sum_{T \in \mathcal{T}_h} \gamma_T (u_h, \mathbf{w} \cdot \nabla u_h)_{0,T_{1,2}} \\ &\leq \frac{1}{2} \sum_{T \in \mathcal{T}_h} \xi^2 \gamma_T \|u_h\|_{0,T}^2 + \gamma_T \|\mathbf{w} \cdot \nabla u_h\|_{0,T_{1,2}}^2 \\ &\leq \frac{1}{2} \xi \|u_h\|_0^2 + \frac{1}{2} \|\mathbf{w} \cdot \nabla u_h\|_{0,h}^2. \end{aligned}$$

Hence,

$$\begin{aligned} a_h^{\text{SD}}(u_h, u_h) &\geq \frac{1}{2} \varepsilon_{\min} \|u_h\|_{1,\Omega_{1,2}}^2 + \xi \|u_h\|_0^2 + \|\mathbf{w} \cdot \nabla u_h\|_{0,h}^2 + \xi(u_h, \mathbf{w} \cdot \nabla u_h)_{0,h} \\ &\geq \frac{\varepsilon_{\min}}{\bar{\varepsilon}} \frac{\bar{\varepsilon}}{2} \|u_h\|_{1,\Omega_{1,2}}^2 + \frac{1}{2} \xi \|u_h\|_0^2 + \frac{1}{2} \|\mathbf{w} \cdot \nabla u_h\|_{0,h}^2. \end{aligned}$$

□

The result in the previous lemma only applies for piecewise linear functions ($k = 1$), cf. remark 2.3.3.

Lemma 2.3.22.

For $\lambda > 2c_\lambda$ there holds

$$C_h(u_h, u_h) \geq g_C \|u_h\|_C^2 \quad \text{for all } u_h \in V_h^\Gamma.$$

with $g_C = g_a$

Proof. Note that the bilinear form $a_h^N(\cdot, \cdot)$ and the norm $\|\cdot\|_N$ is different from the one used in lemma 2.3.7. To retain the result we adapt the Nitsche stabilization parameter λ . To this end we restricted $\lambda > 2c_\lambda$. Now in the proof of lemma 2.3.7 we can choose $\gamma = 4c_{\text{tr}}$ and get the desired result with the same ellipticity constant g_a . Together with lemma 2.3.21 we get

$$C_h(u_h, u_h) = a_h^{\text{SD}}(u_h, u_h) + a_h^N(u_h, u_h) \geq g_a \|u_h\|_{\text{SD}}^2 + g_a \|u_h\|_N^2 = g_a \|u_h\|_C^2$$

□

2.3.2.4 Boundedness**Lemma 2.3.23.**

For every $u \in V_{\text{reg}} + V_h^\Gamma$, $v_h \in V_h^\Gamma$ there holds

$$a_h^{\text{SD}}(u, v_h) \leq 2 \|u\|_{\text{SD}} \|v_h\|_{\text{SD}}.$$

Proof.

$$\frac{1}{2} a(u, v_h) \leq \frac{1}{2} \overbrace{\varepsilon_{\max}}^{\leq \bar{\varepsilon}} |u|_{1, \Omega_{1,2}} |v_h|_{1, \Omega_{1,2}} \quad (2.3.36a)$$

$$b(u, v_h) \leq \xi \|u\|_0 \|v_h\|_0 \quad (2.3.36b)$$

$$\begin{aligned} d(u, v_h) &\leq \left(\sum_{T \in \mathcal{T}_h} \underbrace{\gamma_T(\xi^2 \|u\|_{0,T}^2 + \bar{\varepsilon}^2 \|\Delta u\|_{0,T}^2 + \|\mathbf{w} \cdot \nabla u\|_{0,T}^2)}_{\leq \xi} \right)^{\frac{1}{2}} \\ &\quad \cdot \|\mathbf{w} \cdot \nabla v_h\|_{0,h} \end{aligned} \quad (2.3.36c)$$

For $c(u, v)$ we apply partial integration

$$(\mathbf{w} \cdot \nabla u, v_h)_0 = -(u, \mathbf{w} \cdot \nabla v_h)_0 \leq \left(\sum_{T \in \mathcal{T}_h} \gamma_T^{-1} \|u\|_{0,T}^2 \right)^{\frac{1}{2}} \|\mathbf{w} \cdot \nabla v_h\|_{0,h}$$

□

Recalling lemma 2.3.10 we further have for $u \in V_{\text{reg}} + V_h^\Gamma$, $v \in V_h^\Gamma$

$$a_h^N(u, v_h) \leq G_a \|u\|_N \|v\|_N.$$

Lemma 2.3.24.

There exists a constant c such that for every $u \in V_{\text{reg}} + V_h^\Gamma$, $v \in V_h^\Gamma$

$$C_h(u, v_h) \leq G_C \|u\|_C \|v_h\|_C,$$

with $G_C = 4$.

Proof. Combine lemma 2.3.10 and lemma 2.3.23. $\square$

2.3.2.5 Interpolation error

Similar to lemma 2.3.15 we have for $u \in V_{\text{reg}}$ the following interpolation error bound

$$\|u - I_h^\Gamma u\|_N \leq \|u - I_h^\Gamma u\|_N \leq c\sqrt{\varepsilon\lambda}h\|u\|_{2,\Omega_{1,2}} \quad (2.3.37)$$

For the SD norm we get

Lemma 2.3.25.

Assume $h \leq ch_T$ (quasi-uniformity of the family of triangulations), then there holds the following interpolation error bound for $u \in V_{\text{reg}}$

$$\|u - I_h^\Gamma u\|_{\text{SD}} \leq \|u - I_h^\Gamma u\|_{\text{SD}} \leq (\sqrt{\varepsilon}h + \sqrt{\xi}h^2 + \sqrt{|\mathbf{w}|_\infty}h^{\frac{3}{2}})\|u\|_{2,\Omega_{1,2}}$$

Proof. We split the norm into its element contributions and apply the element local interpolation estimates in (2.3.18a)-(2.3.18d) from the last section.

$$\begin{aligned} \|e_a\|_{\text{SD}}^2 &= \bar{\varepsilon}|e_a|_{1,\Omega_{1,2}}^2 + \xi\|e_a\|_0^2 + \sum_{T \in \mathcal{T}_h} \gamma_T^{-1}\|e_a\|_{0,T}^2 + \|\mathbf{w} \cdot \nabla e_a\|_{0,h}^2 + \bar{\varepsilon}^2\|\Delta e_a\|_{0,h}^2 \\ &\leq \sum_{T \in \mathcal{T}_h} \left(\bar{\varepsilon}|e_a|_{1,T_{1,2}}^2 + \xi\|e_a\|_{0,T}^2 + \gamma_T^{-1}\|e_a\|_{0,T}^2 + \gamma_T|\mathbf{w}|_{\infty,T}^2\|e_a\|_{1,T_{1,2}}^2 + \gamma_T\bar{\varepsilon}^2\|\Delta u\|_{0,T_{1,2}}^2 \right) \\ &\leq \sum_{T \in \mathcal{T}_h} \left(\bar{\varepsilon}h_T^2 + \xi h_T^4 + (\gamma_T^{-1}h_T^2)h_T^2 + \gamma_T|\mathbf{w}|_{\infty,T}^2h_T^2 + \bar{\varepsilon}^2\gamma_T \right) \cdot \|u\|_{2,T_{1,2}}^2 \end{aligned}$$

Now, recalling $\gamma_T|\mathbf{w}|_{\infty,T} \leq 2h_T$, $\bar{\varepsilon}^2\gamma_T \leq \bar{\varepsilon}h_T^2$ and $\gamma_T^{-1}h_T^2 \leq \xi h_T^2 + \frac{1}{2}|\mathbf{w}|_{\infty,T}h_T + \bar{\varepsilon}$ from (2.3.30) concludes the proof. $\square$

We conclude

Lemma 2.3.26.

Assume $h \leq ch_T$ (quasi-uniformity of the family of triangulations), then there holds the following interpolation error bound for $u \in V_{\text{reg}}$

$$\|u - I_h^\Gamma u\|_C \leq (\sqrt{\varepsilon}\lambda h + \sqrt{\xi}h^2 + \sqrt{|\mathbf{w}|_\infty}h^{\frac{3}{2}})\|u\|_{2,\Omega_{1,2}}$$

Proof. Combine the recent lemma, (2.3.37) and recall $\lambda > 1$. $\square$

According to the last two estimates we reconsider the question on how to choose λ . While in the diffusion dominated case the whole error bound (2.3.23) increased for increasing λ we now have also other terms which can absorb the dependency on λ up to a certain limit. We notice that as long as there holds

$$\bar{\varepsilon}\lambda_T \leq \max\{\xi h_T^2, |\mathbf{w}|_{\infty,T}h_T\}$$

the dependency on λ can be absorbed by the other parameters. This justifies the extreme cases “diffusive” and “convective” scaling presented in (2.2.50). Note that ξ has only been introduced for theoretical purposes.

2.3.2.6 A priori error estimates

Lemma 2.3.27.

Let $u_h \in V_h^\Gamma$ be the solution of (2.3.33) with $\lambda > 2c_\lambda$ and $u \in W_{\text{reg}}$ be the solution of problem 2.3.2, then there holds

$$\|u - u_h\|_C \leq \left(1 + \frac{G_C}{g_C}\right) \inf_{v_h \in V_h^\Gamma} \|u - v_h\|_C \quad (2.3.38)$$

Proof. Triangle inequality gives

$$\|u - u_h\|_C \leq \underbrace{\|u - v_h\|_C}_{=:e_a} + \underbrace{\|v_h - u_h\|_C}_{=:e_d} \leq \|e_a\|_C + \|e_d\|_C$$

for any $v_h \in V_h^\Gamma$. Then we apply (in order) lemma 2.3.22, lemma 2.3.20, and lemma 2.3.24 to get

$$\begin{aligned} \|e_d\|_C^2 &\leq \frac{1}{g_C} C_h(u_h - v_h, u_h - v_h) \\ &= \frac{1}{g_C} C_h(u - v_h, u_h - v_h) \\ &\leq \frac{G_C}{g_C} \|e_a\|_C \|e_d\|_C \end{aligned}$$

$\square$

Theorem 2.3.28.

Let $u_h \in V_h^\Gamma$ be the solution of (2.3.33) with $\lambda > 2c_\lambda$ and $u \in W_{\text{reg}}$ be the solution of problem 2.3.2. Furthermore assume $h \leq ch_T$ (quasi-uniformity of the family of triangulations), then there holds

$$\|u - u_h\|_c \leq c(\sqrt{\bar{\varepsilon}}\lambda h + \sqrt{\xi}h^2 + \sqrt{|\mathbf{w}|_\infty}h^{\frac{3}{2}})\|u\|_{2,\Omega_{1,2}} \quad (2.3.39)$$

Proof. Combine lemma 2.3.27 and lemma 2.3.26. $\square$

2.3.2.7 Discussion of results

We comment on the bound derived in (2.3.39) and simplify the discussion by again assuming quasi-uniformity of the family of triangulations. The three error terms correspond to the three physical effects: diffusion ($\sqrt{\bar{\varepsilon}}h$), convection ($\sqrt{|\mathbf{w}|_\infty}h^{\frac{3}{2}}$) and reaction ($\sqrt{\xi}h^2$).

For the diffusion dominated case, i.e. $\bar{\varepsilon} \geq |\mathbf{w}|_\infty h$ and $\bar{\varepsilon} \geq \xi h^2$ we have $\gamma_T = \frac{h_T^2}{\bar{\varepsilon}}$ and the result essentially reduces to the one in section 2.3.1.

In the convection dominated case there holds $|\mathbf{w}|_\infty \geq \bar{\varepsilon}/h$ and $|\mathbf{w}|_\infty \geq \xi h$ and we thus have $\gamma_T \geq \frac{2h}{|\mathbf{w}|_\infty}$. Let $e_h := u - u_h$. From (2.3.39) we obtain

$$\|\mathbf{w} \cdot \nabla e_h\|_0 \leq c|\mathbf{w}|_\infty h \|u\|_{2,\Omega_{1,2}}.$$

Hence, as for the Streamline Diffusion finite element method with the standard linear finite element space, we have an optimal error bound (uniformly in $\bar{\varepsilon}$) for the derivative of the error in streamline direction.

Finally, if we take $\xi > 0$ (not necessarily reaction-domination) we obtain an L^2 -norm error bound that is the same as for the Streamline Diffusion finite element method with the standard linear finite element space, namely

$$\|e_h\|_0 \leq \frac{c}{\sqrt{\xi}} h^{\frac{3}{2}} \|u\|_{2,\Omega_{1,2}}.$$

We now discuss bounds for the error in the interface condition. We only discuss the convection-dominated case. The estimate (2.3.39) implies

$$\lambda \bar{\varepsilon} \|[\![\beta u_h]\!]\|_{\frac{1}{2},h,\Gamma}^2 \leq c|\mathbf{w}|_\infty h^3 \|u\|_{2,\Omega_{1,2}}^2.$$

For the *convective* scaling we have $\lambda \bar{\varepsilon} \sim ch$ and thus obtain

$$\|[\![\beta u_h]\!]\|_\Gamma \leq c\sqrt{|\mathbf{w}|_\infty} h^{\frac{1}{2}} \|u\|_{2,\Omega_{1,2}}$$

uniformly in $\bar{\varepsilon}$. For the *diffusive* scaling we have $\lambda \sim c$ and thus obtain the worse bound

$$\|[\![\beta u_h]\!]\|_{\Gamma} \leq ch^2 \bar{\varepsilon}^{-\frac{1}{2}} \|u\|_{2,\Omega_{1,2}}.$$

The *convective* scaling leads to error bounds for the streamline derivative and the jump term on the interface which are uniform in $\bar{\varepsilon}$.

Remark 2.3.3 (Extension to higher order discretizations). *The SD-Nitsche-XFEM method has a straightforward extension to finite elements of higher order. As soon as higher order finite elements are considered the analysis changes due to the term $(\operatorname{div}(\varepsilon \nabla u_h), \mathbf{w} \cdot \nabla v_h)_{0,h}$ which arises in the Streamline Diffusion stabilization. In the analysis of the Streamline Diffusion method for a standard higher order finite element space V_h one uses an inverse inequality of the form $\|\Delta v_h\|_{0,T} \leq c_{\text{inv}} h_T^{-1} |v_h|_{1,T}$ for all $v_h \in V_h$, cf. [RST08]. This is needed to prove lemma 2.3.21. The inverse inequality, however, does not hold for the XFEM functions in V_h^x as the support $T_i = T \cap \Omega_i$ can be very shape irregular. We only have $\|\Delta v_h\|_{0,T_i} \leq c(T_i) h_{T_i}^{-1} |v_h|_{1,T_i}$ with a factor $c(T_i)$ that depends on the shape regularity of T_i . To control this, instead of γ_T in (2.3.29), one can choose a stabilization parameter γ_{T_i} that is sufficiently small. This would yield a stability result as in lemma 2.3.21. If, however, this parameter is “too small” it is not likely that a result as in lemma 2.3.23, which uses the inequality (2.3.30d), still holds. A cure for this problem could be the higher order version of the ghost penalty stabilization (see section 2.2.3.5) which would allow for a control on $\|\Delta v_h\|_{0,T_{1,2}}$ independent of the shape regularity of T_i . We did not investigate this further.*

Remark 2.3.4 (Non-stationary case). *In the error analysis in section 2.3.2 we only studied the bilinear form for the quasi-stationary problem. Based on the techniques presented in the recent paper [BS11] it may be possible to derive, for the case of a stationary interface, error bounds for the semi-discrete problem (2.2.48). In view of applications the case of a non-stationary interface $\Gamma(t)$ is much more interesting than that of a stationary one. A discretization based on a space-time approach including an error analysis is presented and discussed in chapter 3.*

2.4 Preconditioning of linear systems

In this section we propose and analyze appropriate preconditioners for the linear systems arising from Nitsche-XFEM discretizations for unfitted interface problems. We restrict to the elliptic problem with convection, problem problem 2.1.3. The results in this section have also been published in [LR14].

Let $\mathbf{A}$ be the system matrix of the bilinear form $a_h(\cdot, \cdot)$ in (2.2.13). As a first quantity of interest we consider the spectral condition number $\kappa(\mathbf{A}) = \lambda_{\max}/\lambda_{\min}$ where $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and the smallest eigenvalues of the (symmetric positive definite) system matrix. In [BZ12] it is shown that $\kappa(\mathbf{A})$ depends on how the interface cuts an

element. In that paper an indicator ν for “small cut” situations is defined:

$$\nu := \min_{i=1,2} \min_{\substack{V \in \mathcal{V}, \\ \mathbf{x}_V \in \Omega^\Gamma \setminus \Omega_i}} \frac{|\omega_V| \cap \Omega_i}{|\omega_V|}. \quad (2.4.1)$$

Here, $\mathcal{V}$ is the set of vertices V of the triangulation $\mathcal{T}_h$ and $\mathbf{x}_V$ is the corresponding coordinate in Ω , $\omega_V = \{T \in \mathcal{T}, \mathbf{x}_V \in T\}$ denotes the patch of elements around a vertex V . For $\nu \searrow 0$ we have $\kappa(\mathbf{A}) \nearrow \infty$, s.t. the problem becomes (very) ill-conditioned.

A similar problem has already been observed for the mass matrix in [Reu08]. In that paper a remedy to this problem is a simple diagonal preconditioner.

In general, due to the fact that the conditioning of the stiffness matrix is sensitive to the position of the interface relative to the mesh, the efficient solution of the discrete problems stemming from Nitsche-XFEM discretizations is a challenging task. This is a topic that has hardly been addressed in the literature, so far. Recently, for stabilized versions of the Nitsche-XFEM method condition number bounds of the form ch^{-2} , with a constant c that is independent of how the interface Γ intersects the triangulation, have been derived in [BH12, HLZ14, ZWKB13]. In [HLZ14] an inconsistent stabilization is used to guarantee LBB-stability for the pair of finite element spaces used for the Stokes interface problem. This stabilization also improves the conditioning of the stiffness matrix, leading to a ch^{-2} condition number bound. In [ZWKB13] a stabilized variant of the Nitsche-XFEM discretization of problem 2.1.3 is considered. For this method an ch^{-2} condition number bound is derived. We also mention the engineering paper [LMDM14], where the *unstable* Nitsche-XFEM formulation with weighting $\kappa_i = 0.5$, $i = 1, 2$ is considered and a special preconditioner is designed to obtain solvable system of equations.

In this section we consider the Nitsche-XFEM discretization (2.2.14) with piecewise linear functions ($k = 1$) for the stationary and self-adjoint two-domain Poisson problem (Problem (2.1.3)), without any additional stabilization.

We propose an *optimal preconditioner* which is constructed from approximate subspace corrections. Here, optimality means that the condition number of the preconditioned matrix is independent of h and of how the interface Γ intersects the triangulation and applications at the preconditioner have only $\mathcal{O}(N)$ complexity with $N = \dim(V_h^\Gamma)$. The most important ingredient is the (uniform in h) stable decomposition of V_h^Γ into the subspaces V_h and V_h^x . To emphasize that V_h is a subspace of V_h^Γ we introduce the notation $V_h^s = V_h$. If in the subspace V_h^s one applies a standard multigrid preconditioner and in the subspace V_h^x one applies a simple Jacobi diagonal scaling, the resulting *additive subspace preconditioner is optimal*. The latter is the main result of this section. The analysis uses the very general theory of subspace correction methods [Xu92, Yse93]. Our analysis applies to the two-dimensional case ($d = 2$), but we expect that a very similar optimality result holds for $d = 3$. This claim is supported by results of numerical experiments in section 2.5.3.

The structure of this section is as follows. First, we motivate and apply a transformation of the discrete problem in section 2.4.1 essentially reducing the problem to the case

$\beta_1 = \beta_2 = 1$. In section 2.4.2 we introduce notation for block matrices and assumptions which are needed in the proceeding sections. In section 2.4.3 we discuss a stable decomposition of V_h^Γ into three subspaces with respect to a norm which is natural for the Nitsche-XFEM discretization. The decomposition result yields spectral equivalence of the stiffness matrix and a block diagonal matrix with blocks corresponding to the subspaces V_h^s and V_h^x . The use of approximations for the block matrices is discussed in section 2.4.4. Here we propose an optimal preconditioner which consists of a multigrid solver on V_h^s and diagonal preconditioning within the XFEM subspace V_h^x . The diagonal preconditioning within the XFEM subspace is discussed in section 2.4.5. In section 2.4.6 we discuss extensions of the theoretical result. We remark on a Jacobi preconditioner for the full space V_h^Γ , discuss parameter dependencies of the theoretical results (w.r.t. α and λ) and extensions to unsteady problems.

2.4.1 Basis transformation

While on the continuous level we avoid to apply a reformulation as presented in section 1.2.3, we consider a similar transformation on the discrete level. Note that for every $v \in V_h^\Gamma$ there holds $\beta v \in V_h^\Gamma$ et vice versa. Let $N = \dim(V_h^\Gamma)$. If we replace the basis $U = \{\varphi_j(\mathbf{x})\}_{j=1,\dots,N}$ with $W = \{\beta^{-1}(\mathbf{x})\varphi_j(\mathbf{x})\}_{j=1,\dots,N}$ we neither change the test nor the ansatz space und therefore the solution stays the same. However the representation of the solution in terms of the coefficient vectors is different. Let $\mathbf{u}$ denote the coefficient vector corresponding to the basis U and $\mathbf{w}$ the coefficient vector corresponding to the basis W with (scalar) components $\mathbf{u}_j$ and $\mathbf{w}_j$, respectively. Then we have for $u \in V_h^\Gamma$

$$u(\mathbf{x}) = \sum_{j=1}^N \mathbf{u}_j \varphi_j(\mathbf{x}) = \sum_{j=1}^N \mathbf{w}_j \beta^{-1}(\mathbf{x}) \varphi_j(\mathbf{x}). \quad (2.4.2)$$

Between U and W we define the (mixed) mass matrices $\mathbf{M}^{UU}, \mathbf{M}^{UW} \in \mathbb{R}^{n \times n}$ and the transformation matrix $\mathbf{T} \in \mathbb{R}^{n \times n}$:

$$\mathbf{M}_{k,j}^{UU} := \int_{\Omega} \varphi_k \varphi_j d\mathbf{x}, \quad \mathbf{M}_{k,j}^{UW} := \int_{\Omega} \beta^{-1} \varphi_k \varphi_j d\mathbf{x}, \quad \mathbf{T} := (\mathbf{M}^{UU})^{-1} \mathbf{M}^{UW}. \quad (2.4.3)$$

Note that the transformation from one basis into another is not a local problem, it involves the solution with a global mass matrix. The solution of a mass matrix problem, however, can be obtained efficiently using a diagonally preconditioned CG, cf. [Reu08].

If we use the basis W to formulate our linear systems, the system matrix changes to

$$\begin{aligned} \tilde{\mathbf{A}}_{k,j} = & (\tilde{\alpha} \nabla \varphi_k, \nabla \varphi_j)_{L^2(\Omega_{1,2})} - (\{\tilde{\alpha} \nabla \varphi_k \cdot \mathbf{n}\}, [\varphi_j])_{L^2(\Gamma)} \\ & - (\{\tilde{\alpha} \nabla \varphi_j \cdot \mathbf{n}\}, [\varphi_k])_{L^2(\Gamma)} + (\tilde{\tilde{\lambda}} \frac{\tilde{\lambda}}{h} [\varphi_k], [\varphi_j])_{L^2(\Gamma)}, \quad j, k = 1, \dots, N \end{aligned}$$

with $\tilde{\alpha} = \frac{\alpha}{\beta}$ and $\tilde{\tilde{\lambda}} = \lambda \tilde{\alpha} / \tilde{\alpha}$. Between the formulation in basis U and W there holds the relation

$$\tilde{\mathbf{A}} = \mathbf{T}^T \mathbf{A} \mathbf{T}.$$

In the following we consider the linear systems according to the basis W . To this end we transform the parameters $\alpha \rightarrow \frac{\alpha}{\beta}$, $\beta \rightarrow 1$, $\lambda \rightarrow \lambda\bar{\alpha}/\tilde{\alpha}$ which allows us to keep the notation for the bilinear and linear forms from the preceding sections. For notational convenience, in the remainder of this chapter we will denote the matrix corresponding to the transformed basis by $\mathbf{A}$. The system matrix *without* transformation is then denoted by $\hat{\mathbf{A}} := \mathbf{T}^{-\mathbf{T}}\mathbf{A}\mathbf{T}^{-1}$.

After transformation ($\beta \rightarrow 1$), the (transformed) solution is continuous across the interface and from a linear solvers point of view the problem setting is the same as in [BH12, BZ12, HH02] (and many other publications).

The benefit of this transformation is twofold. First, for all $u \in V_h^s$ there holds $\llbracket u \rrbracket = 0$ at the interface and thus $N_s^H(u, v) = 0$ which does not hold without the transformation. This property is important to prove the stable decomposition result in section 2.4.3. Further the robustness w.r.t. parameter changes in β seems to be significantly increased, cf. the numerical experiments in section 2.5.3.

We note that in the transformed case β is transformed to 1 so that the β -weighted norms and scalar products coincide with the standard L^2 and H^1 versions.

A comparison of the properties of the linear systems before and after the transformation is presented (among other aspects) in section 2.5.3 based on numerical experiments.

Remark 2.4.1 (Preconditioner for the untransformed case). *Assume $\mathbf{C}$ is a good preconditioner for $\mathbf{A}$ in some sense. The question arises if a preconditioner $\hat{\mathbf{C}}$ for $\hat{\mathbf{A}}$ can be designed which has the same quality. Such a preconditioner can be obtained using the basis transformation again. With $\hat{\mathbf{C}} = \mathbf{T}^{-\mathbf{T}}\mathbf{C}\mathbf{T}^{-1}$ there holds $\kappa(\mathbf{C}\mathbf{A}) = \kappa(\hat{\mathbf{C}}\hat{\mathbf{A}})$. Note however, that in each application of the preconditioner $\hat{\mathbf{C}}$ one has to solve two global mass matrix problems. It is thus typically more efficient to transform the linear system first before the application of an iterative solver.*

2.4.2 Preliminaries

In this section we introduce additional assumptions and notation needed later to provide a rigorous proof of condition number bounds for preconditioned system matrices. We further recall a property of stable Nitsche-XFEM discretizations which results in an important property of the system matrix $\mathbf{A}$.

2.4.2.1 Assumptions

Assumption 2.4.1 (Small contrast problem). *We assume that the coefficient α_i , $i = 1, 2$ are moderate in the sense that there exists a moderate constant $c > 1$ such that $\frac{1}{c} \leq \alpha_i \leq c$, $i = 1, 2$.*

Assumption 2.4.2 (Separated vertices). *We assume that on every cut element $T \in \mathcal{T}_h^\Gamma$ at least one vertex V is in Ω_1 and at least one vertex V is in Ω_2 .*

Remark 2.4.2 (Element-wise planar interface approximation). *One can ensure assumption 2.4.2 by replacing the interface Γ by an element-wise planar approximation Γ_h of the interface.*

Assumption 2.4.3 (Resolution of the interface). *Let $\mathbf{x}_V$ be the location of a vertex V . The vertex patch $\omega_V = \{T \in \mathcal{T}_h, \mathbf{x}_V \in T\}$ around each vertex V contains at least one element which is not intersected by the interface $T \notin \mathcal{T}_h^\Gamma$. Each vertex patch ω_V contains at most one connected interface. Further every connected part of the interface Γ_j with $\Gamma = \bigcup_j \Gamma_j$ is at least of the size of h : $|\Gamma_j| \geq h$.*

For a smooth interface this assumption is always justified as long as the mesh size is sufficiently small.

Assumption 2.4.4 (Shape regular two dimensional simplex mesh). *We assume $\Omega \subset \mathbb{R}^d$ with $d = 2$ and Ω is decomposed into a triangulation $\mathcal{T}_h$ of simplices T which is shape regular.*

We further restrict to the case of piecewise linear finite elements and only consider the discretization in (2.2.14). We especially do not consider weightings in the averaging different from $\{\!\!\}\!\!\} = \{\!\!\}\!\!\}^H$ and no additional stabilization techniques. We also restrict to the case of homogeneous Dirichlet boundary conditions.

From a practical point of view the only real restrictions are assumption 2.4.1, the assumption of a two dimensional domain (assumption 2.4.4) and the boundary conditions. The restriction to two dimensional domains is only added for technical reasons in the proof. We expect that the resulting estimates also hold in three dimensions. The assumption on the boundary conditions is made for ease of presentation. In section 2.4.6 we discuss the influence of violations to assumption 2.4.1.

2.4.2.2 Notation

In section 2.2.1.2 we introduced the XFEM characterization of V_h^Γ as the direct sum of the spaces V_h^x and V_h^s . Now we further divide V_h^x into $V_{h,1}^x$ and $V_{h,2}^x$ with

$$V_{h,i}^x := \{v \in V_h^x, \text{supp}(v) \subset \Omega_i\}.$$

We recall the definition 2.2.4 of the Galerkin isomorphism $\mathcal{G}$ and denote by N , N_s and N_x the number of degrees of freedom in V_h^Γ , V_h^s and V_h^x respectively. Let $\mathbf{A}$ be the system matrix of the bilinear form $a_h(\cdot, \cdot)$ in (2.2.13) in the sense that there holds $\mathbf{v}^T \mathbf{A} \mathbf{u} = a_h(\mathcal{G} \mathbf{u}, \mathcal{G} \mathbf{v}) \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^N$.

We introduce notation for the splitting of coefficient vectors $\mathbf{u} \in \mathbb{R}^N$ and finite element functions $u \in V_h^\Gamma$ into its standard FEM and XFEM parts. Given a coefficient vector $\mathbf{u} \in \mathbb{R}^N$ representing a discrete function $u \in V_h^\Gamma$, s.t. $\mathcal{G} \mathbf{u} = u$. We assume that the degrees of freedom are sorted so that the first N_s degrees of freedom correspond to a function in V_h^s . Then we can define the restriction matrices $\mathbf{E}_s = (\mathbf{I}_N, \mathbf{0}) \in \mathbb{R}^{N_s \times N}$ and $\mathbf{E}_x = (\mathbf{0}, \mathbf{I}_{N_x}) \in \mathbb{R}^{N_x \times N}$ where $\mathbf{I}_n$ denotes the identity matrix in $\mathbb{R}^{n \times n}$, $n \in \{N, N_x\}$.

The restriction matrices extract the coefficient vectors for the standard FEM or XFEM function, respectively, such that we have $\mathcal{G}(\mathbf{E}_s \mathbf{u}) \in V_h^s$ and $\mathcal{G}(\mathbf{E}_x \mathbf{u}) \in V_h^x$ for every $u \in V_h^\Gamma$. Further for every matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$ we define block matrices corresponding to standard FEM or XFEM functions as $\mathbf{M}_s := \mathbf{E}_s \mathbf{M} \mathbf{E}_s^\mathbf{T} \in \mathbb{R}^{N_s \times N_s}$ and $\mathbf{M}_x := \mathbf{E}_x \mathbf{M} \mathbf{E}_x^\mathbf{T} \in \mathbb{R}^{N_x \times N_x}$.

For the norm induced by the bilinear form $a_h(\cdot, \cdot)$ we use the notation

$$\|u\|_h := a_h(u, u)^{\frac{1}{2}}, \quad u \in V_h^\Gamma.$$

We further introduce the triple-norm

$$\|u\|_h := ((u, u)_{1, \Omega_{1,2}} + \lambda \|u\|_{\frac{1}{2}, h, \Gamma}^2)^{\frac{1}{2}}.$$

Note that both norms are uniformly equivalent for a sufficiently large λ .

For ease of presentation it is convenient to be able to hide generic constants in some estimates. For this purpose we recall the definition 2.2.1 of the relations $\preceq$, $\succeq$, $\simeq$, where the constant c in that definition is independent on h and how the interface cuts through the mesh. Note that especially dependencies on α and λ can be absorbed in this notation. As long as assumption 2.4.1 is fulfilled the dependencies on α are only mild.

In what follows, whenever we discuss the condition number $\kappa(\mathbf{M})$ of a symmetric positive definite matrix $\mathbf{M}$, we refer to the spectral condition number, the ratio between largest and smallest eigenvalue of $\mathbf{M}$.

2.4.3 Stable subspace splittings of V_h^Γ

We will derive an optimal preconditioner for the bilinear form $a_h(\cdot, \cdot)$ using the theory of *subspace correction methods*. Two excellent overview papers on this topic are [Xu92, Yse93]. The theory of subspace correction methods as described in these overview papers is a very general one, with applications to multigrid and domain decomposition methods. We apply it for a relatively very simple case with three disjoint spaces. We use the notation and some main results from [Yse93]. The three subspaces of $\mathcal{S} := \mathcal{V}_h^\Gamma$ are denoted by $\mathcal{W}_0 := V_h^s$, $\mathcal{W}_i := V_{h,i}^x$, $i = 1, 2$. Thus we have the direct sum decomposition

$$\mathcal{S} = \mathcal{W}_0 \oplus \mathcal{W}_1 \oplus \mathcal{W}_2. \quad (2.4.4)$$

Below $u = u_0 + u_1 + u_2 \in \mathcal{S}$ always denotes a decomposition with $u_l \in \mathcal{W}_l$, $l = 0, 1, 2$. In theorem 2.4.3 below we show that the splitting in (2.4.4) is stable w.r.t. the norm $\|\cdot\|_h$.

2.4.3.1 Stable subspace splitting of $\mathcal{S}$ into $\mathcal{W}_0$ and $\mathcal{W}_1 \oplus \mathcal{W}_2$

The result in the next theorem is the key point in this analysis. We show that the splitting of $\mathcal{S}$ into $\mathcal{W}_0$ and the subspace spanned by the XFEM basis functions $\mathcal{W}_1 \oplus \mathcal{W}_2$

is stable. For this we restrict to the two-dimensional case $d = 2$. We use a transformation of certain patches to a reference patch on $[0, 1]^2$. We first describe this transformation. We construct a subdivision of $\mathcal{T}_h^\Gamma$ into patches $\{\omega_k\}$ as follows, cf. figure 2.4.1. We first define a subset $\mathcal{E}$ of all edges that are intersected by Γ . Consider an edge E_1 which is intersected by Γ such that one vertex V_1 is in Ω_1 and the other, V_1^* , is in Ω_2 . We define this edge as the first element in $\mathcal{E}$. Now fix one direction along the interface and going in this direction along Γ we get an ordered list of all edges intersected by Γ . As last edge in this list we include the starting edge E_1 . As the next edge $E_2 \in \mathcal{E}$ we take the first one after E_1 (in the list) that has no common vertex with E_1 . As $E_3 \in \mathcal{E}$ we take the first one after E_2 that has no common vertex with E_2 , etc.. To avoid technical details we assume that the final edge $E_{N_\mathcal{E}}$ included in $\mathcal{E}$ coincides with E_1 . By construction we get a numbering of certain vertices as in the left part of figure 2.4.1: edge E_j has vertices $V_j \in \Omega_1$, $V_j^* \in \Omega_2$.

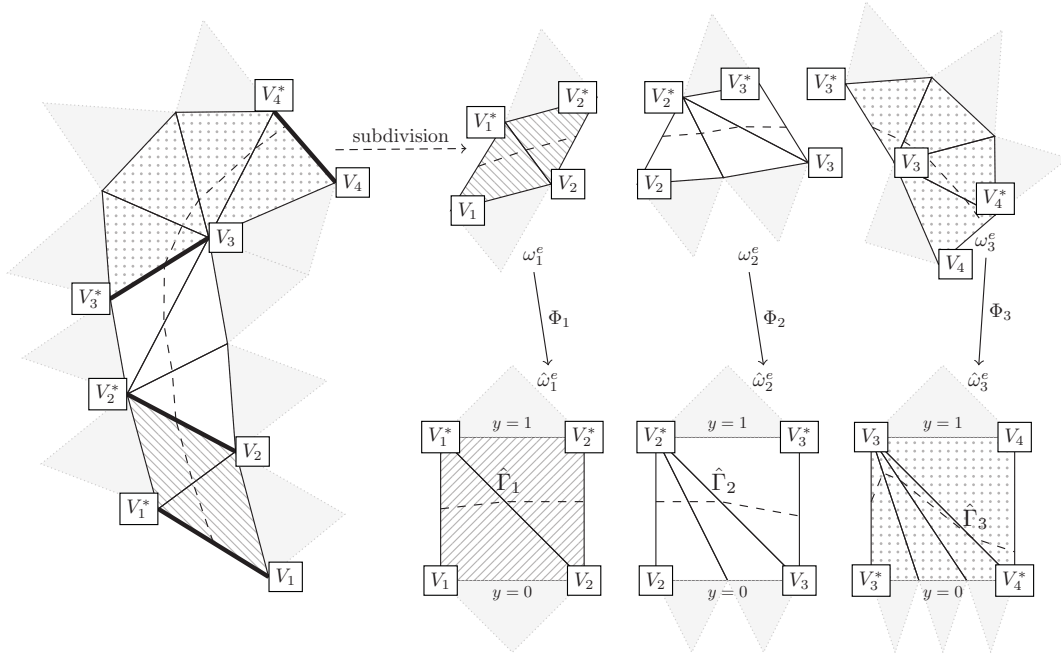


Figure 2.4.1: Sketch of the partitioning of $\mathcal{T}_h^\Gamma$ (and neighboring elements) into (extended) patches ω_k^e and their transformations to a reference configurations.

The elements between two edges $E_k, E_{k+1} \in \mathcal{E}$ form the patch ω_k . The patches $\{\omega_k\}_{1 \leq k \leq N_\omega}$, with $N_\omega = N_\mathcal{E} - 1$, form a disjoint partitioning of $\mathcal{T}_h^\Gamma$. We define the extended patch ω_k^e by adding the neighboring elements which are not in $\mathcal{T}_h^\Gamma$, i.e., $\omega_k^e := \omega_k \cup \{T \in \mathcal{T}_h \setminus \mathcal{T}_h^\Gamma \mid T \text{ has a common edge with a } T' \in \omega_k\}$. The part of the interface Γ contained in ω_k^e is denoted by Γ_k . The triangulation (and corresponding domain) formed by the union of the extended patches ω_k^e is denoted by $\mathcal{T}_h^{\Gamma,e}$. Note that every element $T \in \mathcal{T}_h^{\Gamma,e}$ can appear in at most two patches ω_k^e . Further note that the number of elements within each extended patch ω_k^e is uniformly bounded due to shape regularity of $\mathcal{T}_h$. For each extended patch ω_k^e there exists a piecewise affine transformation

$\Phi_k : \omega_k^e \rightarrow \mathbb{R}^2$ such that $\Phi_k(\omega_k) = [0, 1]^2$. Accordingly we denote a transformed patch by $\hat{\omega}$ and $\hat{\omega}^e$.

Theorem 2.4.1.

Take $d = 2$. There exists a constant C_2 , independent of h and of how the triangulation is intersected by Γ , such that

$$\|u_0\|_h^2 + \|w\|_h^2 \leq C_2 \|u_0 + w\|_h^2 \quad \text{for all } u_0 \in \mathcal{W}_0, w \in \mathcal{W}_1 \oplus \mathcal{W}_2. \quad (2.4.5)$$

Proof. Due to norm equivalence the result in (2.4.5) is equivalent to:

$$\|u_0\|_h^2 + \|w\|_h^2 \preceq \|u_0 + w\|_h^2 \quad \text{for all } u_0 \in \mathcal{W}_0, w \in \mathcal{W}_1 \oplus \mathcal{W}_2.$$

For $w \in \mathcal{W}_1 \oplus \mathcal{W}_2$ we have $w = 0$ on $\Omega \setminus \mathcal{T}_h^{\Gamma,e}$, and $\mathcal{T}_h^{\Gamma,e}$ is partitioned into patches ω_k^e . Hence, it suffices to prove

$$\|u_0\|_{h,\omega_k^e}^2 + \|w\|_{h,\omega_k^e}^2 \preceq \|u_0 + w\|_{h,\omega_k^e}^2 \quad \text{for all } u_0 \in \mathcal{W}_0, w \in \mathcal{W}_1 \oplus \mathcal{W}_2. \quad (2.4.6)$$

We use the transformation to the reference patch $\hat{\omega}^e$ described above. On the reference patch we have transformed spaces $\hat{\mathcal{W}}_0$ (continuous, piecewise linear functions) and $\hat{\mathcal{W}}_1 \oplus \hat{\mathcal{W}}_2$. The functions in $\hat{\mathcal{W}}_1$ ($\hat{\mathcal{W}}_2$) are piecewise linear on the part of the patch below (above) the interface $\hat{\Gamma}$, zero on the line segment $y = 0$ ($y = 1$) and zero on the part of the patch above (below) the interface $\hat{\Gamma}$. The norm $\|u\|_{h,\omega_k^e}$ and the induced norm $\|\hat{u}\|_{\hat{\omega}_k^e} = ((\nabla \hat{u}, \nabla \hat{u})_{L^2(\hat{\omega}_k^e)} + \lambda([\hat{u}], [\hat{u}])_{L^2(\hat{\Gamma}_k)})^{\frac{1}{2}}$, with $\hat{u} = u \circ \Phi_k^{-1}$ on $\hat{\omega}_k^e$, are uniformly equivalent, because the constants in this norm equivalence are determined only by the condition number of the piecewise affine transformation between ω_k^e and $\hat{\omega}_k^e$. Note that neither the spaces $\hat{\mathcal{W}}_i$ nor the norm $\|\cdot\|_{\hat{\omega}_k^e}$ depend on h (the h -dependence is implicit in the piecewise affine transformation). The reference patches $\hat{\omega}_k^e$ all have the same geometric structure, cf. figure 2.4.1. These patches have (due to shape regularity of $\mathcal{T}_h$) a uniformly bounded number of vertices on the line segment that connects the vertices V_i, V_{i+1} (or V_i^*, V_{i+1}^*). In the rest of the proof a generic reference patch and its extension are denoted by $\hat{\omega}$ and $\hat{\omega}^e$, respectively. The interface segment that is intersected by $\hat{\omega}$ is denoted by $\hat{\Gamma}$. We conclude that for (2.4.6) to hold it is sufficient to prove

$$\|u_0\|_{\hat{\omega}^e}^2 + \|w\|_{\hat{\omega}^e}^2 \leq K \|u_0 + w\|_{\hat{\omega}^e}^2 \quad \text{for all } u_0 \in \hat{\mathcal{W}}_0, w \in \hat{\mathcal{W}}_1 \oplus \hat{\mathcal{W}}_2, \quad (2.4.7)$$

with a constant K that is independent of how the patch $\hat{\omega}$ is intersected by the interface $\hat{\Gamma}$. Note that $(\nabla u_0, \nabla w)_{L^2(\hat{\omega}^e \setminus \hat{\omega})} = ([u_0], [w])_{L^2(\hat{\Gamma})} = 0$ for $u_0 \in \hat{\mathcal{W}}_0$ and $w \in \hat{\mathcal{W}}_1 \oplus \hat{\mathcal{W}}_2$. Hence,

$$\|u_0 + w\|_{\hat{\omega}^e}^2 = \|u_0\|_{\hat{\omega}^e}^2 + \|w\|_{\hat{\omega}^e}^2 + 2(\nabla u_0, \nabla w)_{L^2(\hat{\omega})}, \quad u_0 \in \hat{\mathcal{W}}_0, w \in \hat{\mathcal{W}}_1 \oplus \hat{\mathcal{W}}_2$$

holds. Thus it suffices to prove the strengthened Cauchy-Schwarz inequality

$$(\nabla u_0, \nabla w)_{L^2(\hat{\omega})} \leq C^* \|u_0\|_{\hat{\omega}^e} \|w\|_{\hat{\omega}^e} \quad \text{for all } u_0 \in \hat{\mathcal{W}}_0, w \in \hat{\mathcal{W}}_1 \oplus \hat{\mathcal{W}}_2, \quad (2.4.8)$$

with a uniform constant $C^* < 1$. The proof of (2.4.8) is divided into three steps, namely a strengthened Cauchy-Schwarz inequality related to the x -derivative, a suitable Cauchy-Schwarz inequality related to the y -derivative and then combining these estimates.

Step 1. The following holds:

$$|(u_x, w_x)_{L^2(\hat{\omega})}| \leq c_0 \|u_x\|_{L^2(\hat{\omega}^e)} \|w_x\|_{L^2(\hat{\omega})} \quad \text{for all } u \in W_0, w \in \hat{W}_1 \oplus \hat{W}_2, \quad (2.4.9)$$

with a uniform constant $c_0 < 1$. From the Cauchy-Schwarz inequality we get $|(u_x, w_x)_{L^2(\hat{\omega})}| \leq \|u_x\|_{L^2(\hat{\omega})} \|w_x\|_{L^2(\hat{\omega})}$. Within the patch $\hat{\omega} = \{T_i\}$ the x -derivative u_x is piecewise constant and $u_x|_{T_i} = u_x|_{T_{i,N}}$ for the neighboring triangle $T_{i,N} \in \hat{\omega}^e \setminus \hat{\omega}$. This implies $\|u_x\|_{L^2(T_i)} \leq \hat{c} \|u_x\|_{L^2(T_i \cup T_{i,N})}$, with $\hat{c} < 1$ depending only on shape regularity. Thus we obtain $\|u_x\|_{L^2(\hat{\omega})} \leq c_0 \|u_x\|_{L^2(\hat{\omega}^e)}$, with a uniform constant $c_0 < 1$, which yields (2.4.9).

Step 2. The following holds:

$$\begin{aligned} |(u_y, w_y)_{L^2(\hat{\omega})}| &\leq \min\{c_1 \|u_x\|_{L^2(\hat{\omega})}, \|u_y\|_{L^2(\hat{\omega})}\} \|w_y\|_{L^2(\hat{\omega})} \\ &\quad + c_2 \|u_y\|_{L^2(\hat{\omega})} \|\llbracket w \rrbracket\|_{L^2(\hat{\Gamma})} \quad \text{for all } u \in W_0, w \in \hat{W}_1 \oplus \hat{W}_2, \end{aligned} \quad (2.4.10)$$

with suitable uniform constants c_1, c_2 .

Let $\{T_i\}$ be the set of triangles that form $\hat{\omega}$ and let these be ordered such that $\text{meas}_1(T_i \cap T_{i+1}) > 0$. We denote the interior edges by $e_i = T_i \cap T_{i+1}$. To show (2.4.10) we start with partial integration

$$\begin{aligned} \left| \int_{\hat{\omega}} u_y w_y dx \right| &= \left| \sum_{T_i} \int_{\partial T_i} n_{T_i, y} u_y w ds + \int_{\hat{\Gamma}_{T_i}} n_{\Gamma, y} u_y \llbracket w \rrbracket ds \right| \\ &\leq \sum_{e_i} \left| \llbracket u_y \rrbracket_{e_i} \right| \left| \int_{e_i} w ds \right| + \|u_y\|_{L^2(\hat{\Gamma})} \|\llbracket w \rrbracket\|_{L^2(\hat{\Gamma})} \end{aligned} \quad (2.4.11)$$

where for the edges of ∂T_i that lie on $\partial \hat{\omega} = \partial[0, 1]^2$ we used $w = 0$ for $y \in \{0, 1\}$ and $n_{T_i, y} = 0$ for $x \in \{0, 1\}$. To proceed we need technical estimates to bound $\llbracket u_y \rrbracket_{e_i}$ and $\int_{e_i} w ds$. For those estimates we exploit properties of the geometry of $\hat{\omega}$. First consider $u \in \hat{W}_0$ along an interior edge $e_i \not\subset \partial \hat{\omega}$ and denote the unit tangential vector to e_i by $\mathbf{t} = (\tau_x, \tau_y)$. For τ we have $|\tau_y| \geq 1/\sqrt{2} \geq |\tau_x|$. Due to continuity of u along e_i there holds $\llbracket \nabla u \rrbracket_{e_i} \cdot \mathbf{t} = 0$, which implies

$$|\llbracket u_y \rrbracket_{e_i}| = \left| \frac{\tau_x}{\tau_y} \right| |\llbracket u_x \rrbracket_{e_i}| \leq |u_x|_{T_i} + |u_x|_{T_{i+1}}.$$

Thus we obtain

$$|\llbracket u_y \rrbracket_{e_i}| \leq c \min\{\|u_y\|_{L^2(T_i \cup T_{i+1})}, \|u_x\|_{L^2(T_i \cup T_{i+1})}\}. \quad (2.4.12)$$

Next, we consider $w = w_1 + w_2 \in \hat{W}_1 \oplus \hat{W}_2$ along the interior edge e_i . Let T_i be a triangle adjacent to e_i . Without loss of generality we assume that two vertices of

T_i are in $\hat{\Omega}_1$ and we thus have $(w_1)_x = 0$ on T_i . We denote the vertices of e_i by $\mathbf{x}_j = e_i \cap \partial\hat{\omega} \cap \hat{\Omega}_j$, $j = 1, 2$ and the intersection point by $\mathbf{x}_\Gamma = e_i \cap \hat{\Gamma}$ and define the distances $d_j = \|\mathbf{x}_j - \mathbf{x}_\Gamma\|_2$, $j = 1, 2$. As w is piecewise linear along e_i , zero at $\mathbf{x}_1$, and $(w_1)_x = 0$ on T_i , we have $w_1(\mathbf{x}_\Gamma) = \pm d_1 \tau_y(w_1)_y$. Furthermore:

$$\int_{e_i} w \, ds = \frac{1}{2} d_1 w_1(\mathbf{x}_\Gamma) + \frac{1}{2} d_2 w_2(\mathbf{x}_\Gamma) = \frac{1}{2} (d_1 + d_2) w_1(\mathbf{x}_\Gamma) - \frac{1}{2} d_2 \llbracket w \rrbracket(\mathbf{x}_\Gamma).$$

We also have the geometrical information $d_1 \leq d_1 + d_2 \leq \sqrt{2}$, $d_1 \leq c|T_i|^{\frac{1}{2}}$, $|\hat{\Gamma}_{T_i}| \leq \sqrt{2}$ and $d_2 \leq c|\hat{\Gamma}_{T_i}|^{\frac{1}{2}}$. Because $\llbracket w \rrbracket$ is linear along $\hat{\Gamma}_{T_i}$ there also holds $|\hat{\Gamma}_{T_i}|^{\frac{1}{2}} |\llbracket w \rrbracket(\mathbf{x}_\Gamma)| \leq c \|\llbracket w \rrbracket\|_{L^2(\hat{\Gamma}_{T_i})}$. Using these results we get

$$\left| \int_{e_i} w \, ds \right| \leq c \|\llbracket w \rrbracket\|_{L^2(\hat{\Gamma}_{T_i})} + c \|w_y\|_{L^2(T_i)}. \quad (2.4.13)$$

From (2.4.12) and (2.4.13) we obtain

$$\sum_{e_i} \left| \llbracket u_y \rrbracket_{e_i} \right| \left| \int_{e_i} w \, ds \right| \leq c \|u_y\|_{L^2(\hat{\omega})} \|\llbracket w \rrbracket\|_{L^2(\hat{\Gamma})} + c \|u_x\|_{L^2(\hat{\omega})} \|w_y\|_{L^2(\hat{\omega})}. \quad (2.4.14)$$

Combining (2.4.11), (2.4.14) and the Cauchy-Schwarz inequality $\left| \int_{\hat{\omega}} u_y w_y \, dx \right| \leq \|u_y\|_{L^2(\hat{\omega})} \|w_y\|_{L^2(\hat{\omega})}$ results in (2.4.10).

Step 3. The following holds:

$$|(\nabla u, \nabla w)_{L^2(\hat{\omega})}| \leq C^* (\|u_x\|_{L^2(\hat{\omega}^e)} + \|u_y\|_{L^2(\hat{\omega})})^{\frac{1}{2}} (\|\nabla w\|_{L^2(\hat{\omega})}^2 + \lambda \|\llbracket w \rrbracket\|_{L^2(\hat{\Gamma})}^2)^{\frac{1}{2}} \quad (2.4.15)$$

for all $u \in W_0$, $w \in \hat{W}_1 \oplus \hat{W}_2$, with a uniform constant $C^* < 1$.

The proof combines the preceding results. We define $\alpha_x = \|u_x\|_{L^2(\hat{\omega}^e)}$, $\beta_x = \|w_x\|_{L^2(\hat{\omega})}$, $\alpha_y = \|u_y\|_{L^2(\hat{\omega})}$, $\beta_y = \|w_y\|_{L^2(\hat{\omega})}$, $\gamma = \|\llbracket w \rrbracket\|_{L^2(\hat{\Gamma})}$. Then we have with (2.4.9), (2.4.10) and $\theta = \frac{\alpha_x^2}{\alpha_x^2 + \alpha_y^2}$, $\alpha = (\alpha_x^2 + \alpha_y^2)^{\frac{1}{2}}$ and $\beta = (\beta_x^2 + \beta_y^2 + \lambda \gamma^2)^{\frac{1}{2}}$

$$\begin{aligned} |(\nabla u, \nabla w)_{L^2(\hat{\omega})}| &\leq c_0 \alpha_x \beta_x + \min\{c_1 \alpha_x, \alpha_y\} \beta_y + c_2 \alpha_y \gamma \\ &\leq (c_0^2 \alpha_x^2 + \min\{c_1^2 \alpha_x^2, \alpha_y^2\} + c_2^2 \alpha_y^2 \lambda^{-1})^{\frac{1}{2}} (\beta_x^2 + \beta_y^2 + \lambda \gamma^2)^{\frac{1}{2}} \\ &\leq (c_0^2 \theta + \min\{c_1^2 \theta, 1 - \theta\} + c_2^2 (1 - \theta) \lambda^{-1})^{\frac{1}{2}} \alpha \beta. \end{aligned}$$

One easily sees that $c_0^2 \theta + \min\{c_1^2 \theta, 1 - \theta\} \leq \frac{c_0^2 + c_1^2}{1 + c_1^2} < 1$. For sufficiently large λ ($\lambda > \frac{1 + c_1^2}{c_2^2 (1 - c_0^2)}$) (2.4.15) follows for a suitable uniform constant $C^* < 1$.

The result (2.4.15) directly implies (2.4.8) and thus the estimate (2.4.5) holds for λ sufficiently large. For different values $\lambda \geq \lambda^*$, with λ^* the critical value for which the norm equivalence $\|\cdot\|_h \simeq \|\cdot\|_h$ holds, the norms $\|\cdot\|_h$ (depending on λ) are equivalent, with equivalence constants depending only on λ . This implies that (2.4.5) holds for any $\lambda \geq \lambda^*$. $\square$

Remark 2.4.3 (Block preconditioner). *A direct conclusion of theorem 2.4.1 is*

$$\sum_{l \in \{s, x\}} \mathbf{u}^T \mathbf{E}_l^T \mathbf{A}_l \mathbf{E}_l \mathbf{u} \leq C_2 \mathbf{u}^T \mathbf{A} \mathbf{u} \quad \text{for all } \mathbf{u} \in \mathbb{R}^N. \quad (2.4.16)$$

We also have

$$\mathbf{u}^T \mathbf{A} \mathbf{u} \leq 2 \sum_{l \in \{s, x\}} \mathbf{u}^T \mathbf{E}_l^T \mathbf{A}_l \mathbf{E}_l \mathbf{u} \quad \text{for all } \mathbf{u} \in \mathbb{R}^N \quad (2.4.17)$$

and can thus deduce the following property of a block diagonal preconditioner

$$\kappa(\mathbf{B}_\mathbf{A}^{-1} \mathbf{A}) \leq 2C_2, \quad \text{with } \mathbf{B}_\mathbf{A} := \begin{pmatrix} \mathbf{A}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_x \end{pmatrix}.$$

In the next section a stable subspace splitting of $\mathcal{W}_1 \oplus \mathcal{W}_2$ is presented.

2.4.3.2 Stable subspace splitting of $\mathcal{W}_1 \oplus \mathcal{W}_2$ into $\mathcal{W}_1$ and $\mathcal{W}_2$

In the next lemma we derive the stable splitting property of $\mathcal{W}_1 \oplus \mathcal{W}_2$.

Lemma 2.4.2.

There exist constants c_γ, C_γ, C_x , independent of h and of how the triangulation is intersected by Γ , such that

$$c_\gamma \|u_l\|_h^2 \leq |u_l|_{1, \Omega_l}^2 \leq C_\gamma \|u_l\|_h^2 \quad \text{for all } u_l \in \mathcal{W}_l \quad \text{and } l = 1, 2, \quad (2.4.18)$$

$$\|u_1\|_h^2 + \|u_2\|_h^2 \leq C_x \|u_1 + u_2\|_h^2 \quad \text{for all } u_1 + u_2 \in \mathcal{W}_1 \oplus \mathcal{W}_2. \quad (2.4.19)$$

Proof. Take $l = 1$. We have

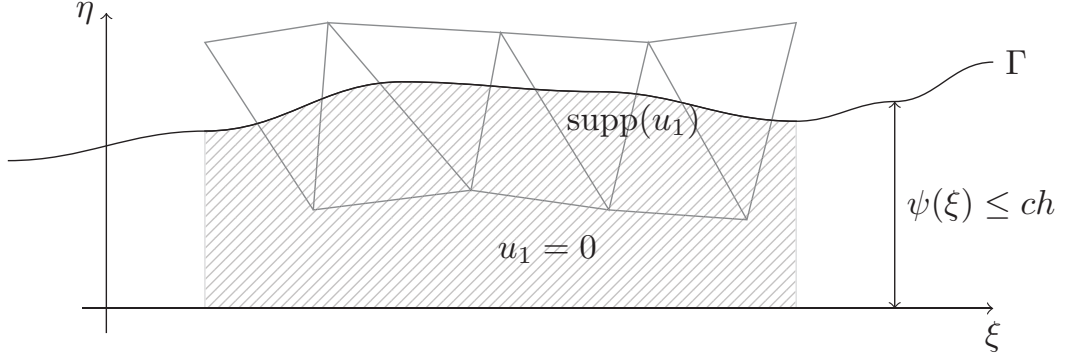
$$\|u_1\|_h^2 \simeq \|u_1\|_h^2 = |u_1|_{1, \Omega_1}^2 + \lambda \|\llbracket u_1 \rrbracket\|_{\frac{1}{2}, h, \Gamma}^2 \simeq |u_1|_{1, \Omega_1}^2 + h^{-1} \|u_1\|_{L^2(\Gamma)}^2 \quad (2.4.20)$$

This implies $|u_1|_{1, \Omega_1} \preceq \|u_1\|_h$. Next we show

$$h^{-1} \|u_1\|_{L^2(\Gamma)}^2 \preceq |u_1|_{1, \Omega_1}^2. \quad (2.4.21)$$

For this purpose we represent Γ locally as the graph of a function ψ , with a local coordinate system (ξ, η) as in figure 2.4.2. Then we can write

$$u_1(\xi, \psi(\xi)) = \underbrace{u_1(\xi, \psi(0))}_{=0} + \int_0^{\psi(\xi)} \frac{\partial u_1}{\partial \eta}(\xi, \eta) d\eta,$$


 Figure 2.4.2: Local representation of Γ as a graph.

and thus

$$u_1(\xi, \psi(\xi))^2 = \left| \int_0^{\psi(\xi)} \frac{\partial u_1}{\partial \eta}(\xi, \eta) d\eta \right|^2 \leq \underbrace{|\psi(\xi)|}_{\leq ch} \int_0^{\psi(\xi)} \left(\frac{\partial u_1}{\partial \eta}(\xi, \eta) \right)^2 d\eta.$$

Integration over ξ yields (2.4.21). In combination with (2.4.20) this yields $\|u_1\|_h^2 \preceq |u_1|_{1,\Omega_1}$, which completes the proof of (2.4.18). We now consider the result in (2.4.19). Due to $\|\cdot\|_h \simeq \|\cdot\|_h$ it suffices to prove

$$\|u_1\|_h^2 + \|u_2\|_h^2 \preceq \|u_1 + u_2\|_h^2 \quad \text{for all } u_1 + u_2 \in \mathcal{W}_1 \oplus \mathcal{W}_2. \quad (2.4.22)$$

The scalar product corresponding to $\|\cdot\|_h$ is denoted by $(\cdot, \cdot)_*$, i.e. $(u, v)_* = (u, v)_{1,\Omega_{1,2}} + \lambda(\llbracket u \rrbracket, \llbracket v \rrbracket)_{\frac{1}{2},h,\Gamma}$. From $(u_1, u_2)_{1,\Omega_{1,2}} = 0$ it follows that

$$|(u_1, u_2)_*| = |\lambda(\llbracket u \rrbracket, \llbracket v \rrbracket)_{\frac{1}{2},h,\Gamma}| \leq \lambda h^{-1} \|u_1\|_{L^2(\Gamma)} \|u_2\|_{L^2(\Gamma)}.$$

Using the results in (2.4.21), (2.4.18) we get, with a suitable constant c and for arbitrary $\delta \in (0, 1)$:

$$\begin{aligned} |(u_1, u_2)_*| &\leq (1 - \delta) \lambda h^{-1} \|u_1\|_{L^2(\Gamma)} \|u_2\|_{L^2(\Gamma)} + \delta c \lambda |u_1|_{1,\Omega_1} |u_2|_{1,\Omega_2} \\ &\leq \max\{1 - \delta, \delta c \lambda\} \|u_1\|_h \|u_2\|_h. \end{aligned}$$

By choosing a suitable δ , we obtain the strengthened Cauchy-Schwarz inequality

$$|(u_1, u_2)_*| \leq C^* \|u_1\|_h \|u_2\|_h \quad \text{for all } u_1 \in \mathcal{W}_1, u_2 \in \mathcal{W}_2,$$

with a constant $C^* < 1$, independent of h and of how the triangulation is intersected by Γ . This result is equivalent to the one in (2.4.22). $\square$

2.4.3.3 Stable subspace splitting of $\mathcal{S}$ into $\mathcal{W}_0$, $\mathcal{W}_1$ and $\mathcal{W}_2$

As a direct consequence of the stable splitting properties derived above we obtain the following main result.

Theorem 2.4.3.

Take $d = 2$. There exists a constant C_3 , independent of h and of how the triangulation is intersected by Γ , such that

$$\|u_0\|_h^2 + \|u_1\|_h^2 + \|u_2\|_h^2 \leq C_3 \|u_0 + u_1 + u_2\|_h^2 \quad \text{for all } u = u_0 + u_1 + u_2 \in \mathcal{S}.$$

Proof. Combine the result in (2.4.5) with the one in (2.4.19). $\square$

2.4.4 Optimal preconditioners based on approximate subspace corrections

The stable decomposition in the preceding section implies that a block diagonal matrix based on the subspaces V_h^s and V_h^x (or V_h^s and $V_{h,1}^x$ and $V_{h,2}^x$) with exact inverses gives $\mathcal{O}(1)$ spectral condition number bounds. However, In practice exact inverses are not feasible. In this subsection we describe how appropriate preconditioning of the matrix blocks results in suitable preconditioners for the full stiffness matrix.

For the subspaces V_h^s and V_h^x we consider symmetric positive definite preconditioners $\mathbf{C}_s$, $\mathbf{C}_x$ which provide the following bounds

$$\gamma_l \mathbf{u}_l^T \mathbf{C}_l \mathbf{u}_l \leq \mathbf{u}_l^T \mathbf{A}_l \mathbf{u}_l \leq \rho_l \mathbf{u}_l^T \mathbf{C}_l \mathbf{u}_l \quad \text{for all } \mathbf{u}_l \in \mathbb{R}^{N_l} \in V_h^l, \quad l \in \{s, x\} \quad (2.4.23)$$

with strictly positive constants $\gamma_l, \rho_l, l \in \{s, x\}$. The *additive subspace preconditioner* is defined by

$$\mathbf{B}^{-1} = \sum_{l \in \{s, x\}} \mathbf{E}_l^T \mathbf{C}_l^{-1} \mathbf{E}_l \in \mathbb{R}^{N \times N} \quad (2.4.24)$$

The theory presented in [Yse93] can be used to quantify the quality of the preconditioner $\mathbf{B}$.

Theorem 2.4.4.

Define $\gamma_{\min} = \min_l \gamma_l$, $\rho_{\max} = \max_l \rho_l$. Let C_2 be the constant of the stable splitting in theorem 2.4.1. The spectrum $\sigma(\mathbf{B}^{-1} \mathbf{A})$ is real and

$$\sigma(\mathbf{B}^{-1} \mathbf{A}) \subset \left[\frac{\gamma_{\min}}{C_2}, 2\rho_{\max} \right]$$

holds.

Proof. We recall a main result from [Yse93, Theorem 8.1] (in matrix notation). If there are strictly positive constants K_1, K_2 such that

$$K_1^{-1} \sum_{l \in \{s, x\}} \mathbf{u}^T \mathbf{E}_l^T \mathbf{C}_l \mathbf{E}_l \mathbf{u} \leq \mathbf{u}^T \mathbf{A} \mathbf{u} \leq K_2 \sum_{l \in \{s, x\}} \mathbf{u}^T \mathbf{E}_l^T \mathbf{C}_l \mathbf{E}_l \mathbf{u} \quad \forall \mathbf{u} \in \mathbb{R}^N$$

is satisfied, then $\sigma(\mathbf{B}^{-1}\mathbf{A}) \subset [K_1^{-1}, K_2]$ holds. For the lower inequality we use theorem 2.4.1 (in the form of (2.4.16)) and (2.4.23), which then results in

$$\mathbf{u}^T \mathbf{A} \mathbf{u} \geq C_2^{-1} \sum_{l \in \{s, x\}} \mathbf{u}^T \mathbf{E}_l^T \mathbf{A}_l \mathbf{E}_l \mathbf{u} \geq \frac{\gamma_{\min}}{C_2} \sum_{l \in \{s, x\}} \mathbf{u}^T \mathbf{E}_l^T \mathbf{C}_l \mathbf{E}_l \mathbf{u} \quad \forall \mathbf{u} \in \mathbb{R}^N. \quad (2.4.25)$$

For the upper bound we note

$$\mathbf{u}^T \mathbf{A} \mathbf{u} \leq 2 \sum_{l \in \{s, x\}} \mathbf{u}^T \mathbf{E}_l^T \mathbf{A}_l \mathbf{E}_l \mathbf{u} \leq 2\rho_{\max} \sum_{l \in \{s, x\}} \mathbf{u}^T \mathbf{E}_l^T \mathbf{C}_l \mathbf{E}_l \mathbf{u} \quad \forall \mathbf{u} \in \mathbb{R}^N. \quad (2.4.26)$$

Now we apply the above-mentioned result with $K_1 = C_2/\gamma_{\min}$ and $K_2 = 2\rho_{\max}$. $\square$

The result in theorem 2.4.1 yields that the constant C_2 is independent of h and of how the triangulation intersects the interface Γ .

Remark 2.4.4 (Block preconditioner with three blocks). *Instead of the splitting into the two subspaces one could consider the splitting into three subspaces following theorem 2.4.3. For an according block preconditioner $\mathbf{B}$ one would get $\sigma(\mathbf{B}^{-1}\mathbf{A}) \subset [\frac{\gamma_{\min}}{C_3}, 3\rho_{\max}]$ with γ_l and ρ_l the constants of according preconditioners for the corresponding blocks.*

A special case of theorem 2.4.23 is the case where exact subspace corrections are used, that means we set $\mathbf{B} = \mathbf{B}_A$, the corresponding additive subspace preconditioner has a condition number which is independent of h and the cut position (see remark 2.4.3). In practice exact solvers on the subspaces are in general not feasible and therefore replaced with suitable preconditioners $\mathbf{C}_l$ which provide appropriate (uniform) constants $\gamma_{\min}$ and $\rho_{\max}$.

We first consider the preconditioning of $\mathbf{A}_s$. $\mathbf{A}_s$ is a matrix stemming from a standard finite element discretization of a Poisson equation (with discontinuous diffusion coefficients). As a preconditioner $\mathbf{C}_s$ for $\mathbf{A}_s$ we can use a standard symmetric multigrid method (which is a multiplicative subspace correction method). From the literature [Hac03, Xu92, Yse93] we know that for this choice of $\mathbf{C}_s$ we have spectral inequalities as in (2.4.23), with $\rho_s = 1$ and a constant $\gamma_s > 0$ that is independent of h and of how Γ intersects the triangulation.

It remains to find an appropriate preconditioner $\mathbf{C}_x$ of $\mathbf{A}_x$. For this we propose the simple Jacobi method, i.e., diagonal scaling as a preconditioner for $\mathbf{A}_x$. This is discussed next.

2.4.5 Diagonal preconditioner on the XFEM subspace

We define the matrix $\mathbf{D}_x := \text{diag}(\mathbf{A}_x)$.

Lemma 2.4.5.

For the Jacobi preconditioner $\mathbf{D}_x$ there are strictly positive constants γ_x, ρ_x , independent of h and of how the triangulation is intersected by Γ such that

$$\gamma_x \mathbf{u}_x^T \mathbf{D}_x \mathbf{u}_x \leq \mathbf{u}_x^T \mathbf{A}_x \mathbf{u}_x \leq \rho_x \mathbf{u}_x^T \mathbf{D}_x \mathbf{u}_x \quad \text{for all } \mathbf{u}_x \in \mathbb{R}^{N_x} \quad (2.4.27)$$

holds.

Proof. We have $\mathbf{u}_x^T \mathbf{D}_x \mathbf{u}_x = \sum_{j \in \mathcal{J}_\Gamma} \mathbf{u}_{x,j} a_h(\varphi_j^\Gamma, \varphi_j^\Gamma)$. We split the set $\mathcal{J}_\Gamma$ into $\mathcal{J}_{\Gamma,i} := \{v \in \mathcal{J}_\Gamma, v \notin \Omega_i\}$, $i = 1, 2$. For each $T \in \mathcal{T}_h^\Gamma$ we define $T_i = T \cap \Omega_i$, and for each T_i we denote by $V(T_i)$ the set vertices of T that are *not* in Ω_i . Note that $V(T_i) \neq \emptyset$ and $V(T_i)$ does not contain all vertices of T due to assumption 2.4.2. Using (2.4.18) and the construction of the XFEM basis functions we get

$$\begin{aligned} \mathbf{u}_x^T \mathbf{D}_x \mathbf{u}_x &= \sum_{i=1,2} \sum_{j \in \mathcal{J}_{\Gamma,i}} \mathbf{u}_{x,j}^2 a_h(\varphi_j^\Gamma, \varphi_j^\Gamma) \simeq \sum_{i=1,2} \sum_{j \in \mathcal{J}_{\Gamma,i}} \mathbf{u}_{x,j}^2 |\varphi_j^\Gamma|_{1,\Omega_i}^2 \\ &= \sum_{i=1,2} \sum_{T \in \mathcal{T}_h^\Gamma} \sum_{j \in V(T_i)} \mathbf{u}_{x,j}^2 |\varphi_j^\Gamma|_{1,T_i}^2 = \sum_{i=1,2} \sum_{T \in \mathcal{T}_h^\Gamma} \sum_{j \in V(T_i)} \mathbf{u}_{x,j}^2 \|(\nabla \varphi_j)|_{T_i}\|_2^2 |T_i| \end{aligned} \quad (2.4.28)$$

with $\|\cdot\|_2$ the Euclidean vector norm, where in the last step we used the fact that $\nabla \varphi_j$ is a constant vector on each T_i . Using (2.4.18) and the fact that also ∇u is a constant vector on each T_i we get (with $u = \mathcal{G}\mathbf{u}$ where $\mathcal{G}$ is the Galerkin isomorphism on V_h^x)

$$\mathbf{u}_x^T \mathbf{A}_x \mathbf{u}_x = \|u\|_h^2 \simeq \sum_{i=1,2} \sum_{T \in \mathcal{T}_h^\Gamma} \|\nabla u\|_{L^2(T_i)}^2 = \sum_{i=1,2} \sum_{T \in \mathcal{T}_h^\Gamma} |T_i| \|(\nabla u)|_{T_i}\|_2^2. \quad (2.4.29)$$

Now note that $(\nabla u)|_{T_i} = \sum_{j \in V(T_i)} \mathbf{u}_{x,j} (\nabla \varphi_j^\Gamma)|_{T_i} = \sum_{j \in V(T_i)} \mathbf{u}_{x,j} (\nabla \varphi_j)|_T$. Because $V(T_i)$ does not contain all vertices of T , the vectors in the set $\{(\nabla \varphi_j)|_T \mid j \in V(T_i)\}$ are linearly independent and the angles between the vectors depend only on the geometry of the triangulation $\mathcal{T}_h$. This implies that

$$\|(\nabla u)|_{T_i}\|_2^2 \simeq \sum_{j \in V(T_i)} \mathbf{u}_{x,j}^2 \|(\nabla \varphi_j)|_T\|_2^2$$

Combining this with the results in (2.4.28) and (2.4.29) completes the proof. $\square$

We have thus found optimal preconditioners for $\mathbf{A}_s$ and $\mathbf{A}_x$. We introduce the *optimal* block preconditioner

$$\mathbf{B}_C := \begin{pmatrix} \mathbf{C}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_x \end{pmatrix} \quad (2.4.30)$$

which, due to theorem 2.4.4, has the property $\kappa(\mathbf{B}_C^{-1} \mathbf{A}) \simeq 1$.

2.4.6 Extension of results

We comment on several extensions of the previous results.

2.4.6.1 Jacobi preconditioner

Instead of an optimal multigrid preconditioner in the subspace $\mathcal{W}_0 = V_h^s$, one can also use a simpler (suboptimal) Jacobi preconditioner, i.e. $\mathbf{C}_s = \text{diag}(\mathbf{A}_s)$. For this choice the spectral constants in (2.4.23) are $\gamma_s \simeq h^2$ and $\rho_s \simeq 1$. The two subspaces are disjoint and thus if one applies a Jacobi preconditioner in the subspaces, the additive subspace preconditioner $\mathbf{C}$ in (2.4.24) coincides with a Jacobi preconditioner $\mathbf{D}_{\mathbf{A}} := \text{diag}(\mathbf{A})$. From theorem 2.4.4 we can conclude that $\kappa(\mathbf{D}_{\mathbf{A}}^{-1}\mathbf{A}) \leq ch^{-2}$ holds, with a constant c independent on h and the cut position. Similar uniform $\mathcal{O}(h^{-2})$ condition number bounds have recently been derived in the literature, cf. [ZWKB13] and [BZ12]. In these papers, however, for obtaining such a bound an additional (inconsistent) stabilization term is added to the bilinear form $a_h(\cdot, \cdot)$. Our analysis shows that although the condition number of the stiffness matrix corresponding to $a_h(\cdot, \cdot)$ does not have a uniform (w.r.t. the interface cut) bound ch^{-2} , a simple diagonal scaling results in a matrix with a spectral condition number that is bounded by ch^{-2} , with a constant c that is independent of how Γ is intersected by the triangulation. We note that adding a stabilization as treated [BZ12] may have a positive effect not only on the condition number, but also on robustness of the discretization with respect to large jumps in the diffusion coefficient.

2.4.6.2 Parameter dependency

We briefly discuss the dependency of the quality of the preconditioners on the parameters α_i , $i = 1, 2$ and λ . In this remark c is a generic constant independent of h , λ , α_i , $i = 1, 2$ and the cut position. There are essentially three places where the parameter dependency plays a role. First, we discuss the dependency of the splitting into non-overlapping subspaces (matrix blocks). Second, we are interested in the spectral bounds for the Jacobi preconditioner $\mathbf{C}_x$ for $\mathbf{A}_x$ in (2.4.27). Third, we need spectral bounds for the multigrid preconditioner $\mathbf{C}_s$ for $\mathbf{A}_s$ in (2.4.23).

The first dependency is characterized by C_2 in theorem 2.4.1. Note that throughout this section we used assumption 2.4.1. In a more detailed analysis we have to replace estimates of the form

$$\|u\|_h \simeq \|u\|_h$$

with

$$\sqrt{\alpha_{\min}} \|u\|_h \leq \|u\|_h \leq \sqrt{\alpha_{\max}} \|u\|_h.$$

This yields $C_2 \leq c \frac{\alpha_{\max}}{\alpha_{\min}}$. The spectral condition number bound for the block diagonal with exact inverses is directly related (see remark 2.4.3) to that constant C_2 and we get

estimate $\kappa(\mathbf{B}_\mathbf{A}^{-1}\mathbf{A}) \leq c \frac{\alpha_{\max}}{\alpha_{\min}}$. Numerical experiments in section 2.5.3.6 indicate that this estimate is sharp. Note, that C_2 is bounded independent of λ .

A similar refinement of the analysis in lemma 2.4.5, especially w.r.t. to the parameter-dependency of (2.4.18) reveals that we have $\rho_x \simeq 1$ and $\gamma_x \simeq (\frac{\alpha_{\max}}{\alpha_{\min}}\lambda)^{-1}$. To investigate this numerically we introduce the preconditioner

$$\mathbf{B}_\mathbf{D} := \begin{pmatrix} \mathbf{A}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_x \end{pmatrix} \quad (2.4.31)$$

which thus has the condition number bound $\kappa(\mathbf{B}_\mathbf{D}^{-1}\mathbf{A}) \leq c(\frac{\alpha_{\max}}{\alpha_{\min}})^2\lambda$. In section 2.5.3.6 we observe in numerical experiments that the linear dependency on λ seems to be sharp, but the dependency on $\frac{\alpha_{\max}}{\alpha_{\min}}$ seems to be linear instead of quadratic.

For the multigrid preconditioner $\mathbf{C}_s$ we expect bounds of the form $\rho_s = 1$ and $\gamma_s \geq c(\frac{\alpha_{\max}}{\alpha_{\min}})^{-1}$ which results in the estimate $\kappa(\mathbf{B}_\mathbf{C}^{-1}\mathbf{A}) \leq c(\frac{\alpha_{\max}}{\alpha_{\min}})^2\lambda$ for the optimal preconditioner in (2.4.30). For the Jacobi preconditioner of the standard finite element part $\mathbf{D}_s$ we expect $\rho_s \leq c$ and $\gamma_s \geq c(\frac{\alpha_{\max}}{\alpha_{\min}})^{-1}h^2$ which then gives $\kappa(\mathbf{D}_\mathbf{A}\mathbf{A}) \leq c(\frac{\alpha_{\max}}{\alpha_{\min}})^2\lambda h^{-2}$.

Note that the dependency on λ appears only due to the Jacobi preconditioning of the XFEM block. To get condition numbers independent of λ one could consider using exact solves with $\mathbf{A}_x$ in the block preconditioner. This however can only be computational feasible if the number of degrees of freedoms in V_h^x is small.

Remark 2.4.5 (Iteration numbers of preconditioned CG method). *The performance of a preconditioned CG method is not only determined by the minimum and the maximum eigenvalue. The distribution of the eigenvalues of the preconditioned matrix is important. It can happen that only a few eigenvalues are close to the smallest (or largest) eigenvalue. In that case the iteration number can be significantly smaller than a prediction based only on the spectral condition number. In the numerical examples in section 2.5.3 we observe such an effect where estimates for the iteration number based only on the condition number are not sharp.*

2.4.6.3 Time dependent problems

For simple time stepping schemes extensions of the analysis can be easily derived. Consider e.g. the implicit Euler method as discussed in section 2.2.6.1 (with $\theta = 1$). The system matrix one has to solve for in every time step is $\mathbf{S} = \mathbf{M} + \Delta t\mathbf{A}$ with $\mathbf{M}$ the mass matrix. For the mass matrix problem we have $\mathbf{M} \simeq \mathbf{D}_\mathbf{M} = \text{diag}(\mathbf{M})$ as has been shown in [Reu08]. Thus the estimates in theorem 2.4.1 and lemma 2.4.5 also extend to norms corresponding to $\mathbf{S}$ and an optimal preconditioner is also obtained by a combination of multigrid (on V_h^s) and Jacobi preconditioner (on V_h^x). We briefly discuss the case of a simple Jacobi preconditioner. We have with $\mathbf{D}_\mathbf{S} = \text{diag}(\mathbf{S})$

$$\mathbf{u}^T \mathbf{S} \mathbf{u} \preceq \mathbf{u}^T \mathbf{D}_\mathbf{S} \mathbf{u} = \mathbf{u}^T (\mathbf{D}_\mathbf{M} + \Delta t \mathbf{D}_\mathbf{A}) \mathbf{u} \preceq \mathbf{u}^T (\mathbf{M} + \Delta t h^{-2} \mathbf{A}) \mathbf{u} \preceq (1 + \Delta t h^{-2}) \mathbf{u}^T \mathbf{S} \mathbf{u}$$

which implies $\kappa(\mathbf{D}_\mathbf{S}^{-1}\mathbf{S}) \preceq 1 + \Delta t h^{-2}$.

2.5 Numerical examples

In this section we want to investigate the practical behavior of the Nitsche-XFEM and the SD-Nitsche-XFEM method for a set of test problems. In section 2.5.1 we consider a simple stationary two-dimensional configuration with a circular inner domain and no convection ($\mathbf{w} = 0$). We investigate the approximation quality of the introduced extended finite element space and compare it to standard finite element spaces. We further compare the discretization error obtained by the Nitsche-XFEM method and a standard Galerkin method.

The experiments are followed by a geometrically more challenging example in section 2.5.2. Here, the inner domain has the shape of a “starfish”. Furthermore in that example slightly more general interface conditions are introduced. The convergence of errors in all norm that are relevant for the error analysis are investigated.

The purpose of the numerical examples in section 2.5.3 is the investigation of the conditioning of the linear systems arising from the Nitsche-XFEM discretization and the performance of the preconditioners presented in section 2.4. The dependency of the linear systems on h , λ , α , β and the interface position is investigated.

In all the three sections only the stationary case without convection is considered. In the last two examples we consider the SD-Nitsche-XFEM method for convection-dominated cases.

In section 2.5.4 the SD-Nitsche-XFEM method is considered for a two-dimensional stationary problem with a straight interface and very high mesh Péclet numbers. Discretization errors for a problem with a smooth solution and a problem with sharp (parabolic) boundary layers are investigated and compared to the unstabilized Nitsche-XFEM method. Further the impact of different scalings for the Nitsche stabilization parameter is considered.

The last examples in section 2.5.5 concludes the numerical examples. The problem is three-dimensional and non-stationary. A comparison between the stabilized and unstabilized Nitsche-XFEM method with different scalings of the stabilization parameter λ is carried out.

Remark 2.5.1 (Approximation of the domains). *As noted in remark 2.2.1 we so far assumed that we can integrate on T_i and Γ_T exactly on every element $T \in \mathcal{T}_h$. This is typically not true. We approximate the interface Γ by Γ_h a piecewise planar approximation of the interface Γ . Details on the numerical integration are addressed in chapter 4. For the numerical examples in this section errors due to numerical integration are negligibly small if not addressed otherwise.*

2.5.1 Elliptic interface problem: The disk problem

As a first example we consider a rather simple geometrical configuration: a circle inside a square. The problem that we consider is the simplest version of the mass transport problem, we consider problem 2.1.1. In this example we want to investigate and compare the discretization and approximation errors of standard finite element spaces and the XFEM space introduced in section 2.2.1.

2.5.1.1 Problem description

The outer domain is $\Omega = [-1, 1]^2$ and the interface is given by

$$\Gamma := \{(x, y) : x^2 + y^2 = R^2\} \quad \text{with } R = 0.3.$$

Γ is the zero level of the level set function $\varphi(\mathbf{x}) = \sqrt{x^2 + y^2} - R$. The inner domain is $\Omega_1 := \{\mathbf{x} \in \Omega, \phi(\mathbf{x}) < 0\}$. We prescribe the right hand side source term f and the Dirichlet boundary conditions so that the solution is given similar to “Case 2” in [Ngu09]:

$$u(\mathbf{x}) = \begin{cases} \alpha_2 U(r(\mathbf{x})) + \beta_2, & \mathbf{x} \in \Omega_1 \\ \alpha_1 U(r(\mathbf{x})) + \beta_1, & \mathbf{x} \in \Omega_2 \end{cases} \quad \text{with} \quad U(r) = r^2 - R^2 \text{ and } r(\mathbf{x}) = \sqrt{x^2 + y^2}$$

with $(\alpha_1, \alpha_2) = (1, 5)$ and $(\beta_1, \beta_2) = (2, 1)$. As we are primarily interested in the approximation properties of finite element spaces we also consider the transformed problem without a discontinuity:

$$\beta_i \rightarrow 1, \alpha_i \rightarrow \frac{\alpha_i}{\beta_i}$$

with the transformed solution

$$\tilde{u}(\mathbf{x}) = \begin{cases} \alpha_2 \beta_1 U(r(\mathbf{x})) + \beta_1 \beta_2, & \mathbf{x} \in \Omega_1 \\ \alpha_1 \beta_2 U(r(\mathbf{x})) + \beta_1 \beta_2, & \mathbf{x} \in \Omega_2 \end{cases}$$

which still has a kink at the interface. Note that approximating $\tilde{u}$ in $V_h \subset H_0^1(\Omega)$ is equivalent to approximating u in $\beta^{-1}V_h \subset H_{0,\beta}^1(\Omega)$.

2.5.1.2 Approximation Errors

We consider the best approximation to u and $\tilde{u}$ in the norm $\|\cdot\|_0$ for the standard finite element space V_h on seven successively (uniformly) refined meshes. The coarsest mesh (L1) is a triangular mesh obtained by dividing each square of a uniform 8×8 mesh into two triangles.

In table 2.5.1 we observe that the approximation of u (which has a discontinuity) with the standard finite element space V_h leads to a very slow convergence with order ≈ 0.5 .

Level	$\inf_{v_h \in V_h} \ u - v_h\ _0$ (eoc)	$\inf_{v_h \in \beta^{-1}V_h} \ u - v_h\ _0$ (eoc)	$\inf_{v_h \in V_h^\Gamma} \ u - v_h\ _0$ (eoc)
L1	3.86×10^{-1} (-)	6.72×10^{-2} (-)	3.73×10^{-2} (-)
L2	2.33×10^{-1} (0.73)	2.46×10^{-2} (1.45)	1.02×10^{-2} (1.88)
L3	1.74×10^{-1} (0.42)	8.08×10^{-3} (1.61)	2.51×10^{-3} (2.02)
L4	1.17×10^{-1} (0.58)	2.96×10^{-3} (1.45)	6.43×10^{-4} (1.97)
L5	8.84×10^{-2} (0.40)	8.76×10^{-4} (1.76)	1.62×10^{-4} (1.99)
L6	6.03×10^{-2} (0.55)	3.35×10^{-4} (1.39)	4.06×10^{-5} (2.00)
L7	4.37×10^{-2} (0.47)	1.20×10^{-4} (1.49)	1.02×10^{-5} (2.00)

Table 2.5.1: Approximation errors for finite element spaces V_h , $\beta^{-1}V_h$ and V_h^Γ under mesh refinement.

For the approximation of u in $\beta^{-1}V_h$ (which is equivalent to approximating $\tilde{u}$ in V_h) we observe that the convergence is of order ≈ 1.5 . If we check for the best approximation to u in V_h^Γ the convergence order is ≈ 2 . The XFEM enrichment thus cures the problem of the standard finite element space with respect to approximating discontinuities (across an interface). The results are consistent with the considerations in section 2.2.1.

2.5.1.3 Discretization Errors

Instead of the approximation error we now want to consider the discretization error and in addition to the $\|\cdot\|_0$ -norm we consider the $|\cdot|_{1,\Omega_{1,2}}$ -semi-norm. We consider the discretization with the standard and XFEM finite element space and introduce the according Galerkin projections. In order to minimize the effect of discontinuous parameters (and the corresponding impact on the approximation error) for the standard finite element space we transform the continuous problem to a problem with a kink only, discretize this problem using the standard Galerkin method and transform the discrete solution back. The overall discrete solution is then in $\beta^{-1}V_h$ where $\beta^{-1}V_h \subset H_{0,\beta}^1(\Omega)$ but $\beta^{-1}V_h \not\subset V_h$.

$$G^x : H_{0,\beta}^1(\Omega) \cap H^2(\Omega_{1,2}) \rightarrow V_h^\Gamma, \text{ s.t. } a_h(G^x u, v_h) = a_h(u, v_h) \quad \forall v_h \in V_h^\Gamma \quad (2.5.1)$$

$$G : H_{0,\beta}^1(\Omega) \rightarrow \beta^{-1}V_h, \text{ s.t. } \tilde{a}(G(\beta u), v_h) = \tilde{a}((\beta u), v_h) \quad \forall v_h \in V_h \quad (2.5.2)$$

with $\tilde{a}(u, v) := (\frac{\alpha}{\beta} \nabla u, \nabla v)_{L^2(\Omega_{1,2})}$, $u, v \in H^1(\Omega)$ where in $a_h(\cdot, \cdot)$ we set $\lambda = 20$.

In table 2.5.2 the results are shown on seven subsequently (uniformly) refined meshes. For the discretization with standard finite elements we observe an $\mathcal{O}(\sqrt{h})$ -convergence in the $|\cdot|_{1,\Omega_{1,2}}$ -norm. For the $\|\cdot\|_0$ -norm we observe an $\mathcal{O}(h)$ -convergence which is half an order suboptimal w.r.t. the best approximation in $\beta^{-1}V_h$. For the discretization with Nitsche-XFEM we observe optimal order convergence rates, i.e. a second order convergence behavior in the norm $\|\cdot\|_0$ and a first order convergence in the $|\cdot|_{1,\Omega_{1,2}}$ -norm which is agreement with the a priori error analysis in section 2.3.1.

Level	$\ u - Gu\ _0$ (eoc)	$ u - Gu _{1,\Omega_{1,2}}$ (eoc)	$\ u - G^x u\ _0$ (eoc)	$ u - G^x u _{1,\Omega_{1,2}}$ (eoc)
L1	1.85×10^{-1} (-)	1.52×10^0 (-)	7.46×10^{-2} (-)	8.49×10^{-1} (-)
L2	1.13×10^{-1} (0.71)	1.31×10^0 (0.22)	1.95×10^{-2} (1.94)	4.53×10^{-1} (0.91)
L3	7.53×10^{-2} (0.59)	1.09×10^0 (0.25)	5.71×10^{-3} (1.77)	2.35×10^{-1} (0.94)
L4	3.49×10^{-2} (1.11)	7.60×10^{-1} (0.53)	1.33×10^{-3} (2.10)	1.17×10^{-1} (1.01)
L5	1.51×10^{-2} (1.21)	4.93×10^{-1} (0.63)	3.34×10^{-4} (1.99)	5.89×10^{-2} (0.99)
L6	8.05×10^{-3} (0.91)	3.64×10^{-1} (0.44)	9.07×10^{-5} (1.88)	2.95×10^{-2} (1.00)
L7	4.38×10^{-3} (0.88)	2.69×10^{-1} (0.44)	2.19×10^{-5} (2.05)	1.47×10^{-2} (1.00)

Table 2.5.2: Convergence of standard finite element and Nitsche-XFEM discretizations on successively refined meshes for the example in section 2.5.1. Note that the Galerkin projection G in (2.5.2) maps into $\beta^{-1}V_h$.

2.5.1.4 Higher order discretization

We repeat the experiment for the higher order discretization with polynomial degree $k = 2$ and $k = 3$. To this end, we change the setup and set $U(r) = \cos(\frac{\pi r^2}{2R^2})$ such that the solution is not in V_h^Γ . We discuss the Nitsche stabilization parameter λ and the weights κ_i used in the averaging $\{\!\!\{\}$, the interface approximation and linear systems in this example.

Remark 2.5.2 (Weighting of the average (κ) and stabilization parameter (λ)). *For $k = 2$ and $k = 3$ we choose the weighting $\kappa_i \in \{0, 1\}$ discussed in remark 2.3.1. The stabilization parameter λ should scale with k^2 with the polynomial order k due to the inverse estimate in (2.3.10) (cf. also [Mas12, lemma 3.5]). We choose $\lambda = 2k(k + 1)$ such that we have $\lambda = 12$ for $k = 2$ and $\lambda = 24$ for $k = 3$. Further numerical experiments showed that the averaging choice $\kappa = \kappa^{\text{HANSBO}}$ gives similar results for this example.*

Remark 2.5.3 (Interface approximation). *The piecewise planar approximation Γ_h of Γ that we use (see chapter 4 for details) gives only a second order approximation. However the geometrical approximation can be refined by additional adaptive refinements that are applied for quadrature purposes only. In this example we apply 8 additional adaptive refinements to drive the quadrature error negligibly small, i.e. the quadrature error is $\leq c \cdot 2^{-16} \cdot h^2$ for a constant $C \sim \mathcal{O}(1)$ which is sufficiently small within the considered range for h .*

Remark 2.5.4 (Linear solver). *While for the piecewise linear discretization preconditioning is fairly well understood and does not pose a problem (see theory in section 2.4 and numerical experiments in section 2.5.3), for higher order polynomials conditioning (even after diagonal scaling) without additional stabilization (e.g. ghost penalty) can get arbitrarily bad. For the current example we used direct solvers. We did not investigate the problem further.*

In table 2.5.3 and table 2.5.4 we observe the results. The errors in the domains, i.e. in the norms $\|\cdot\|_0$ and $\|\cdot\|_{1,\Omega_{1,2}}$ behave as predicted in the error analysis (section 2.3.1) while the interface errors are one half order better than in the estimates.

Level	$ e_h _{1,\Omega_{1,2}}$ (eoc)	$\ e_h\ _0$ (eoc)	$\ \llbracket \beta u_h \rrbracket\ _\Gamma$ (eoc)	$\ \llbracket \alpha \nabla e_h \cdot \mathbf{n} \rrbracket\ _\Gamma$ (eoc)
L1	1.48×10^1 (-)	6.27×10^{-1} (-)	7.84×10^{-2} (-)	3.85×10^0 (-)
L2	6.49×10^0 (1.19)	1.25×10^{-1} (2.33)	2.87×10^{-2} (1.45)	4.25×10^0 (-0.14)
L3	2.08×10^0 (1.64)	1.17×10^{-2} (3.42)	8.36×10^{-3} (1.78)	1.97×10^0 (1.11)
L4	5.66×10^{-1} (1.88)	9.20×10^{-4} (3.66)	1.36×10^{-3} (2.63)	5.52×10^{-1} (1.84)
L5	1.46×10^{-1} (1.95)	7.74×10^{-5} (3.57)	1.91×10^{-4} (2.83)	1.44×10^{-1} (1.93)
L6	3.71×10^{-2} (1.98)	6.77×10^{-6} (3.51)	2.36×10^{-5} (3.02)	3.60×10^{-2} (2.00)

 Table 2.5.3: Convergence under successively uniform mesh refinement of different measures of error $e_h := u - u_h$ for the circle example ($k = 2$).

Level	$ e_h _{1,\Omega_{1,2}}$ (eoc)	$\ e_h\ _0$ (eoc)	$\ \llbracket \beta u_h \rrbracket\ _\Gamma$ (eoc)	$\ \llbracket \alpha \nabla e_h \cdot \mathbf{n} \rrbracket\ _\Gamma$ (eoc)
L1	8.26×10^0 (-)	4.43×10^{-1} (-)	3.53×10^{-2} (-)	4.49×10^0 (-)
L2	1.52×10^0 (2.45)	7.93×10^{-2} (2.48)	1.14×10^{-2} (1.63)	1.75×10^0 (1.36)
L3	2.24×10^{-1} (2.76)	5.08×10^{-3} (3.97)	1.11×10^{-3} (3.36)	3.16×10^{-1} (2.46)
L4	2.55×10^{-2} (3.13)	3.39×10^{-4} (3.90)	7.02×10^{-5} (3.99)	4.14×10^{-2} (2.93)
L5	2.97×10^{-3} (3.10)	2.15×10^{-5} (3.98)	4.28×10^{-6} (4.04)	5.20×10^{-3} (2.99)
L6	3.59×10^{-4} (3.05)	1.35×10^{-6} (3.99)	2.40×10^{-7} (4.16)	6.35×10^{-4} (3.03)

 Table 2.5.4: Convergence under successively uniform mesh refinement of different measures of error $e_h := u - u_h$ for the circle example ($k = 3$).

2.5.2 Elliptic interface problem: The starfish problem

This example is geometrically more challenging. The geometrical configuration of this example is taken from [Li95]. Due to its shape we call it the “starfish” example. The outer domain is $\Omega = [-1, 1]^2$ and the interface is given as

$$\Gamma := \{(x, y) : x = R(\theta) \cos(\theta), y = R(\theta) \sin(\theta), \theta \in [0, 2\pi]\}$$

with

$$R(\theta) = r_0 + 0.2 \sin(\omega\theta), \quad r_0 = 0.5, \quad \omega = 5.$$

Γ can also be defined as the zero level of the level set function $\phi(\mathbf{x}) = r(\mathbf{x}) - R(\theta)$ with $r(\mathbf{x}) = \sqrt{x^2 + y^2}$ and $\theta(\mathbf{x}) = \arctan(x/y)$. The inner domain is $\Omega_1 := \{\mathbf{x} \in \Omega, \phi(\mathbf{x}) < 0\}$. A sketch of the geometrical configuration is displayed in figure 2.5.1 (left).

We consider a generalization of problem 2.1.3 with respect to the interface conditions which allows non-homogeneous right hand side terms for the flux and jump condition at the interface.

Problem 2.5.1.

$$-\operatorname{div}(\alpha \nabla u) = f \quad \text{in } \Omega_i, \quad i = 1, 2, \quad (2.5.3a)$$

$$\llbracket \alpha \nabla u \cdot \mathbf{n} \rrbracket_\Gamma = g_\alpha \quad \text{on } \Gamma, \quad (2.5.3b)$$

$$\llbracket \beta u \rrbracket_\Gamma = g_\beta \quad \text{on } \Gamma, \quad (2.5.3c)$$

$$u = g_D \quad \text{on } \partial\Omega. \quad (2.5.3d)$$

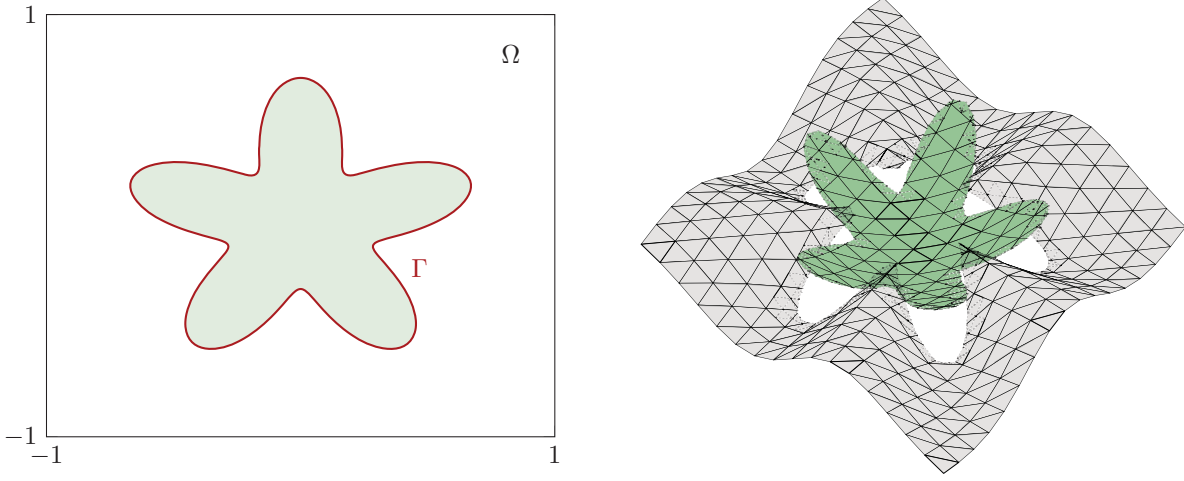


Figure 2.5.1: Sketch (left) of the setup of the numerical example from section 2.5.2 and elevation of a discrete solution (right).

Due to the different interface conditions, the discretization is modified by a change in the discrete linear form f . The bilinear form $a_h(\cdot, \cdot)$ is unchanged.

$$f(v) = (f, v)_0 + (g_\beta, \llbracket \alpha \nabla v \cdot \mathbf{n} \rrbracket) + \bar{\alpha} \frac{\lambda}{h} \llbracket \beta v \rrbracket_\Gamma + (g_\alpha, \llbracket \beta v \rrbracket)_\Gamma, \quad v \in V_h^\Gamma$$

We prescribe f , g_α , g_β and the Dirichlet data g_D so that the solution is

$$u(\mathbf{x}) = \begin{cases} C_1 \sin(\omega \theta(\mathbf{x}) + \pi), & \mathbf{x} \in \Omega_1 \\ C_2 r^2(\mathbf{x}), & \mathbf{x} \in \Omega_2 \end{cases},$$

with $C_1 = 1$ and $C_2 = 4$. The parameters are set to $(\alpha_1, \alpha_2) = (2, 1)$, $(\beta_1, \beta_2) = (1, 3/2)$ and $\lambda = 4$. Starting on a 8×8 mesh we successively refine (uniformly) and measure the errors w.r.t. different norm contributions.

Level	$ e_h _{1, \Omega_{1,2}}$ (eoc)	$\ e_h\ _0$ (eoc)	$\ \llbracket \beta e_h \rrbracket\ _\Gamma$ (eoc)	$\ \llbracket \alpha \nabla e_h \cdot \mathbf{n} \rrbracket\ _\Gamma$ (eoc)
L1	6.76×10^0 (-)	6.75×10^{-1} (-)	4.78×10^{-1} (-)	5.64×10^0 (-)
L2	3.75×10^0 (0.85)	1.90×10^{-1} (1.83)	1.61×10^{-1} (1.57)	4.63×10^0 (0.28)
L3	2.06×10^0 (0.86)	5.54×10^{-2} (1.78)	4.10×10^{-2} (1.98)	2.46×10^0 (0.91)
L4	1.07×10^0 (0.94)	1.63×10^{-2} (1.77)	1.23×10^{-2} (1.74)	1.49×10^0 (0.72)
L5	5.45×10^{-1} (0.98)	4.10×10^{-3} (1.99)	3.05×10^{-3} (2.01)	6.64×10^{-1} (1.16)
L6	2.75×10^{-1} (0.99)	1.03×10^{-3} (1.99)	7.10×10^{-4} (2.10)	3.39×10^{-1} (0.97)
L7	1.38×10^{-1} (1.00)	2.59×10^{-4} (1.99)	1.86×10^{-4} (1.93)	1.79×10^{-1} (0.92)

Table 2.5.5: Convergence under successively uniform mesh refinement of different measures of error $e_h := u - u_h$ for the starfish example ($k = 1$).

In table 2.5.5 the results for piecewise linear functions ($k = 1$) are displayed. Further in figure 2.5.1 (right) the elevation of the solution on mesh level L2 is shown.

We observe a similar behavior as in the example in section 2.5.1: We get optimal convergence rates in the norms $\|\cdot\|_0$ and $\|\cdot\|_{1,\Omega_{1,2}}$. The errors $\|\llbracket \beta u_h \rrbracket\|_\Gamma$ and $\|\llbracket \alpha \nabla e_h \cdot \mathbf{n} \rrbracket\|_\Gamma$ with $e_h := u - u_h$ converge half an order faster than predicted in the error estimates.

2.5.3 Elliptic interface problem: Conditioning

The aim of this example is to investigate the conditioning of linear systems arising from the Nitsche-XFEM discretization and the performance of iterative solvers and preconditioners for those systems especially with respect to the estimates in section 2.4. This example has also been discussed in [LR14]. We recall that $\mathbf{A}$ denotes the system matrix obtained *after* the basis transformation, cf. section 2.4.1 and the system matrix *without* transformation is denoted by $\hat{\mathbf{A}}$.

In section 2.5.3.1 we introduce the basic setup of this example. To examine the robustness of linear solvers against different cut positions, we consider different positions for the interface Γ and simple Jacobi preconditioning. In section 2.5.3.2 this is done for an unstructured mesh by changing the position of the interface along the x -axis. In that case it can only happen that a single or a small number of elements have “small cuts”. A more challenging situation is constructed using a structured mesh in section 2.5.3.3 where almost all cut elements have “small cuts”. Based on this extreme case we also examine the performance of the block-preconditioning (additive subspace preconditioning) of the matrix $\mathbf{A}$. The blocks are related to the finite element spaces V_h and V_h^x . In section 2.5.3.4, section 2.5.3.5 and section 2.5.3.6 parameter studies are presented to check for the dependency of the conditioning on the mesh size h (section 2.5.3.4), the Nitsche stabilization parameter λ (section 2.5.3.5), and the problem parameters α and β (section 2.5.3.6). In section 2.5.3.7 the optimal preconditioner based on a multigrid solver for the V_h -block for a three dimensional test case analogously to that in section 2.5.3.1 is tested. A summary of the results is given in section 2.5.3.8.

2.5.3.1 Basic setup

We restrict to the symmetric problem and thus consider the stationary diffusion problem, problem 2.1.3. The domain is the unit square $\Omega = [0, 1]^2$ with an interface Γ which is a square with corners that are rounded off. A sketch including proper dimensioning is displayed in figure 2.5.2. The rounded square is centered around $\mathbf{x}_0 = (x_0, y_0)$, it is denoted as Ω_1 . We set the dimensions to $d = 0.2$ and $r = 0.05$. To investigate conditioning of the system, especially with respect to the dependency on the interface position, we consider different positions for (x_0, y_0) . The Henry and diffusion parameters are also fixed to $(\alpha_1, \alpha_2) = (3, 2)$ and $(\beta_1, \beta_2) = (2, 1)$. The Nitsche stabilization parameter is set to $\lambda = 4$. As a right hand side source term we choose $f = 1$ in Ω_1 and $f = 0$ in Ω_2 . We consider two different types of triangulations, an unstructured mesh and a structured mesh, both with approximately the same number of elements. Further we consider four levels of refinements for each triangulation denoted as L1,...,L4.

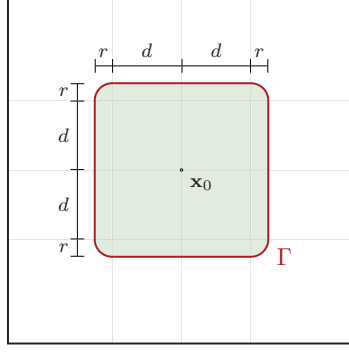


Figure 2.5.2: Sketch of the setup of the numerical example from section 2.5.3

For the discretization we consider the method as in (2.2.14), where for the weighting in the averaging $\{\!\!\{\}$ we consider two choices: the “naive” choice $\kappa_i = 0.5$, $i = 1, 2$ and the HANSBO weighting $\kappa = \kappa^H$. We consider the linear systems after the transformation proposed in section 2.4.1 and the linear systems without transformation. As default we consider the transformed case.

As quantities of interest we consider the condition number of the system matrix $\mathbf{A}$ (and $\hat{\mathbf{A}}$), the condition number of the preconditioned matrix and the number of iterations a preconditioned CG solve takes to reduce the initial residual by a factor of 10^{-6} .

2.5.3.2 Dependency on the cut position on an unstructured grid

By varying x_0 , the center of Ω_i , in the range $[0.46, 0.54]$ we want to examine how the properties of the linear systems depend on the cut configuration. We keep $y_0 = 0.5$ fixed.

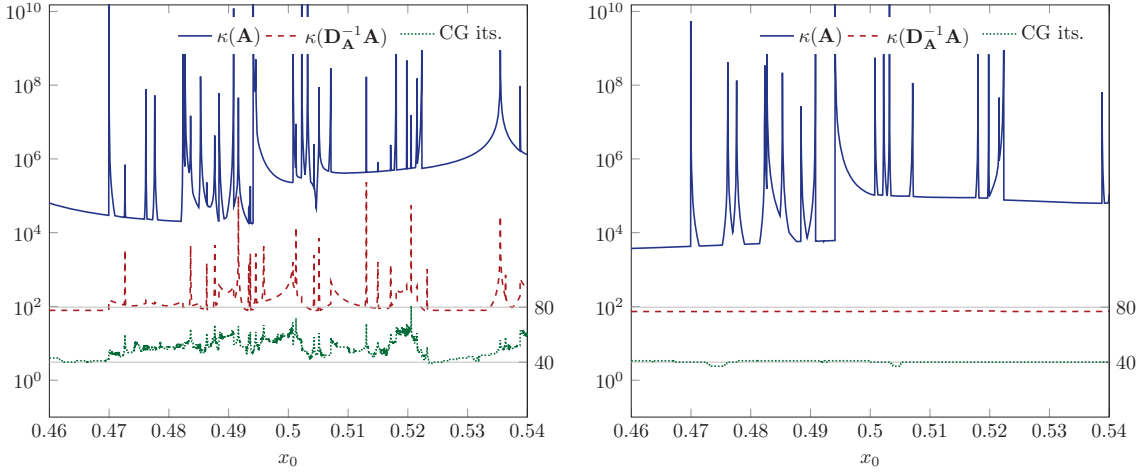


Figure 2.5.3: Condition of the linear systems arising for the test case in section 2.5.3.2 for different interface positions x_0 and different discretizations: using the “naive” averaging $\kappa_i = \frac{1}{2}$ (left) and using the HANSBO averaging (right).

$\kappa(\mathbf{D}_{\mathbf{A}}^{-1}\mathbf{A})$ (its.)		$\kappa_{1/2} = 0.5$ (transf.)	$\kappa = \kappa^H$ (transf.)	$\kappa = \kappa^H$ (untransf.)	Std. FEM (transf.)
L1	max	' ∞ '(44)	19.2(23)	32.7(30)	14.8(19)
	min	61.7(23)	18.6(20)	23.4(25)	14.6(17)
	median	63.7(26)	18.7(22)	27.9(28)	14.7(19)
	avg.	' ∞ '(28.2)	18.8(21.7)	27.8(28)	14.7(19)
L2	max	' ∞ '(81)	76.5(41)	107(53)	62.3(39)
	min	78.1(39)	72.6(37)	87.8(45)	61.9(34)
	median	107(49)	73.0(40)	94.0(51)	62.0(37)
	avg.	' ∞ '(49.5)	73.3(40.2)	95.0(50.3)	62.0(37.4)
L3	max	' ∞ '(257)	308(84)	374(96)	267(81)
	min	492(89)	287(74)	337(82)	266(71)
	median	476736(120)	295(81)	349(91)	266(77)
	avg.	' ∞ '(127)	296(80.3)	349(91.4)	266(76)
L4	max	()	1310(174)	1390(186)	1120(168)
	min	()	1210(153)	1300(160)	1120(147)
	median	()	1220(165)	1320(177)	1120(158)
	avg.	()	1220(165)	1320(178)	1120(159)

Table 2.5.6: Dependency of conditioning on cut position and mesh size.

In figure 2.5.3 the main characteristics of the transformed linear systems are shown for the unstructured grid on level L2 for $x_0 \in \{0.46 + n \cdot 2 \cdot 10^{-5}, n \in \mathbb{N}_0, n \leq 8000\}$. The results, discussed in the following, look very similar if one considers the untransformed formulation instead of the transformed one. The mesh is shown on the left hand side of figure 2.5.4. A similar test case to investigate the dependency of the condition number of the system matrix on the interface position has also been considered in [FR14] for a different finite element method approach.

For both discretizations, using the “naive” and the HANSBO weighting, one observes that the condition number of $\mathbf{A}$ has several blow-ups, although some blow-ups seem to exist only for the “naive” weighting. The locations of the blow-ups correspond to situations where ν (cf. (2.4.1)) tends to zero. While this behavior is qualitatively the same for the Jacobi-preconditioned matrix $\mathbf{D}_{\mathbf{A}}^{-1}\mathbf{A}$ when the “naive” weighting is applied, the situation is completely different if the HANSBO weighting is applied. Here the condition number of the Jacobi-preconditioned matrix is essentially independent of the cut configuration which is in agreement with the theoretical results of section 2.4. This directly reflects in the number of CG iterations that are necessary to solve the linear systems. While the iteration count depends significantly on the cut configuration for the “naive” weighting, the iteration counts for the HANSBO weighting are bounded independent of the cut configuration. We considered the same procedure on one coarser level (L1) and two finer levels (L3 and L4) to see how the quantities of interest depend on the mesh size. In table 2.5.6 the minimum, maximum, median and average value for the main characteristics are also given. We observe the same patterns. Furthermore,

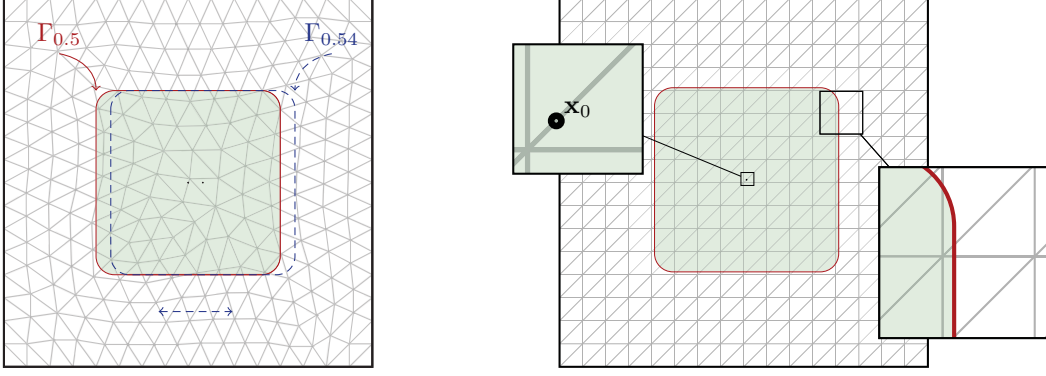


Figure 2.5.4: Computational grids of examples in section 2.5.3.2(left) and section 2.5.3.3(right). $\Gamma_{0.5}$ and $\Gamma_{0.54}$ are the interfaces corresponding to $\mathbf{x}_0 = (0.5, 0.5)$ and $\mathbf{x}_0 = (0.54, 0.5)$, respectively. The shown interface for the “sliver cut case” is obtained for $k = 2$ leading to $\mathbf{x}_0 = (0.5, 0.5) + 2^{-7}(1, 1)$.

on the finer level L3 even the best case configuration for the “naive” weighting is worse than the worst case for the HANSBO weighting, where the results do not significantly depend on the interface position. We did not compute all values for $\kappa_{1/2} = 0.5$ on level L4 in table 2.5.6 as the calculation of eigenvalues (to estimate the condition number) for matrices with a high condition number is extremely costly. Further we observe that the condition number scales with h^{-2} which is in agreement with the predictions in section 2.4. The results for the untransformed linear systems are comparable with the transformed linear systems up to a constant. As a reference, we also present numbers for the Jacobi-preconditioned matrix $\mathbf{A}_s$ which is the block matrix corresponding to the standard finite element space V_h after transformation.

2.5.3.3 Dependency on the cut position on a structured grid (sliver cut case)

Compared to section 2.5.3.2 we now consider a more extreme situation. We use the uniform grid and set $\mathbf{x}_0 = (0.5, 0.5) + \varepsilon(1, 1)$ with $\varepsilon = \varepsilon_k = 2^{-6-\frac{k}{2}}$, $k \in \{0, \dots, 52\}$. In this configuration almost all cut elements have small cuts (cf. right sketch in figure 2.5.4). A similar test case has been considered in [BH12] as “sliver cut case”. Furthermore this setting allows to examine the behavior for $\nu \rightarrow 0$ in more detail. Again, we only consider the transformed formulation. Note however that the results are very similar for the untransformed formulation. In figure 2.5.5 the main characteristics are again shown, now depending on ε .

One observes a severe dependency of the condition number on ε which scales like ε^{-2} . For the “naive” weighting the dependency of the preconditioned matrix is better (it scales with ε^{-1}) but still unbounded for $\varepsilon \rightarrow 0$. This is in contrast to the HANSBO weighting where the preconditioned matrix is well-conditioned independent of ε .

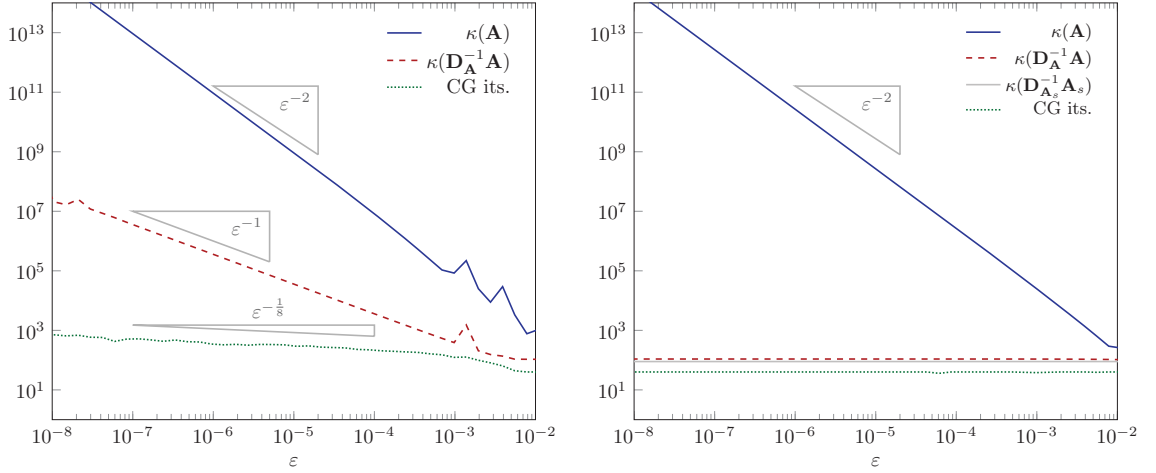


Figure 2.5.5: Condition of linear systems for different values ε for “sliver cut case” in section 2.5.3.3. For the left plot the weighting $\kappa_{1/2} = 0.5$ and for the right plot the HANSBO weighting $\kappa = \kappa^H$ is applied.

2.5.3.4 Dependency on mesh size h

In remark 2.4.3 and section 2.4.6.2 we commented on estimates for the block diagonal preconditioners $\mathbf{B}_{\mathbf{A}}$ and $\mathbf{B}_{\mathbf{D}}$ which are independent on h after transformation. In this section we want to validate these predictions and investigate also the behavior for the untransformed case numerically. For this purpose we reconsider the setup of the preceding section 2.5.3.3, but fix $\varepsilon = \varepsilon_{28} = 2^{-20} \approx 10^{-6}$ so that the condition number is large ($\approx 10^{11}$). Further we also fix the discretization method to the HANSBO weighting variant. We examine the change in the condition number of the block-preconditioned matrix under (consecutive) mesh refinements for the transformed and the untransformed formulation. For the untransformed system matrix we consider the (block-)diagonal preconditioners $\mathbf{B}_{\hat{\mathbf{A}}}$, $\mathbf{B}_{\hat{\mathbf{D}}}$ and $\mathbf{D}_{\hat{\mathbf{A}}}$ to the system matrix $\hat{\mathbf{A}}$ which are defined analogously to the preconditioners $\mathbf{B}_{\mathbf{A}}$, $\mathbf{B}_{\mathbf{D}}$ and $\mathbf{D}_{\mathbf{A}}$ for the matrix $\mathbf{A}$. Note that for the preconditioners $\mathbf{B}_{\hat{\mathbf{D}}}$ and $\mathbf{B}_{\hat{\mathbf{A}}}$ the theoretical results from section 2.4 do not apply. Note further, that we do not consider transformed preconditioners as in remark 2.4.1.

In table 2.5.7 we observe for the untransformed formulation $\kappa(\mathbf{B}_{\hat{\mathbf{A}}}^{-1}\hat{\mathbf{A}}) \simeq h^{-1}$ while we have $\kappa(\mathbf{B}_{\mathbf{A}}^{-1}\mathbf{A}) \simeq 1$ for the transformed formulation. The same behavior is observed for preconditioning with $\mathbf{B}_{\mathbf{D}}$ (or $\mathbf{B}_{\hat{\mathbf{D}}}$). For the Jacobi preconditioner we observe that the condition number of the transformed and the untransformed system behave similar, $\kappa(\mathbf{D}_{\mathbf{A}}^{-1}\mathbf{A}) \simeq h^{-2}$, $\kappa(\mathbf{D}_{\hat{\mathbf{A}}}^{-1}\hat{\mathbf{A}}) \simeq h^{-2}$. However, the iteration counts for the transformed case are significantly better.

The constants with which the condition numbers are bounded depend on the parameters λ , α and β which is discussed in the next subsection.

		L1	L2	L3	L4
untransf.	$\kappa(\mathbf{B}_{\hat{\mathbf{A}}}^{-1}\hat{\mathbf{A}})$ (its.)	2.19×10^1 (21)	4.34×10^1 (27)	8.61×10^1 (41)	1.69×10^2 (56)
	$\kappa(\mathbf{B}_{\hat{\mathbf{D}}}^{-1}\hat{\mathbf{A}})$ (its.)	2.33×10^1 (21)	4.62×10^1 (29)	9.53×10^1 (41)	1.89×10^2 (57)
	$\kappa(\mathbf{D}_{\hat{\mathbf{A}}}^{-1}\hat{\mathbf{A}})$ (its.)	4.75×10^1 (28)	1.47×10^2 (49)	5.00×10^2 (92)	1.84×10^3 (179)
transf.	$\kappa(\mathbf{B}_{\mathbf{A}}^{-1}\mathbf{A})$ (its.)	4.98×10^0 (13)	4.95×10^0 (13)	4.82×10^0 (12)	4.82×10^0 (11)
	$\kappa(\mathbf{B}_{\mathbf{D}}^{-1}\mathbf{A})$ (its.)	5.12×10^0 (13)	5.06×10^0 (13)	4.94×10^0 (12)	4.94×10^0 (11)
	$\kappa(\mathbf{D}_{\mathbf{A}}^{-1}\mathbf{A})$ (its.)	2.78×10^1 (22)	1.11×10^2 (40)	4.42×10^2 (73)	1.77×10^3 (127)

Table 2.5.7: Dependency on mesh size h for different preconditioners.

2.5.3.5 Dependency on stabilization parameter λ

In this section we want to briefly test the dependency of the iterative solvers combined with suitable preconditioners with respect to changes in the stabilization parameter. Due to the discussion in section 2.4.6.2 for the preconditioners $\mathbf{D}_{\mathbf{A}}$ and $\mathbf{B}_{\mathbf{D}}$ we expect a linear growth in the condition number for an increasing λ . We consider the same setting as in section 2.5.3.2. Now we vary the stabilization parameter λ . The results are shown in table 2.5.8. We observe that the condition number for the preconditioners $\mathbf{D}_{\mathbf{A}}$ and $\mathbf{B}_{\mathbf{D}}$ increase linearly with λ as has been predicted in the theoretical estimates. This behavior seems to be less severe for the transformed formulation. Although the condition number also increases linearly the iteration count seems to be affected only mildly.

	λ	4×10^0	4×10^1	4×10^2	4×10^4	4×10^6
untransf.	$\kappa(\mathbf{B}_{\hat{\mathbf{A}}}^{-1}\hat{\mathbf{A}})$ (its.)	4.34×10^1 (27)	4.73×10^2 (42)	4.77×10^3 (50)	4.78×10^5 (66)	4.78×10^7 (92)
	$\kappa(\mathbf{B}_{\hat{\mathbf{D}}}^{-1}\hat{\mathbf{A}})$ (its.)	4.62×10^1 (29)	6.02×10^2 (46)	6.31×10^3 (74)	6.46×10^5 (169)	6.90×10^7 (484)
	$\kappa(\mathbf{D}_{\hat{\mathbf{A}}}^{-1}\hat{\mathbf{A}})$ (its.)	1.47×10^2 (49)	6.57×10^2 (88)	6.64×10^3 (145)	7.03×10^5 (319)	7.10×10^7 (804)
transf.	$\kappa(\mathbf{B}_{\mathbf{A}}^{-1}\mathbf{A})$ (its.)	4.95×10^0 (13)	2.50×10^0 (9)	2.29×10^0 (7)	2.27×10^0 (5)	2.27×10^0 (5)
	$\kappa(\mathbf{B}_{\mathbf{D}}^{-1}\mathbf{A})$ (its.)	5.06×10^0 (13)	2.14×10^1 (13)	2.07×10^2 (14)	2.11×10^4 (19)	2.32×10^6 (35)
	$\kappa(\mathbf{D}_{\mathbf{A}}^{-1}\mathbf{A})$ (its.)	1.11×10^2 (40)	9.49×10^1 (36)	2.07×10^2 (38)	2.11×10^4 (47)	2.15×10^6 (83)

Table 2.5.8: Dependency of conditioning on λ for different preconditioners.

The block diagonal preconditioner with exact inverses $\mathbf{B}_{\mathbf{A}}$ is only positively affected. This is due to the fact that a large λ increases the angle between the subspaces V_h and V_h^x .

2.5.3.6 Dependency on problem parameters α, β

Similar to the brief discussion on the dependency on λ we consider the same example for variations in α and β . First we vary β_1/β_2 by changing β_1 and always keep $\beta_{\min} = \beta_2 = 1$ fixed and also fix $(\alpha_1, \alpha_2) = (3, 2)$. For variations in β we only consider the untransformed case as in the transformed case variations in β are equivalent to variations in α . Changes in

α are considered by variations in the ratio α_1/α_2 . For this, we set $(\beta_1, \beta_2) = (2, 1)$, $\alpha_2 = 2$ and vary α_1 .

	β_1/β_2	2×10^0	2×10^1	2×10^2	2×10^3
untransf.	$\kappa(\mathbf{B}_{\hat{\mathbf{A}}}^{-1}\hat{\mathbf{A}})$ (its.)	4.34×10^0 (27)	1.93×10^4 (74)	2.20×10^6 (111)	2.22×10^8 (114)
	$\kappa(\mathbf{B}_{\hat{\mathbf{D}}}^{-1}\hat{\mathbf{A}})$ (its.)	4.62×10^0 (29)	2.13×10^4 (87)	2.43×10^6 (139)	2.45×10^8 (96)
	$\kappa(\mathbf{D}_{\hat{\mathbf{A}}}^{-1}\hat{\mathbf{A}})$ (its.)	1.47×10^2 (29)	2.22×10^4 (146)	2.76×10^6 (219)	2.85×10^8 (277)

Table 2.5.9: Dependency on β for the preconditioners for the untransformed linear system.

	α_1/α_2	1.5×10^{-4}	1.5×10^{-2}	1.5×10^0	1.5×10^2	1.5×10^4
untransf.	$\kappa(\mathbf{B}_{\hat{\mathbf{A}}}^{-1}\hat{\mathbf{A}})$	2.14×10^3 (36)	2.74×10^1 (28)	4.34×10^1 (27)	1.85×10^3 (45)	1.82×10^5 (63)
	$\kappa(\mathbf{B}_{\hat{\mathbf{D}}}^{-1}\hat{\mathbf{A}})$	2.84×10^3 (38)	3.39×10^1 (30)	4.62×10^2 (29)	2.53×10^3 (49)	2.51×10^5 (71)
	$\kappa(\mathbf{D}_{\hat{\mathbf{A}}}^{-1}\hat{\mathbf{A}})$	2.92×10^3 (61)	6.71×10^1 (44)	1.47×10^2 (49)	9.07×10^3 (90)	9.10×10^5 (111)
transf.	$\kappa(\mathbf{B}_{\mathbf{A}}^{-1}\mathbf{A})$	5.00×10^3 (29)	5.23×10^1 (25)	4.95×10^0 (13)	5.54×10^1 (26)	5.19×10^3 (27)
	$\kappa(\mathbf{B}_{\mathbf{D}}^{-1}\mathbf{A})$	6.77×10^3 (30)	6.99×10^1 (28)	5.06×10^0 (13)	9.87×10^1 (28)	9.74×10^3 (31)
	$\kappa(\mathbf{D}_{\mathbf{A}}^{-1}\mathbf{A})$	8.39×10^3 (68)	9.09×10^1 (50)	1.11×10^2 (40)	5.90×10^3 (60)	5.92×10^5 (86)

Table 2.5.10: Dependency of conditioning on α for different preconditioners.

Again we examine the conditioning of the different preconditioners in table 2.5.10 and table 2.5.9. We first discuss the untransformed formulation. The condition number of all preconditioned matrices and their iteration counts increase quadratically with β_1/β_2 . We observe a different behavior w.r.t. changes in α . The condition number increases approximately linearly with changes in α . Moreover, the iteration counts show a dependency that seems to be less strong.

Next, we discuss the results for the transformed formulation. As in the untransformed case, changes in $\alpha_{\max}/\alpha_{\min}$ result in a linear growth of the condition number for all preconditioners. The constants, however, seem to be smaller than for the untransformed case.

2.5.3.7 Three-dimensional test case

We consider a setup in three dimensions analogously to that in section 2.5.3.4. The domain is the unit cube $\Omega = [0, 1]^3$ with a cube that is rounded off as the dividing interface. The cube, denoted as Ω_1 , is centered around $\mathbf{x}_0 = (0.5, 0.5, 0.5) + \varepsilon(1, 1, 1)$ with a small “shift parameter” $\varepsilon = 2^{-20}$. The dimensions of the cube are chosen as in section 2.5.3.1 ($d = 0.2$, $r = 0.05$) and a uniform triangulation of Ω is used. We use seven levels of uniform refinement denoted by L0,...,L6 where the coarsest level is a $2 \times 2 \times 2$ -grid.

The diffusion parameters are fixed to $(\alpha_1/\beta_1, \alpha_2/\beta_2) = (1, 3)$ and consider only the transformed problem. The Nitsche stabilization parameter is set to $\lambda = 5$. As a right hand side source term we choose $f = 1$ in Ω_1 and $f = 0$ in Ω_2 .

We investigate the performance of the CG method preconditioned with $\mathbf{B}_C$ (cf. (2.4.30)) where a multigrid preconditioner for $\mathbf{C}_s$ and diagonal scaling $\mathbf{D}_x$ is used. The multigrid solver carries out one complete V-cycle using a damped Jacobi (damping-factor 0.8) as pre- and post-smoother and a Jacobi-preconditioned CG solver on the coarsest level (L1). In table 2.5.11 the iteration counts that were necessary to reduce the initial residual by a factor of 10^{-6} for the levels L2 to L6 are shown.

	L2	L3	L4	L5	L6
CG iterations	22	25	27	29	32

Table 2.5.11: Iteration counts of multigrid-preconditioned CG method ($\lambda = 5$, $\alpha_2/\alpha_1 = 3$).

We observe that the iteration counts stay essentially bounded such that the effort for solving the linear systems is $\mathcal{O}(N)$ with N the number of degrees of freedom, i.e. $\mathbf{B}_C$ is an optimal preconditioner. The mild increase in iteration numbers further decreases if the Jacobi preconditioner $\mathbf{D}_x$ used in the subspace V_h^x is replaced by a symmetric Gauss-Seidel preconditioner. For this choice we obtain the numbers 21,23,23,25,27 for the levels L2 to L6.

2.5.3.8 Summary of results

We summarize the results w.r.t. the choice of the averaging weights, the transformation proposed in section 2.4.1 and the parameter dependencies of the preconditioners on λ , α , β .

The results in section 2.5.3.2 and section 2.5.3.3 suggest that the choice of the averaging weights is crucial. A restriction on the weights κ_i , $i = 1, 2$ as in (2.2.16) is thus not only necessary to *prove* stability (see section 2.3.1.3) but also to obtain a method that is practically useful in the case of “small cuts”.

The transformed formulation seems to be superior over the untransformed one. It is significantly more robust w.r.t. changes in β and λ while it is comparable w.r.t. the dependencies on α . Most importantly, it allows for a block diagonal preconditioning with optimal bounds on the condition number. Only based on the transformed formulation the optimal preconditioner $\mathbf{B}_C$ with a multigrid preconditioner for V_h and a diagonal scaling for V_h^x could be constructed.

For the discussion of the parameter dependencies we restrict to the transformed case. The dependency of the condition number on $\frac{\alpha_{\max}}{\alpha_{\min}}$ for all preconditioners seems to be approximately linear. Linear is also the dependency on λ of the preconditioners that involve diagonal preconditioning on the subspace V_h^x , e.g. $\mathbf{D}_A$ and $\mathbf{B}_D$.

2.5.4 Stationary, convection-dominated problem

In this example we consider a convection dominated problem and consider the discretization with the SD-Nitsche-XFEM discretization. Furthermore in all the experiments we use a slightly different stabilization parameter as in (2.2.45):

$$\gamma_T = \begin{cases} (1 - \frac{1}{P_h^T}) \frac{h_T}{2\|\mathbf{w}\|_{\infty,T}} & \text{if } P_h^T > 1 \\ 0 & \text{if } P_h^T \leq 1 \end{cases} \quad (2.5.5)$$

This choice can also be found in [ESW05]. We investigate two examples in two space dimensions with a known solution. Again we use the notation $\varepsilon = \alpha$ to emphasize that α is small. This example can also be found in [LR12].

2.5.4.1 Problem with a smooth solution

We consider a two-dimensional stationary problem with a smooth solution. The interface is $\Gamma = \{y = 0\}$ and the domains are $\Omega_1 = [-1, 1] \times [-1, 0]$ and $\Omega_2 = [-1, 1] \times [0, 1]$. The piecewise constant coefficients ε, β are chosen as $\varepsilon = (\varepsilon_1, \varepsilon_2) = (2 \cdot 10^{-7}, 1 \cdot 10^{-7})$, $\beta = (\beta_1, \beta_2) = (3, 2)$ and a stationary velocity field is given by $\mathbf{w} = (1, 0)$. We adapt the right hand side f and the Dirichlet boundary conditions such that u defines the solution to our problem, with

$$u(x, y) = \begin{cases} \frac{2}{3} \sin(\pi(x + y)) & \text{for } (x, y) \in \Omega_1, \\ \sin(\pi(x + \frac{4}{3}y)) & \text{for } (x, y) \in \Omega_2. \end{cases}$$

The problem is solved on an unstructured mesh with 240 elements (on the coarsest level) by the proposed stabilized method. The coarsest mesh (L1) is uniformly refined five times. The norms used in the error analysis of section 2.3 for the error $e_h = u - u_h$ are listed in table 2.5.12 and table 2.5.13 for the “convective” and the “diffusive” scaling of the Nitsche penalty parameter λ .

Level	$\ \mathbf{w} \cdot \nabla e_h\ _0$ (eoc)	$\ e_h\ _0$ (eoc)	$\ [\![\beta e_h]\!]\ _\Gamma$ (eoc)
L1	1.58×10^0 (-)	1.67×10^{-1} (-)	1.25×10^{-2} (-)
L2	7.83×10^{-1} (1.0)	4.41×10^{-2} (1.9)	2.06×10^{-3} (2.6)
L3	3.88×10^{-1} (1.0)	9.58×10^{-3} (2.2)	5.60×10^{-4} (1.9)
L4	1.93×10^{-1} (1.0)	2.04×10^{-3} (2.2)	1.13×10^{-4} (2.3)
L5	9.62×10^{-2} (1.0)	4.57×10^{-4} (2.2)	3.48×10^{-5} (1.7)
L6	4.80×10^{-2} (1.0)	1.07×10^{-4} (2.1)	1.08×10^{-5} (1.7)

Table 2.5.12: Convergence under successively uniform mesh refinement of different measures of error $e_h := u - u_h$ for the convection-diffusion example with a smooth solution for the “convective” scaling.

We observe the expected linear convergence in the norm $\|\mathbf{w} \cdot \nabla e_h\|_0$. The (β -weighted) L^2 -norm of the error ($\|e_h\|_0$) converges with $\mathcal{O}(h^2)$ which is half an order better than in

Level	$\ \mathbf{w} \cdot \nabla e_h\ _0$ (eoc)	$\ e_h\ _0$ (eoc)	$\ [\![\beta e_h]\!]\ _\Gamma$ (eoc)
L1	1.57×10^0 (-)	1.75×10^{-1} (-)	5.76×10^{-1} (-)
L2	7.83×10^{-1} (1.0)	4.41×10^{-2} (2.0)	1.14×10^{-1} (2.3)
L3	3.88×10^{-1} (1.0)	9.62×10^{-3} (2.2)	2.76×10^{-2} (2.1)
L4	1.93×10^{-1} (1.0)	2.06×10^{-3} (2.2)	6.03×10^{-3} (2.2)
L5	9.62×10^{-2} (1.0)	4.60×10^{-4} (2.2)	1.76×10^{-3} (1.8)
L6	4.80×10^{-2} (1.0)	1.07×10^{-4} (2.1)	5.40×10^{-4} (1.7)

Table 2.5.13: Convergence under successively uniform mesh refinement of different measures of error $e_h := u - u_h$ for the convection-diffusion example with a smooth solution for the “diffusive” scaling.

the a priori estimates. For the interface jump error the order of convergence appears to be close to 2 which is better than expected for both scalings of the Nitsche stabilization. The convective scaling leads to an interface error which is roughly 50 times smaller than for the diffusive scaled Nitsche method.

2.5.4.2 Problem with a sharp layer

In this example a two-dimensional stationary problem with a parabolic layer at the interface is considered. The interface is $\Gamma = \{y = 0\}$ and the domains are $\Omega_1 = [0.25, 2] \times [-1, 0]$ and $\Omega_2 = [0.25, 2] \times [0, 1]$. The piecewise constant coefficients ε , β are chosen as $\varepsilon = (\varepsilon_1, \varepsilon_2) = (9 \times 10^{-7}, 4 \times 10^{-7})$, $\beta = (\beta_1, \beta_2) = (27, 11)$ and a stationary velocity field is given by $\mathbf{w} = (1, 0)$. We adapt the right hand side f and the boundary

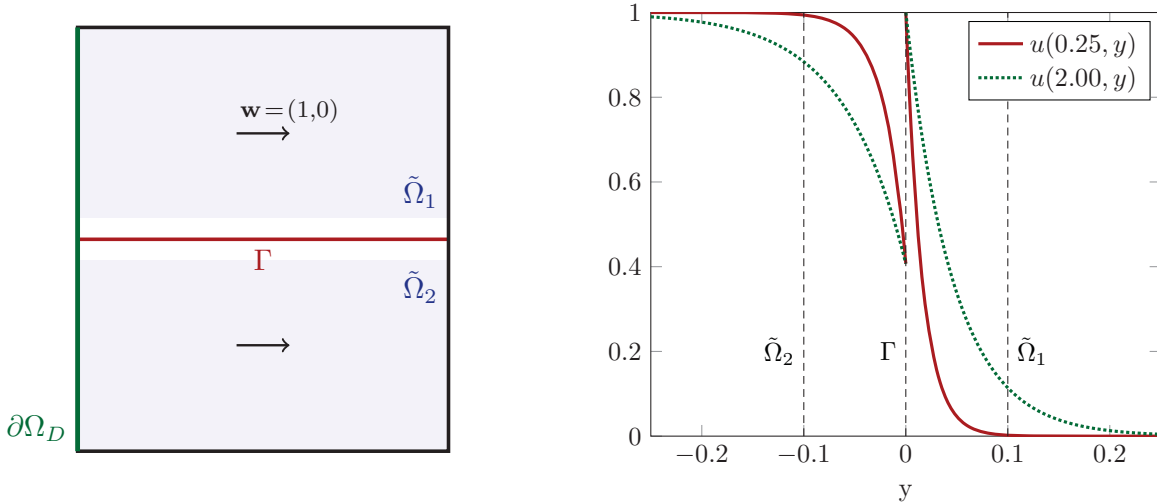


Figure 2.5.6: Sketch of setup and solution at in- and outflow of example in section 2.5.4.2.

conditions such that the solution to our problem is given by

$$u(x, y) = \begin{cases} 1 - \frac{16}{27} \exp(\frac{C_n}{\sqrt{x}} y) & \text{for } (x, y) \in \Omega_1, \\ \exp(\frac{-C_p}{\sqrt{x}} y) & \text{for } (x, y) \in \Omega_2, \end{cases}$$

where the constants C_p and C_n are chosen s.t. the width of the layers at the outflow ($x = 2$) is approximately 10% of the domain size ($C_n \approx 23.0$, $C_p \approx 30.7$). The solution at the inflow and outflow boundary close to the interface is displayed in figure 2.5.6. According to the solution u we prescribe Dirichlet boundary conditions on $\partial\Omega_D := \{x = 0.25\}$ and Neumann boundary conditions $g = \varepsilon \nabla u \cdot \mathbf{n}$ on $\partial\Omega \setminus \partial\Omega_D$.

The problem is discretized on an unstructured triangular mesh with 400 elements on the coarsest mesh which is uniformly refined five times. The errors within the domain are measured in $\tilde{\Omega} = \{|y| > 0.1\}$ away from the interface. In figure 2.5.7 and figure 2.5.8 the convergence of the errors in the (β -weighted) L^2 -norm of the solution and the streamline derivative as well as the interface jump and flux error are displayed.

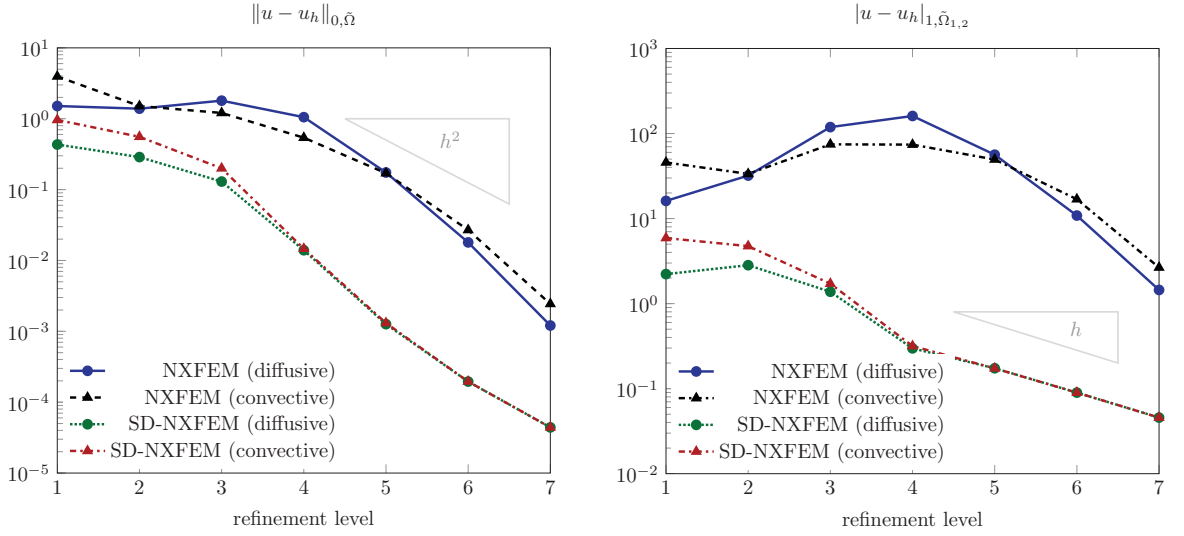


Figure 2.5.7: Convergence plots for the volume error for example in section 2.5.4.2.

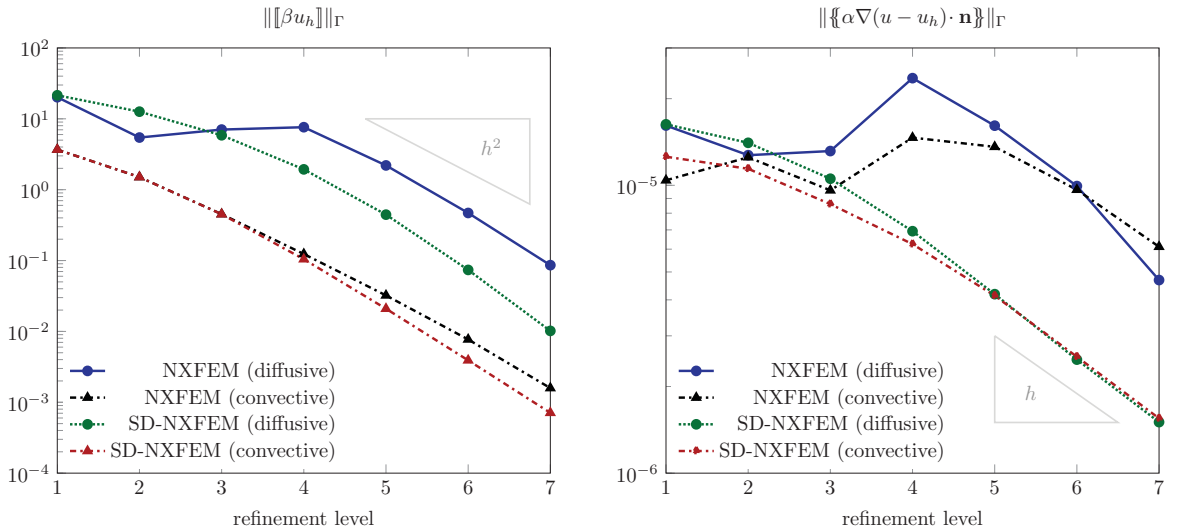


Figure 2.5.8: Convergence plots for the interface errors for example in section 2.5.4.2.

We observe that the error of the streamline derivative is drastically improved by the stabilized methods. In contrast to the stabilized methods the error of the unstabilized methods are not even monotonically decreasing. In the (β -weighted) L^2 -norm one also observes a significant improvement due to the stabilization. Concerning the different scalings of the Nitsche stabilization it is expected that the convective scaling leads to a better resolution of the interface jump condition. This is confirmed by the results in figure 2.5.8.

2.5.5 Transient convection-dominated problem

As a last and most realistic example we consider a *three-dimensional* time dependent *convection dominated* problem with a stationary interface. This example and the numerical results for the SD-Nitsche-XFEM method have also been published in [LR12].

2.5.5.1 Problem description

The domain $\Omega := [0, 2] \times [0, 2] \times [0, 1]$ is separated into a cylindrical domain $\Omega_1 := \{(x, y, z) \in \mathbb{R}^3 : (x - 1)^2 + (y - 1)^2 < R^2\}$, with $R = 0.25$, and $\Omega_2 := \Omega \setminus \Omega_1$ by the *stationary* interface $\Gamma := \partial\Omega_1 \setminus \partial\Omega$. The piecewise constant coefficients ε, β are chosen as $\varepsilon = (\varepsilon_1, \varepsilon_2) = (10^{-4}, 2 \times 10^{-4})$, $\beta = (\beta_1, \beta_2) = (3, 1)$ and a stationary velocity field is given by

$$\mathbf{w}|_{\Omega_1} = (0, 0, 0), \quad \mathbf{w}|_{\Omega_2} = (1 + R^2(d_y^2 - d_x^2)r^{-4}, -2R^2(d_x d_y)r^{-4}, 0) \quad (2.5.6)$$

where $d_x := x - 1$, $d_y := y - 1$ and $r := (d_x^2 + d_y^2)^{\frac{1}{2}}$. A sketch of the domains and of $\mathbf{w}$ in term of field-lines is given in figure 2.5.9.

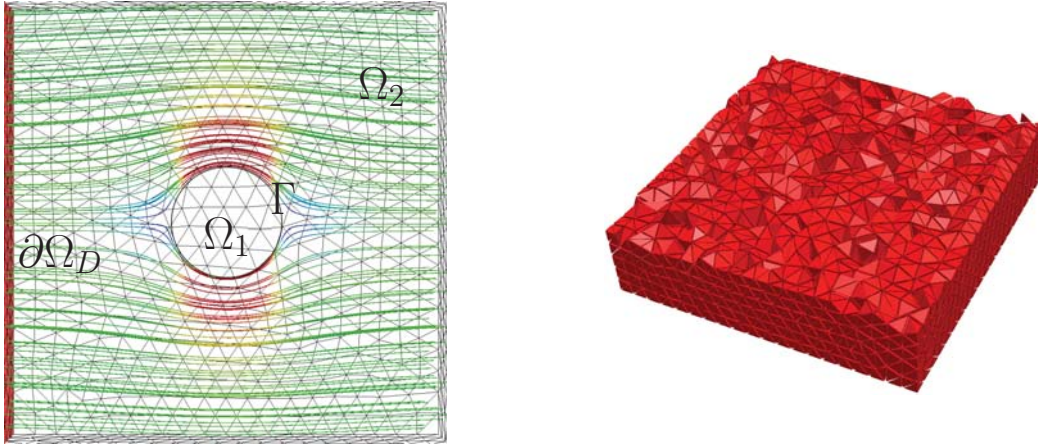


Figure 2.5.9: Sketch of interface position and flow field (left) and mesh (right)

Assumption 1.2.4 (*compatible velocity*) on the velocity field is satisfied: $\operatorname{div} \mathbf{w} = 0$ in both domains and $\mathbf{w} \cdot \mathbf{n} = 0$ on Γ . We impose a Dirichlet boundary condition on the

inflow boundary $\partial\Omega_D := \{(x, y, z) \in \Omega : x = 0\}$, s.t. $u|_{\partial\Omega_D} = 0.05$ and a homogeneous Neumann boundary condition $\varepsilon \nabla u \cdot \mathbf{n} = 0$ on $\partial\Omega \setminus \partial\Omega_D$. As initial condition we take $u = 0$ on Ω_1 , $u = 0.05$ on Ω_2 . Note that this initial condition does not satisfy the Henry interface condition (2.1.1c).

This time dependent convection-diffusion problem is strongly convection dominated with a physical Péclet number $P_L := \frac{2\|\mathbf{w}\|}{\varepsilon} \approx 2 \cdot 10^4$. Furthermore, due to the inconsistent (w.r.t. condition (2.1.1c)) initial condition a *parabolic* boundary layer of thickness $\mathcal{O}(\sqrt{\varepsilon t})$ at the interface will form directly after $t = 0$, independent of the velocity field. For $t \rightarrow \infty$ the solution converges to the stationary piecewise constant function $u = 0.05 \beta^{-1}$. In figure 2.5.10 the solution along a line is displayed, where one observes the predicted boundary layer behavior. In the experiments we consider $t = 1$.

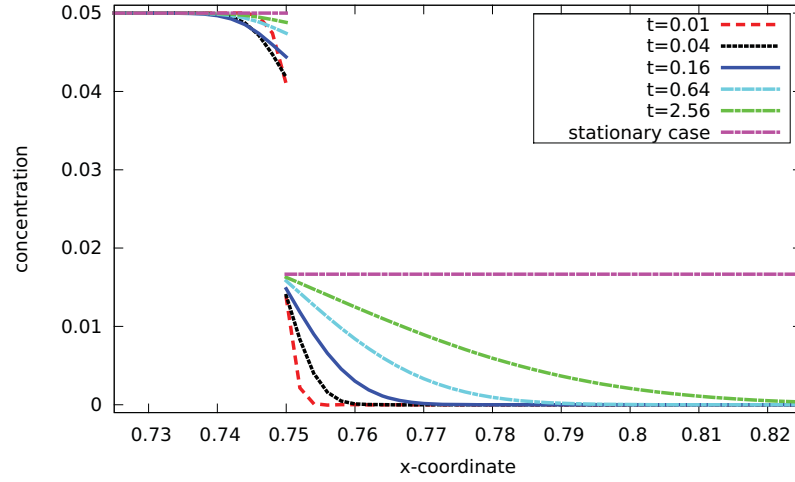


Figure 2.5.10: Concentration profile through line $\gamma_x := \{(x, y, z) \in \Omega : y = 1, z = 0.5\}$ for several values of t in example in section 2.5.5.

2.5.5.2 Discretization

We use the mesh with 30000 elements displayed in figure 2.5.9 with an average mesh size $h = 0.05$ and element Péclet numbers up to $P_h^T \approx 250$. Thus, the mesh resolution is too low to resolve the boundary layer (for $t \leq 1$).

We are primarily interested in the accuracy of the spatial discretization. Hence, in the implicit Euler method ((2.2.55) with $\theta = 1$) we choose a small time step size $\Delta t = 10^{-4}$, such that the total discretization error is dominated by the spatial discretization error.

Again we consider the same four methods as in section 2.5.4.2. The solution is essentially two-dimensional which allowed us to compute a reference solution on a very fine two-dimensional mesh which is aligned to the interface and resolves the boundary layer for $t > 10^{-2}$. This reference solution is used to provide the profiles in figure 2.5.10 and the reference profiles in figure 2.5.12 below.

2.5.5.3 Numerical results

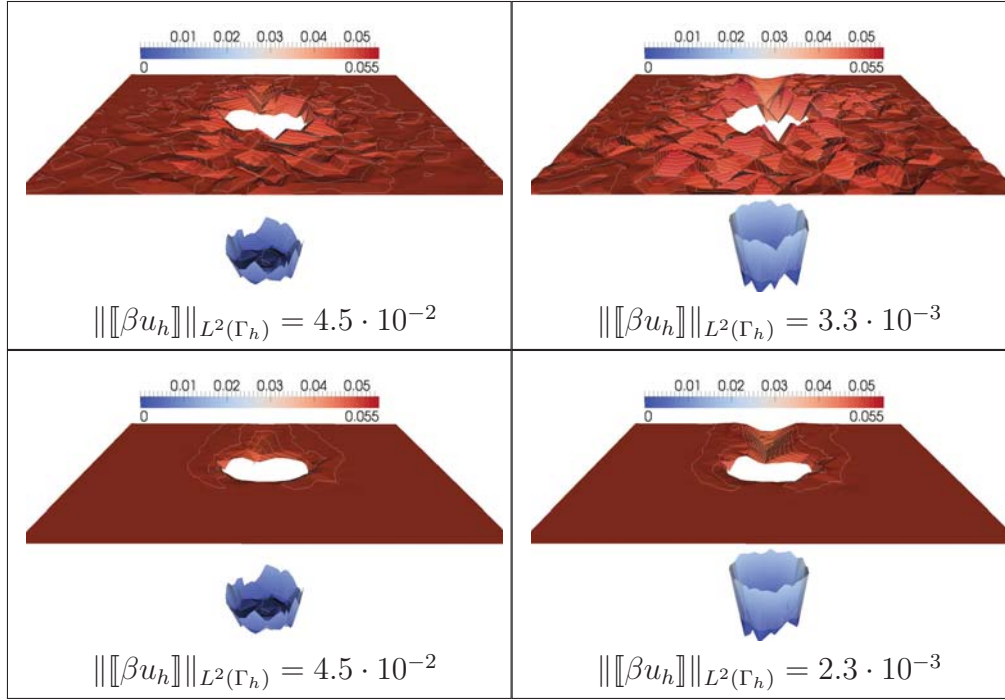


Figure 2.5.11: Numerical solution in the plane $z = 0.5$ at $t = 1$ for Nitsche-XFEM (top) and SD-Nitsche-XFEM (bottom), with diffusive scaling (left) and convective scaling (right) of the Nitsche stabilization.

In figure 2.5.11 the numerical solution in the plane $z = 0.5$ at $t = 1$ (where the boundary layer has a width of approximately 0.01 in Ω_2) is shown for four different methods. Below each picture we also give the L^2 norm of the jump $\llbracket \beta u_h \rrbracket$ on the approximate interface Γ_h .

We observe several effects. The first one also occurred in the numerical experiment treated in section 2.5.4.2: if one considers the different scalings in the Nitsche method, i.e. the left and the right columns in figure 2.5.11, then the convective scaling results in a better approximation of the interface condition. But it also increases the effect of non-physical oscillations. Comparing the first and the second row in figure 2.5.11, we see that the Streamline Diffusion stabilization suppresses the oscillations whereas the quality of the approximation of the interface condition is not negatively affected by this stabilization.

In figure 2.5.12 the numerical solutions of the same four methods as in figure 2.5.11 together with the reference solution, on the line $z = 0.5$, $y = 1.0$ in Ω_2 at time $t = 1$ are shown. One can observe that the boundary layer which is represented well by the reference solution is not resolved accurately by any of the four methods. Especially for $x > 1.25$, i.e. downwind of Ω_1 none of the methods yields a discrete solution that is close to the reference solution. The solutions u_h of the SD-Nitsche-XFEM methods are much

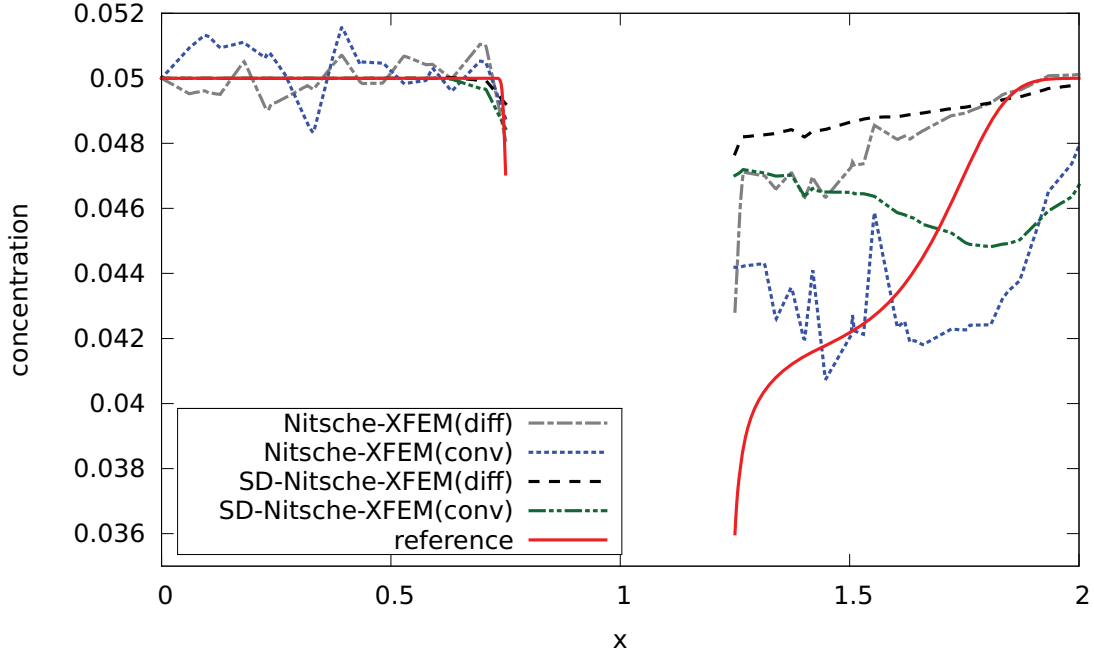


Figure 2.5.12: Numerical solutions on the line $z = 0.5, y = 1.0$ at time $t = 1$ obtained with Nitsche-XFEM, SD-Nitsche-XFEM, and the reference solution.

smoother than the solutions obtained without stabilization and upwind of Ω_1 , where the solution is almost constant outside the boundary layer, it is very accurate.

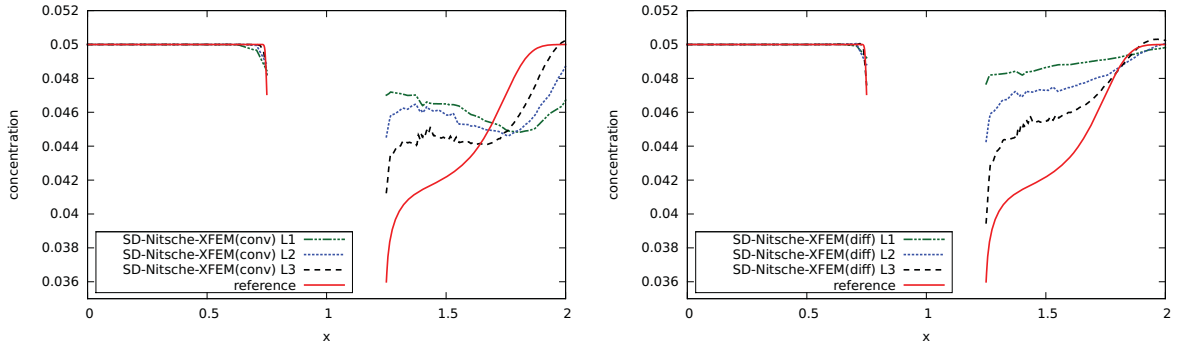


Figure 2.5.13: Numerical solutions on the line $z = 0.5, y = 1.0$ at time $t = 1$ obtained with SD-Nitsche-XFEM applying the convective scaling (left) and the diffusive scaling (right) on three consecutively refined meshes and the reference solution.

In figure 2.5.13 the results of the SD-Nitsche-XFEM methods on three successively (uniformly) refined meshes are shown. The resolution of the boundary layer at $t = 1$ improves if the grid is refined, but on level 3 the discrete solution downwind of Ω_1 is still not in good agreement with the reference solution. This can be explained as follows. For small times t the boundary layers are much smaller, namely $\mathcal{O}(\sqrt{\varepsilon t})$, cf. figure 2.5.10,

and cannot be resolved. For small t we thus have (very) large spatial discretization errors. If time evolves until $t = 1$ these large errors are transported in downwind direction and are only mildly damped. This time dependent transport effect causes the large errors downwind of Ω_1 ($x > 1.25$) in figure 2.5.12 and figure 2.5.13. We also note that we can not expect the Streamline Diffusion stabilization to enhance the resolution of our discretization. But, and this effect becomes evident in this example, it suppresses non-physical oscillations. We further observe that there is a qualitative difference of the concentration profiles downwind of the cylinder for the diffusive and the convective scaling. Nevertheless, we can not conclude that one of both is better than the other.

CHAPTER 3

Mass transport through a moving interface

In the previous chapter, chapter 2, we discussed the special case of the problem in (1.2.1) where the interface is stationary. The more general case, the case of a moving interface, is more difficult. This concerns the theoretical analysis as well as the derivation and implementation of a suitable discretization. In this chapter we discuss the discretization of the mass transport problem in an Eulerian setting for a moving interface with an implicit description.

Outline of this chapter

In section 3.1 the mathematical model is presented and a well-posed weak formulation of this model is given. The arising numerical challenges for the discretization and different solution strategies used in the literature are presented. In section 3.2 we present an approach to solve the problem numerically using a combination of a space-time formulation, the XFEM space and the Nitsche technique. Error analysis of this discretization is challenging. In section 3.3 we present an error analysis which provides a second order error bound in space and time. The analysis is essentially based on the one presented in [LR13]. An important difference, however, is the weaker interpolation operator used in this work. In section 3.4 interesting numerical examples show the performance of the proposed discretization method for test problems. A challenging aspect of the combination of space-time formulations and the XFEM space is the efficient solution of linear systems arising from such discretizations. We conclude the chapter with the discussion of first solution strategies in section 3.5.

3.1 Problem description

We recall the problem under consideration. In this section we consider the following problem with a moving interface.

Problem 3.1.1.

$$\partial_t u + \mathbf{w} \cdot \nabla u - \operatorname{div}(\alpha \nabla u) = f \quad \text{in } \Omega_i(t), \quad i = 1, 2, \quad t \in (0, T], \quad (3.1.1a)$$

$$[\![\alpha \nabla u \cdot \mathbf{n}]\!]_{\Gamma} = 0 \quad \text{on } \Gamma(t), \quad t \in (0, T], \quad (3.1.1b)$$

$$[\![\beta u]\!]_{\Gamma} = 0 \quad \text{on } \Gamma(t), \quad t \in (0, T], \quad (3.1.1c)$$

$$u(\cdot, 0) = u^0 \quad \text{in } \Omega_i(t), \quad i = 1, 2, \quad (3.1.1d)$$

$$u(\cdot, t) = g_D \quad \text{on } \partial\Omega, \quad t \in (0, T]. \quad (3.1.1e)$$

Note that due to assumption 1.2.4 (compatible velocity) we require $\mathbf{w} \cdot \mathbf{n} = \mathcal{V} \cdot \mathbf{n}$ on $\Gamma(t)$ and $\operatorname{div} \mathbf{w} = 0$ in $\Omega_i(t)$, $i = 1, 2$, $\forall t \in (0, T]$. Here, $\mathcal{V} \cdot \mathbf{n}$ is the velocity of the interface motion in normal direction.

3.1.1 Weak formulation

In this section we discuss a weak formulation of the problem 3.1.1 for the case of a time-dependent interface $\Gamma = \Gamma(t)$ and a time-dependent velocity-field $\mathbf{w} = \mathbf{w}(x, t)$. The major difficulty of this setting arises from the fact that the space $H_{0,\beta}^1(\Omega)$ in (2.1.6) is no longer independent of the time t and thus a weak formulation as in section 2.1.2 can not be applied. In [GR11, Chapter 10.3] a *space-time* weak formulation including a proper analysis is presented. We briefly present the weak formulation and the most important results. For details, we refer to [GR11, Chapter 10.3].

Anisotropic Sobolev spaces. We introduce notation for the space-time geometries and suitable (anisotropic) Sobolev spaces on them. The space-time domain is denoted by $Q = \Omega \times I \in \mathbb{R}^{d+1}$ with $I = (0, T]$. It is separated into the sub-domains $Q_i := \{(\mathbf{x}, t) \in Q : x \in \Omega_i(t), t \in I\}$, $i = 1, 2$ by the space-time interface $\overline{Q_1} \cap \overline{Q_2} = \Gamma_* = \bigcup_{t \in (0, T]} \Gamma(t)$. We need anisotropic Sobolev spaces (also called *t*-anisotropic Sobolev spaces) in which spatial derivatives (D_x) and temporal derivatives (∂_t) are treated differently. For $i = 1, 2$ we define

$$H^{k,l}(Q_i) := \{u \mid \partial_t^p D_x^\alpha u \in L^2(Q_i), p, q \in \mathbb{N}, q = |\alpha|, \frac{q}{k} + \frac{p}{l} \leq 1\}. \quad (3.1.2)$$

For this definition of the space it is clear that for $u \in H^{2,2}(Q_i)$ we have $\nabla u \in H^{1,1}(Q_i)$. Thus we retain the isotropic Sobolev space for $k = l$, $H^k(Q_i) = H^{k,k}(Q_i)$. For the cylindrical case ($\Omega_i = \text{const}$) the space $H^1(I; L^2(\Omega_i)) \cap L^2(I; H^2(\Omega_i))$ which is very similar to $H^{2,1}(Q_i)$ has been introduced in [LSU68, LM72] for parabolic problems.

The space $H^{1,0}(Q_i)$ is important as for functions in $H^{1,0}(Q_i)$ and sufficiently smooth Γ_* the trace operation on the space-time interface Γ_* and the boundary $\partial\Omega \times (0, T]$ is well-defined. We denote the space of functions in $H^{k,l}(Q_i)$ with zero values at the *spatial boundary* of the space-time cylinder $\partial\Omega \times (0, T]$ as $H_0^{k,l}(Q_i)$. We further denote the dual space of $H_0^{1,0}(Q)$ (continuous across the interface) as $H^{-1,0}(Q)$.

Similar to the weak formulation for the stationary interface we introduce a space in which all functions fulfill the interface condition $\llbracket \beta u \rrbracket = 0$:

$$V_\beta := \{u \in L^2(Q_i), u \in H_0^{1,0}(Q_i), i = 1, 2, \llbracket \beta u \rrbracket_{\Gamma_*} = 0\} \quad (3.1.3)$$

Note that there holds $v \in V_\beta \Leftrightarrow \beta v \in H_0^{1,0}(Q)$. The solution to problem 3.1.1 should be searched for in a space with additional regularity on the temporal derivative. We introduce W_β and $W_{\beta,0}$, two of such spaces. For all functions in $W_{\beta,0}$ homogeneous initial values are imposed.

$$W_\beta := \{u \in V_\beta, \partial_t u \in H^{-1,0}(Q)\}, \quad W_{\beta,0} := \{u \in W_\beta, u(\cdot, 0) = 0 \text{ in } \Omega\} \quad (3.1.4)$$

The spaces W_β and $W_{\beta,0}$ equipped with the norm

$$\|\cdot\|_W := (\|\cdot\|_{L^2(Q)}^2 + \|\nabla \cdot\|_{L^2(Q_1 \cup Q_2)}^2 + \|\partial_t \cdot\|_{H^{-1,0}(Q)}^2)^{\frac{1}{2}}$$

are Hilbert spaces. We further introduce the β -weighted scalar products (for $S_i \in \{Q_i, \Omega_i(t)\}$, $t \in (0, T]$)

$$(u, v)_{S_{1,2}} = (u, v)_{0,S_{1,2}} = (\beta u, v)_{L^2(S_1 \cup S_2)} = \sum_{i=1,2} \int_{S_i} \beta_i u v \, d\mathbf{x} dt, \quad (3.1.5)$$

$$(u, v)_{1,S_{1,2}} = (\beta \nabla u, \nabla v)_{L^2(S_1 \cup S_2)} = \sum_{i=1,2} \int_{S_i} \beta_i \nabla u \nabla v \, d\mathbf{x} dt. \quad (3.1.6)$$

with the induced (semi-)norms $\|\cdot\|_{S_{1,2}}$ and $|\cdot|_{1,S_{1,2}}$. Note that in these scalar products and seminorms there is a scaling with the piecewise constant function β and that opposite to the isotropic norm $|\cdot|_1$ on $H^1(Q)$ there is no first derivative w.r.t. time in (3.1.6).

For sufficiently smooth functions we define a spatial scalar product for a fixed time t , $m \in \{0, 1\}$

$$(u, v)_{m,t} := \sum_{|\alpha| \leq m} \int_{\Omega} \beta(\cdot, t) D^\alpha u(\cdot, t) D^\alpha v(\cdot, t) \, dx.$$

The induced norm is denoted by $\|\cdot\|_{m,t}$. We further introduce the weaker norm for $u \in L^2(\Omega)$

$$\|u\|_{-1,t} := \sup_{v \in H_0^1(\Omega_{1,2}(t))} \frac{(u, v)_{\Omega_{1,2}(t)}}{\|v\|_{1,\Omega_{1,2}(t)}}.$$

The weak formulation reads as:

Determine $u \in W_{\beta,0}$ such that

$$\langle \partial_t u, v \rangle - \int_Q u \mathbf{w} \cdot \nabla v \, d\mathbf{x} \, dt + \sum_{i=1,2} \int_{Q_i} \alpha_i \nabla u_i \nabla v \, d\mathbf{x} \, dt = \int_Q f v \, d\mathbf{x} \, dt \quad (3.1.7)$$

for all $v \in H_0^{1,0}(Q)$ with $\langle \cdot, \cdot \rangle = \langle \cdot, \cdot \rangle_{H^{-1,0}(Q) \times H_0^{1,0}(Q)}$. Important for the discretization is the following characterization of this formulation. Note that $\beta v \in H_0^{1,0}(Q)$ holds for all $v \in V_\beta$. Thus using the test space V_β in (3.1.7) gives

$$\langle \partial_t u, \beta v \rangle - (u, \mathbf{w} \cdot \nabla v)_{Q_{1,2}} + (\alpha \nabla u, \nabla v)_{Q_{1,2}} = (f, v)_{Q_{1,2}} \quad (3.1.8)$$

For $f \in L^2(Q)$ a unique solution $u \in W_{\beta,0}$ to (3.1.7) exists and $\|u\|_W \leq c \|f\|_{L^2(Q)}$ for a c independent of f , cf. [GR11, Chapter 10.3].

3.1.2 Solution strategies

The discretization of problem 3.1.1 is very challenging due to the fact that the equations in (3.1.1a) are defined on *time-dependent* domains and are coupled through an interface condition which leads to discontinuities across the interface. Problems similar to problem 3.1.1 appear also in other fields, for instance in fluid-structure interaction problems or combustion. Different approaches exist to tackle the problem. We give a brief overview of methods which are suitable to deal with problems of the form of problem 3.1.1.

The balance laws in fluid dynamics are usually described in a fixed control volume, that is, in an *Eulerian frame* of reference. Another possibility to state balance equations is the use of an *Lagrangian frame* of reference, that means that the frame of reference follows a certain particle. Based on this different ways of formulating the same physical balance laws, different methods to discretize equations exist. We discuss four different approaches to deal with time integration on *time-dependent* domains. An illustration of the four approaches is given in figures 3.1.1-3.1.4.

Method of Lines. In chapter 2 we applied the method of lines to derive fully discrete formulation of the Nitsche-XFEM method for an unsteady problem with a stationary interface. In the method of lines a time derivative is at some place replaced with a finite difference approximation, for instance

$$\partial_t u(\mathbf{x}, t) \approx \delta_t^{FD} u(\mathbf{x}, t) = \frac{u(\mathbf{x}, t + \Delta t) - u(\mathbf{x}, t - \Delta t)}{2\Delta t}.$$

This makes sense as long as u is sufficiently smooth in time, because then we have $\delta_t^{FD} u \rightarrow \partial_t u$ for $\Delta t \rightarrow 0$. However, this is in general not true for solutions of problem 3.1.1. Assume $\mathbf{x} \in \Gamma(t)$ such that for sufficiently small Δt we have $\mathbf{x} \in \Omega_1(t - \Delta t) \cap \Omega_2(t + \Delta t)$, then for $\Delta t \rightarrow 0$ we have $|\delta_t^{FD}| \rightarrow \infty$ as long as $\beta_1 \neq \beta_2$. This is indicated in figure 3.1.1.

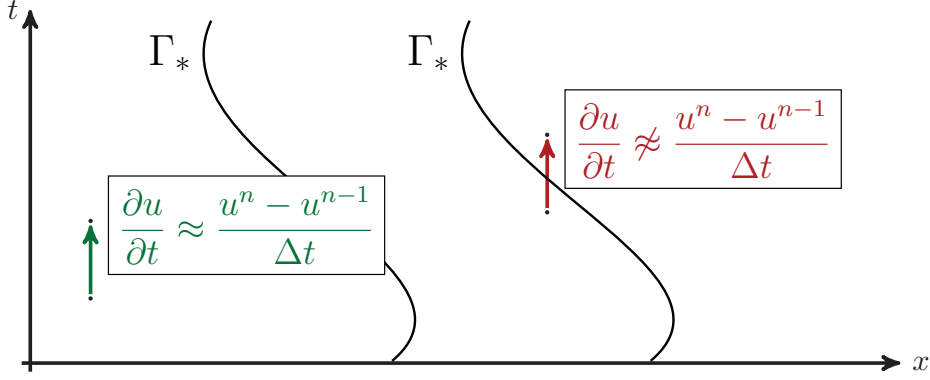


Figure 3.1.1: Sketch of the method of lines for a two-phase problem.

For the case $\beta_1 = \beta_2$ the situation is better as the solution u is continuous in time. Nevertheless higher order time derivatives are in general not continuous across the space-time interface (as long as $\alpha_1 \neq \alpha_2$). Thus, a method of lines approach will not achieve higher order accuracy in time. In [FZ09] and [Zun13] a combination of a XFEM discretization in space and the method of lines for problems with moving domains is considered. In both cases a problem with only a *weak discontinuity*, i.e. $\beta_1 = \beta_2$, is considered. In [Zun13] an implicit Euler method is analyzed and first order accuracy in time is proven.

For the general case $\beta_1 \neq \beta_2$ we can transform the problem to the form of problem 1.2.1 and apply the method of lines, cf. also the discussion in section 3.2.4.2. Note however that we can not expect convergence of higher order due to the low regularity of $\partial_t \tilde{u}$, where $\tilde{u}$ is the transformed variable $\tilde{u} = \beta u$.

Arbitrary-Lagrangian-Eulerian formulations. A very popular approach for problems involving moving boundaries or interfaces is the Arbitrary-Lagrangian-Eulerian (ALE) formulation where the underlying computational mesh is deformed in a way such that moving boundaries (or interfaces) stay fixed with respect to the reference configuration. Let ξ be the coordinate of a boundary (or interface) point in the reference coordinates. Then the evolution of a value at a grid point can be used to express the time derivative

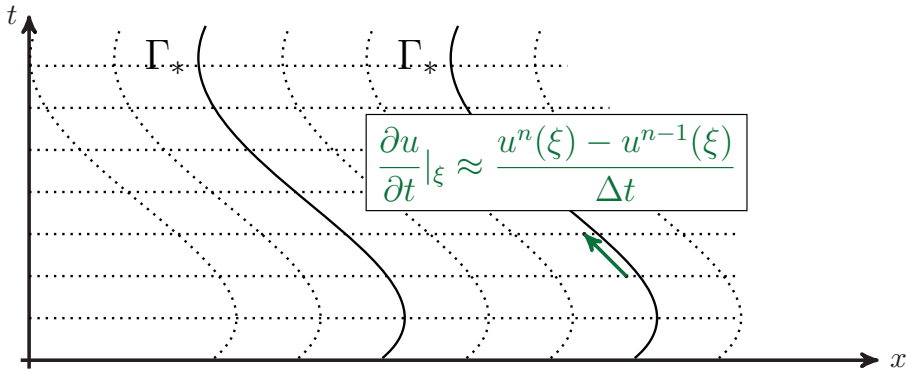


Figure 3.1.2: Sketch of the Arbitrary-Lagrangian-Eulerian (ALE) method.

(in the reference system) as the rate of change of that value (in the physical domain) and

the gradient in advection direction:

$$\partial_t u|_\xi = \partial_t u + \mathbf{w}_{\text{rel}} \cdot \nabla u$$

with $\mathbf{w}_{\text{rel}} = \frac{\partial \mathbf{x}}{\partial t}|_{\xi=\text{const}}$ the mesh motion at the corresponding position. The solution is smooth in time with respect to points with constant reference coordinates and thus a standard method of lines approach can be applied to discretize $\partial_t u|_\xi$ (cf. figure 3.1.2). This approach is often used, for instance for fluid-structure interaction problems. The ALE description is sometimes also combined with a space-time formulation for the time discretization (see discussion below).

Depending on the application the successive deformation of the mesh can lead to very large deformations. In those situations the computational domain has to be meshed again after a few time steps. This remeshing is usually challenging and can be very time-consuming. If the problem of remeshing does not appear (due to small deformations only) or can be solved satisfactory the method works well. We refer the reader to [DH03, Chapter 3.10] and [DHPRF04] and references therein for an overview on ALE methods.

For us this approach is not an option as we do not want to fit our computational mesh to the interface.

Semi-Lagrangian methods. Another approach is the use of the so called *semi-Lagrangian* method or characteristic finite element method, where the material derivative $\dot{u} = \partial_t u + \mathbf{w} \cdot \nabla u$ is approximated with a finite difference approximation along a (approximated) space-time trajectory. That means one approximates (for an implicit Euler

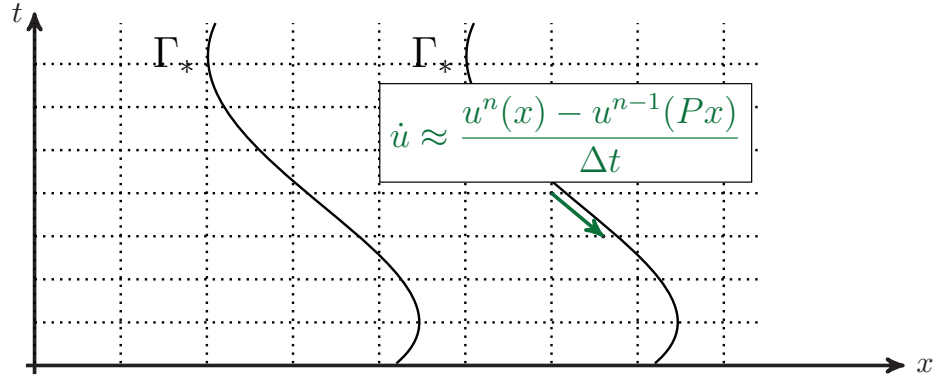


Figure 3.1.3: Sketch of the semi-lagrangian method.

discretization)

$$\dot{u}(\mathbf{x}, t_n) = \partial_t u(\mathbf{x}, t_n) + \mathbf{w} \cdot \nabla u(\mathbf{x}, t_n) \approx \frac{u(\mathbf{x}, t_n) - u(P(\mathbf{x}), t_{n-1})}{\Delta t}$$

where $P(\mathbf{x})$ is the origin of the trajectory through $\mathbf{x}$ in the sense that $P(\mathbf{x}) = \mathbf{y}(0)$ with $\mathbf{y}$ the solution of $\dot{\mathbf{y}} = \mathbf{w}(\mathbf{y}(t), t_{n-1} + t)$, $\mathbf{y}(\Delta t) = \mathbf{x}$ (cf. figure 3.1.3). This guarantees that

for every point $\mathbf{x}$ in domain i the corresponding origin point $P(\mathbf{x})$ also lies in domain i . Thus the approximation of the time derivative takes place along a line where the solution is smooth, s.t. simple approximation ideas as in the method of lines can be applied. Another advantage of the method is the fact that due to the Lagrangian handling of the time derivative no convection term appears in the resulting equations for the new unknown. One then arrives at a problem of the form

Problem 3.1.2.

$$\begin{aligned} \frac{1}{\Delta t} u^n(\mathbf{x}) - \operatorname{div}(\alpha \nabla u^n(\mathbf{x})) &= f^n(\mathbf{x}) + \frac{1}{\Delta t} u^{n-1}(P(\mathbf{x})) && \text{in } \Omega_i, \quad i = 1, 2, \\ \llbracket \alpha \nabla u^n \cdot \mathbf{n} \rrbracket_\Gamma &= 0 && \text{on } \Gamma, \\ \llbracket \beta u^n \rrbracket_\Gamma &= 0 && \text{on } \Gamma, \\ u^n &= g_D && \text{on } \partial\Omega. \end{aligned}$$

The Nitsche-XFEM method discussed in chapter 2 is very well suited for the discretization of this semi-discrete problem. However the approach comes with several disadvantages. In a finite element setting the operator $P(\mathbf{x})$ is non-local which makes it difficult to efficiently implement the terms corresponding to the old time values especially on unstructured meshes. Also, a suitable choice for $P(\mathbf{x})$ is not obvious if $\mathbf{x}$ is close to a Neumann-boundary condition. Furthermore a rigorous error analysis of this kind of methods seems to be very hard. Methods based on the idea of characteristics (or a *semi-Lagrangian* point of view) have been applied for convection-diffusion equations in [DR82, RT02], for the Navier-Stokes equations in [CHCOB09, ME98], for surfactants equations in [HLZ13] and in [HWGW14] a method combining the *semi-Lagrangian* point of view with a XFEM discretization in space has been considered, however, without any analysis.

Space-Time formulations. The last approach that we want to mention is also the one that we consider in the remainder of this chapter. The basic idea is to consider the problem 3.1.1 directly as a stationary problem in $d + 1$ dimension and to discretize it as such. In order to keep the computational costs comparable to standard time discretization strategies (for example the method of lines), the space-time domain is divided into time slabs which correspond to the time intervals usually used in time stepping methods. Then, a variational formulation is applied which decouples the time slabs such that the computational structure is that of a time stepping scheme. We discuss the details in the remaining part of this section.

Discretizations based on space-time formulations have been considered for fluid flow in, for instance [TBML92, TLB92, Beh01, SvdVvD06, SBvdV06, KvdVvdV06, Beh08, Neu13, Wel13]. In [Tho97] the space-time formulation for parabolic problems is discussed and extensively analyzed. In all these publications problems with matching boundaries(interfaces) in space-time and smooth solutions are considered.

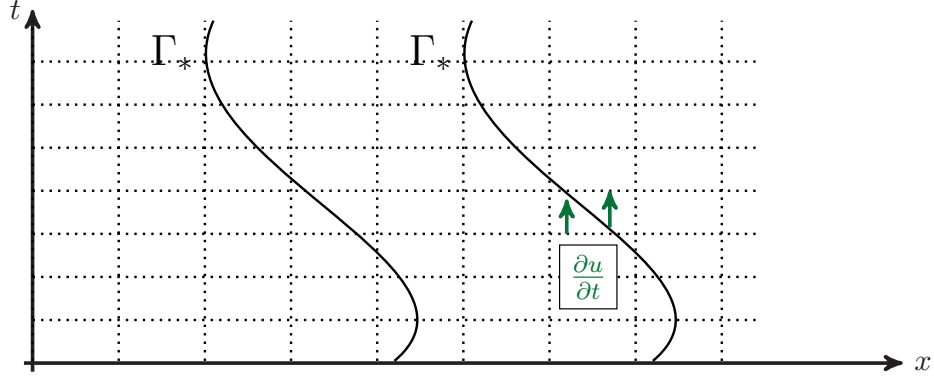


Figure 3.1.4: Sketch of a space-time method.

3.2 Discretization of the moving interface problem

In this section we present the Space-Time-DG Nitsche-XFEM discretization of problem 3.1.1. In the name of the method the “Space-Time-DG” corresponds to a space-time Discontinuous Galerkin formulation, “XFEM” corresponds to the way the discrete finite element space is constructed and “Nitsche” refers to the technique that is used to (weakly) impose the interface condition. We introduce the components one after another. In comparison to the discretization in chapter 2 the essential new aspect is the formulation of the problem in a space-time setting. This needs some preparation. Accordingly, in section 3.2.1 we introduce notation. In section 3.2.2 we introduce the space-time DG formulation for a simple (one-domain) parabolic problem. Corresponding to the space-time setting we have to generalize the realization of an appropriate XFEM space. This is done in section 3.2.3. Afterwards, in section 3.2.4 we present the Space-Time-DG Nitsche-XFEM discretization.

3.2.1 Space-time notation

We adopt and adapt the notation for triangulations, domains, etc. from chapter 2 and introduce additional notation for space-time related geometries and quantities. Let d be the spatial dimension of Ω , such that $\Omega \in \mathbb{R}^d$. Within each time slab $Q^n := \Omega \times I_n$, $I_n = (t_{n-1}, t_n]$ we assume that the triangulation of the spatial domain Ω is a shape regular decomposition into simplices $\mathcal{T}_n = \{T\}$. The corresponding characteristic mesh size is denoted by h_n . Corresponding to a triangulation $\mathcal{T}_n$ we have a canonical triangulation of Q^n into $d + 1$ -dimensional prisms. This triangulation is denoted by $\mathcal{T}_{n,*} = \{Q^T\}$ where for each prism we have $Q^T = T \times I_n$ for a corresponding d -dimensional simplex T .

Note, that for different time slabs the triangulation is allowed to change. Further, the triangulation is *not fitted* to the interface $\Gamma(t)$ (cf. figure 3.2.1).

The space-time interface $\Gamma_* := \bigcup_{t \in (0, T]} \Gamma(t)$ separates the space-time domain into its subdomains $Q_i := \bigcup_{t \in (0, T]} \Omega_i(t)$, $i = 1, 2$. Within each time slab we define $\Gamma_*^n :=$

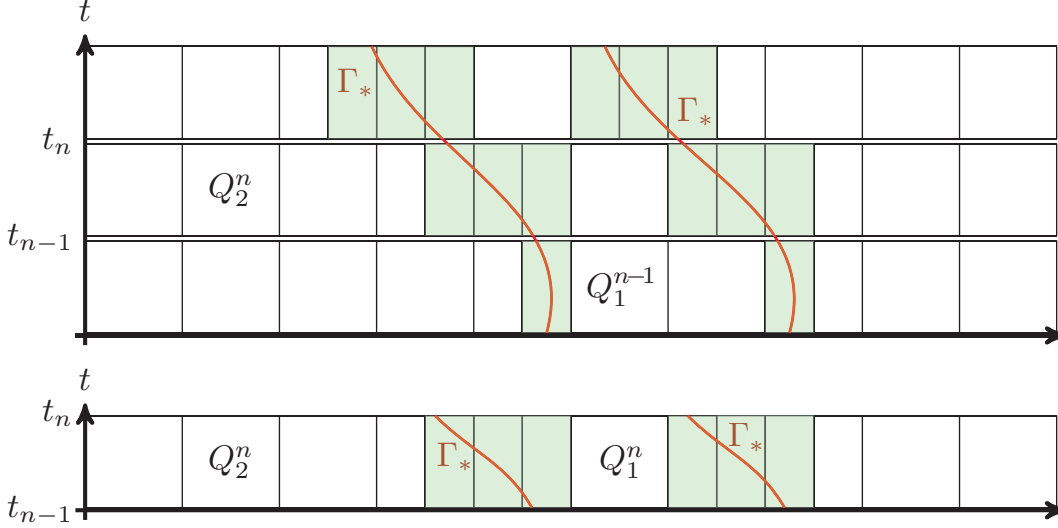


Figure 3.2.1: Sketch of the space-time domains Q_i^n . Note that within each time slab the triangulation has a tensor product structure $\mathcal{T}_h \times [t_{n-1}, t_n]$.

$\bigcup_{t \in I_n} \Gamma(t)$ and the subdomains $Q_i^n := \bigcup_{t \in I_n} \Omega_i(t)$, $i = 1, 2$.

We introduce some notation for *cut prism elements*, i.e. elements Q^T with $\Gamma_* \cap Q^T \neq \emptyset$. For any prism $Q^T \in \mathcal{T}_{n,*}$ with $Q_i^T := Q^T \cap Q_i^n$ we denote the part of Q^T in Q_i^n and with $\Gamma_T^* := Q^T \cap \Gamma_*$ the part of the interface that lies in Q^T . With $\mathcal{T}_{n,*}^\Gamma$ we denote the set of (prism) elements that are “close to the interface”, $\mathcal{T}_{n,*}^\Gamma := \{Q^T : Q^T \cap \Gamma_* \neq \emptyset\}$. The corresponding domain is $Q^{n,\Gamma} = \{\mathbf{x} \in Q^T : Q^T \in \mathcal{T}_{n,*}^\Gamma\}$. Further we define the set of elements with nonzero support in one domain: $\mathcal{T}_{n,*}^i := \{Q^T : Q^T \cap Q_i \neq \emptyset\}$, $i = 1, 2$, the corresponding domain is denoted by $Q_i^{n,+} = \{\mathbf{x} \in Q^T : Q^T \in \mathcal{T}_{n,*}^i\}$. We also define the domain of uncut elements in domain i as $Q_i^{n,-} = Q_i^n \setminus Q^{n,\Gamma} = Q_i^{n,+} \setminus Q^{n,\Gamma}$. We further introduce the abbreviation $\Omega_i^n = \Omega_i(t_n)$.

3.2.2 Space-time DG formulation for a parabolic model problem

To introduce the space-time Discontinuous Galerkin formulation we consider the (much) simpler one-domain problem

Problem 3.2.1.

$$\partial_t u - \operatorname{div}(\alpha \nabla u) = f \quad \text{in } \Omega \times (0, T] \quad (3.2.1a)$$

$$u = 0 \quad \text{on } \partial\Omega \times (0, T], \quad (3.2.1b)$$

$$u = u^0 \quad \text{on } \Omega \times \{0\}. \quad (3.2.1c)$$

For ease of presentation we assume the use of an equidistant time discretization. A partitioning of the time interval is given by $0 = t_0 < t_1 < \dots < t_N = T$ with a uniform time step $\Delta t = T/N$. We define the time interval $I_n = (t_{n-1}, t_n]$ and assume a shape

regular simplicial triangulation $\mathcal{T}_h^n$ of the spatial domain Ω . In general, the triangulation is allowed to change with each time step n . Let V_h be a standard finite element space corresponding to $\mathcal{T}_h^n$. On the time slab $Q^n := \Omega \times I_n$ we introduce the tensor-product finite element space

$$W_n := \{v : Q^n \rightarrow \mathbb{R} | v(\mathbf{x}, t) = \sum_{m=0}^k t^m \varphi_m(\mathbf{x}), \varphi_m \in V_h, m = 0, \dots, k\}. \quad (3.2.2)$$

Note that due to the tensor-product structure the order of the ansatz functions can be chosen differently in space and time directions. We typically consider the case of piecewise linear functions for V_h combined with linear functions in time ($k = 1$ in (3.2.2)). The corresponding finite element space for the whole space-time domain is

$$W := \{v : Q \rightarrow \mathbb{R} | v|_{Q^n} \in W_n\}. \quad (3.2.3)$$

Note that there is no temporal continuity between the time slabs imposed on W .

The discretization is derived as follows. First we test (3.2.1a) with a test function $v \in W_n$ and integrate over one time slab Q^n . As usual we do partial integration in space for the second order term which yields (for a sufficiently smooth function u)

$$\int_{Q^n} \partial_t u v + \alpha \nabla u \nabla v d\mathbf{x} dt = \int_{Q^n} f v d\mathbf{x} dt \quad (3.2.4)$$

To abbreviate notation later on, we introduce the (bi)linear forms

$$a_s^n(u, v) := \int_{Q^n} \alpha \nabla u \nabla v d\mathbf{x} dt, \quad d_s^n(u, v) := \int_{Q^n} \partial_t u v d\mathbf{x} dt, \quad f_s^n(v) := \int_{Q^n} f v d\mathbf{x} dt. \quad (3.2.5)$$

We use the subscript s to indicate the correspondence to the simpler problem 3.2.1. To include information from the past we apply a standard upwind technique. The time derivative can be seen as an advection term in the space-time domain. In that sense $\Omega \times \{t_{n-1}\}$ is the inflow boundary of Q^n where inflow information has to be provided. This is done by adding the upwind stabilization

$$b_s^n(u, v) := \int_{\Omega} \llbracket u \rrbracket^{n-1} v_+^{n-1} d\mathbf{x} \quad (3.2.6)$$

with the time jump and time trace operators

$$\llbracket u \rrbracket^n := u_+^n - u_-^n, \quad u_-^n := u(\cdot, t_n), \quad u_+^n := \lim_{\epsilon \downarrow 0} u(\cdot, t_n + \epsilon).$$

Adding the equations for each time slab n together we get the overall space-time DG discretization of problem 3.2.1. u_-^0 in the upwind term is given initial data and moved to the r.h.s. side of the variational formulation. For notational convenience we define

$\llbracket u \rrbracket^0 = u_+^0$, s.t. $b^1(u, v) = (u_+^0, v_+^0)$. The discrete variational formulation reads as: Find $u_h \in W$ such that

$$B_s(u_h, v_h) := d_s(u_h, v_h) + a_s(u_h, v_h) + b_s(u_h, v_h) = f_s(v_h) + c_s^0(v_h) \quad \forall v_h \in W. \quad (3.2.7)$$

with $c_s^0(v_h) := \int_{\Omega} u^0 v_{h,+}^0 d\mathbf{x}$ while the other (bi)linear forms are obtained by summing the corresponding (bi)linear forms over all time slabs, for instance $d_s(\cdot, \cdot) = \sum_{n=1}^N d_s^n(\cdot, \cdot)$. Note that the coupling in time has a direction in the sense that the time steps $< k$ do not depend on the solution at time step k . The discretization thus naturally divides into time stepping problems which can be solved one after another. The problem for each time slab n is obtained by testing (3.2.7) with $v \in W \cap \{v(\mathbf{x}, t) = 0, t \notin I_n\}$ and reads as: Given $u_-^{n-1} \in L^2(\Omega)$ and $f \in L^2(Q^n)$, determine $u_h \in W_n$ such that for every $v_h \in W_n$ there holds

$$B_s^{n,*}(u_h, v_h) := d_s^n(u_h, v_h) + a_s^n(u_h, v_h) + b_s^{n,*}(u_h, v_h) = f_s^n(v_h) + c_s^{n-1}(v_h)$$

with $b_s^{n,*}(u, v) := \int_{\Omega} u_+^{n-1} v_+^{n-1} d\mathbf{x}$ and $c_s^{n-1}(u, v) := \int_{\Omega} u_-^{n-1} v_+^{n-1} d\mathbf{x}$. Due to the discontinuous-in-time finite element space W this time discretization is a Discontinuous Galerkin (in time) method. In [Tho97] the method including a thorough error analysis is discussed.

Remark 3.2.1 (Petrov-Galerkin DG methods). *In the presented method the test and the solution space coincide. This is not necessary. A popular alternative is the use of a Petrov-Galerkin method using continuous (in time) finite elements for the solution space combined with discontinuous finite elements (in time) of a lower degree for the test space. This reduces the number of unknowns, but still allows for a time stepping procedure. In the context of an XFEM finite element space and a two-domain discretization with a space-time finite element method the use of different spaces for solution and test space is not straight-forward, cf. remark 3.2.6. In the following we thus restrict to the discussion of Discontinuous Galerkin in time methods where test and ansatz spaces coincide.*

Remark 3.2.2 (Energy-stability). *Assume that there is no source term $f = 0$. A property of the exact solution to the problem 3.2.1, is that the energy of the system can not increase (as long as there are no sources due to f or boundary conditions). Here, the energy is $W(u) = \frac{1}{2} \|u\|^2$ and there holds $\partial_t W(u) \leq 0$. A nice property of the DG discretization is that this property also holds on the discrete level, in the sense $\frac{1}{2} \|u_-^n\|_{\Omega}^2 \leq \frac{1}{2} \|u_-^{n-1}\|_{\Omega}^2$, cf. [LR13, lemma 3.5].*

Remark 3.2.3 (Stability of the DG time integration method). *The discontinuous-in-time Galerkin method as a time integration method is stable and has the optimal smoothing property, but is not A-stable, cf. [GR11, Section 11.5.2].*

3.2.3 Space-time extended finite elements

We present the XFEM space analogously to the presentation of the XFEM space for the stationary problem in chapter 2. Again, we restrict to the case of piecewise bilinear

functions (linear in space and linear in time) here. The XFEM space in a space-time setting has previously been introduced in [GR11, Chapter 11.5.2], [LR13] and [Leh15]. The idea of a combination of space-time methods and XFEM has also been suggested in [CB04, CB06] for a one dimensional hyperbolic problem without any error analysis.

The generalization of the XFEM space introduced in chapter 2 is obtained by the use of a generalization of the restriction operators $\mathcal{R}_i$. The space-time XFEM spaces are given by

$$\begin{aligned} W_n^\Gamma &:= \mathcal{R}_1^n W_n \oplus \mathcal{R}_2^n W_n \\ W^\Gamma &:= \{v : Q \rightarrow \mathbb{R} \mid v|_{Q^n} \in W_n^\Gamma\} = \mathcal{R}_1 W \oplus \mathcal{R}_2 W. \end{aligned} \quad (3.2.8)$$

with the corresponding canonical restrictions $\mathcal{R}_i^n$ on $L^2(Q^n)$, $\mathcal{R}_i$ on $L^2(Q)$ given by

$$\mathcal{R}_i^n v = \begin{cases} v|_{Q_i^n} & \text{on } Q_i^n \\ 0 & \text{on } Q^n \setminus Q_i^n, \end{cases} \quad \mathcal{R}_i v = \begin{cases} v|_{Q_i} & \text{on } Q_i \\ 0 & \text{on } Q \setminus Q_i. \end{cases}$$

We will also use the notation $v_i := \mathcal{R}_i v$.

We again give a characterization corresponding to the name extended finite element space (XFEM) for the case of *piecewise linear functions in space and time*. Let $\{q_j\}_{j \in \mathcal{J}}$ be the nodal basis in the finite element space V_h . The vertex corresponding to q_j is denoted by x_j . To each q_j there correspond two space-time basis functions, namely $q_{j,0}(x, t) := \frac{1}{\Delta t}(t_n - t)q_j(x)$ and $q_{j,1}(x, t) := \frac{1}{\Delta t}(t - t_{n-1})q_j(x)$. The index set of basis functions in the space-time finite element space W_n “close to the interface” is given by

$$\mathcal{J}_{\Gamma_*^n} := \{(j, 0), (j, 1) \mid \text{meas}_d(\Gamma_*^n \cap \text{supp}(q_j)) > 0\}.$$

Let $H_{\Gamma_*^n}$ be the characteristic function corresponding to Q_2^n , i.e. $H_{\Gamma_*^n}(x, t) = 1$ if $(x, t) \in Q_2^n$ and zero otherwise. For each space-time node index $(j, \ell) \in \mathcal{J}_{\Gamma_*^n}$ a so-called enrichment function corresponding to the node $(x_j, t_{n-\ell})$ is given by

$$\Phi_{j,\ell}(x, t) := |H_{\Gamma_*^n}(x, t) - H_{\Gamma_*^n}(x_j, t_{n-\ell})|. \quad (3.2.9)$$

New basis functions are defined as follows:

$$q_{j,\ell}^{\Gamma_*^n} := q_{j,\ell} \Phi_{j,\ell}, \quad (j, \ell) \in \mathcal{J}_{\Gamma_*^n}. \quad (3.2.10)$$

An illustration for the spatially one-dimensional case is given in figure 3.2.2.

The term $H_{\Gamma_*^n}(x_j, t_{n-\ell})$ in the definition of $\Phi_{j,\ell}$ is constant and may be omitted (as it doesn't introduce new functions in the function space), but ensures that $q_{j,\ell}^{\Gamma_*^n}(x_j, t_{n-\ell}) = 0$ holds in all space-time grid points $(x_j, t_{n-\ell})$. The space-time XFEM space on the time slab $Q^n = \Omega \times I_n$ is given by

$$W_n^\Gamma = W_n \oplus W_n^x, \quad \text{with } W_n^x := \text{span}\{q_{j,\ell}^{\Gamma_*^n} \mid (j, \ell) \in \mathcal{J}_{\Gamma_*^n}\}.$$

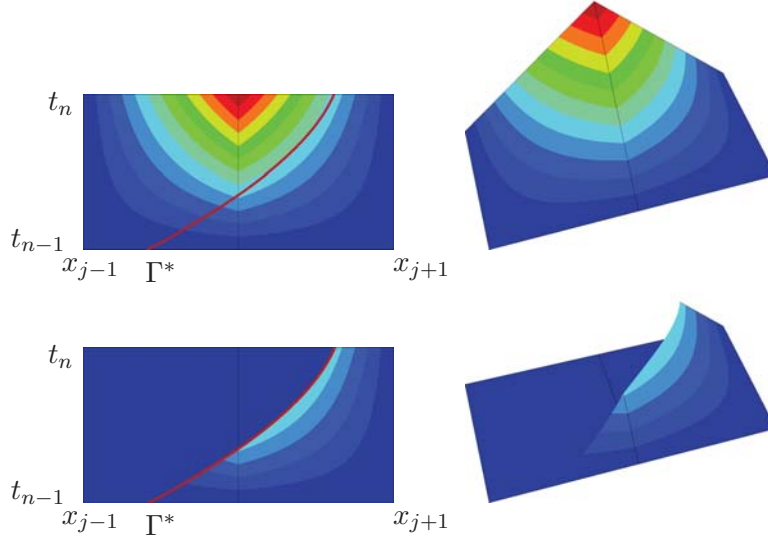


Figure 3.2.2: Sketch of bilinear space-time XFEM functions in $1 + 1$ dimensions. The top row shows the basis function $q_{j,1}$, the bottom row the XFEM function $q_{j,1}^{\Gamma_*^n}$. The functions are shown in the space-time domain (left) and as an elevation plot (right).

This characterization shows that the extended finite element space W_n^Γ is obtained by adding to the standard space W_n new basis functions that are discontinuous across the space-time interface Γ_*^n , cf. (3.2.10). There holds the approximation property

$$\inf_{v_h \in W^\Gamma} \|v - v_h\|_{m,Q} \leq c(\Delta t^{2-m} + h^{2-m}) \|v\|_{H^2(Q_{1,2})} \quad \forall v \in H^2(Q_{1,2}), \quad m = 0, 1.$$

A proof of this is given in section 3.3.2.

3.2.4 Nitsche formulation for interface conditions in space-time

At this point we defined a finite element space which, by construction, gives good approximation properties. However, the interface condition is not respected. In this section we generalize the Nitsche technique presented in chapter 2, section 2.2.2 to the space-time case.

3.2.4.1 Derivation of the method

We assume that the solution to problem 3.1.1 is smooth, multiply (3.1.1a) with an arbitrary test function $\beta v \in W^\Gamma$ and integrate over the space-time domain. Afterwards, on every time slab we apply partial integration in space for the diffusion operator which

yields

$$\sum_{n=1}^N \left\{ (\partial_t u + \mathbf{w} \cdot \nabla u, v)_{Q_{1,2}^n} + (\alpha \nabla u, \nabla v)_{Q_{1,2}^n} - \int_{t_{n-1}}^{t_n} \int_{\Gamma(t)} \llbracket \alpha \nabla u \cdot \mathbf{n} \beta v \rrbracket d\mathbf{s} dt \right\} = (f, v)_{Q_{1,2}} \quad (3.2.11)$$

To transform the iterated integrals which appear in the Nitsche formulation to an integral on the space-time interface we use the following transformation formula:

$$\int_{t_{n-1}}^{t_n} \int_{\Gamma(t)} f(\mathbf{s}, t) d\mathbf{s} dt = \int_{\Gamma_*^n} f(\mathbf{s}) (1 + (\mathbf{w} \cdot \mathbf{n})^2)^{-\frac{1}{2}} d\mathbf{s} =: \int_{\Gamma_*^n} f(\mathbf{s}) \nu(\mathbf{s}) d\mathbf{s},$$

with $\nu(\mathbf{s}) = (1 + (\mathbf{w} \cdot \mathbf{n})^2)^{-\frac{1}{2}}$ and $\mathbf{w}$ the (interface) velocity. Note that $d\mathbf{s}$ denotes both the surface measure on $\Gamma(t)$ as well as on Γ_* . Under the assumption that the space-time interface is sufficiently smooth, there holds for a constant $c_0 > 0$

$$c_0 \leq \nu(\mathbf{s}) \leq 1 \quad \text{for all } \mathbf{s} \in \Gamma_*.$$

We define the scalar product on the space-time interface (on one time slab)

$$(u, v)_{\Gamma_*^n} := \int_{\Gamma_*^n} u v d\mathbf{s}.$$

We manipulate the normal flux term similar to what we did in chapter 2, section 2.2.2.1, where we replaced $\llbracket \alpha \nabla u \cdot \mathbf{n} \beta v \rrbracket$ with $\{\!\!\{ \alpha \nabla u \cdot \mathbf{n} \}\!\!\} \llbracket \beta v \rrbracket$ due to $\alpha \nabla u \cdot \mathbf{n}$ being uni-valued at the interface. Similar to what we did in section 2.2.2.1, we define

$$N_c^n(u, v) := -(\{\!\!\{ \alpha \nabla u \cdot \mathbf{n} \}\!\!\}, \llbracket \beta v \rrbracket)_{\Gamma_*^n}.$$

and, with additional symmetry and stabilization terms define

$$N^n(u, v) := N_c^n(u, v) + N_c^n(v, u) + N_s^n(u, v), \quad N_s^n(u, v) := (\bar{\alpha} \lambda h^{-1} \llbracket \beta u \rrbracket, \llbracket \beta v \rrbracket)_{\Gamma_*^n}. \quad (3.2.12)$$

We define $N(\cdot, \cdot)$, $N_c(\cdot, \cdot)$, $N_s(\cdot, \cdot)$ by summation over all time slabs, for instance $N(\cdot, \cdot) = \sum_{n=1}^N N^n(\cdot, \cdot)$. The weights in $\{\!\!\{ \}\!\!\}$ are chosen as in the HANSBO-choice for the case of a stationary interface. Take $t \in I_n$, $T \in \mathcal{T}_n$ and let $T_i(t) := T \cap \Omega_i(t)$. We define the weights

$$(\kappa_i(t))_{|T} := \frac{|T_i(t)|}{|T|}. \quad (3.2.13)$$

Note that those weights only depend on the *spatial* configuration at a given time t and there holds $\kappa_1(t) + \kappa_2(t) = 1$. We define the weighted average

$$\{\!\!\{ v \}\!\!\} := \kappa_1(t) (\mathcal{R}_1 v)_{|\Gamma(t)} + \kappa_2(t) (\mathcal{R}_2 v)_{|\Gamma(t)}. \quad (3.2.14)$$

Note that now κ_i depends on time.

To finish the discretization we need to (weakly) add continuity in time. This is done with an upwind stabilization term as in (3.2.6). We define the upwind bilinear form

$$b(u, v) = \sum_{n=1}^N b^n(u, v) := \sum_{n=1}^N (\llbracket u \rrbracket^{n-1}, v_+^{n-1})_{\Omega_{1,2}^{n-1}} = \sum_{n=1}^N \sum_{i=1,2} \int_{\Omega_i^{n-1}} \beta_i \llbracket u \rrbracket^{n-1} v_+^{n-1} d\mathbf{x}. \quad (3.2.15)$$

Together, we obtain a discrete variational formulation:

Find $u_h \in W^\Gamma$ such that

$$\begin{aligned} B(u_h, v_h) &= f(v_h) + c(u^0, v_h) \quad \forall v_h \in W^\Gamma \\ \text{with } B(u_h, v_h) &:= d(u_h, v_h) + a(u_h, v_h) + b(u_h, v_h) + N(u_h, v_h) \end{aligned} \quad (3.2.16)$$

and the bilinear forms

$$a(u, v) = \sum_{n=1}^N a^n(u, v) := (\alpha \nabla u, \nabla v)_{Q_{1,2}}, \quad (\text{diffusion}) \quad (3.2.17a)$$

$$d(u, v) = \sum_{n=1}^N d^n(u, v) := \sum_{n=1}^N (\partial_t u + \mathbf{w} \cdot \nabla u, v)_{Q_{1,2}^n}, \quad (\text{space-time convection}) \quad (3.2.17b)$$

$$c(w, v) := (w, v_+^0)_{\Omega_{1,2}^{n-1}}. \quad (\text{initial data}) \quad (3.2.17c)$$

and

$$f(v) = \sum_{n=1}^N f^n(v) := (f, v)_{Q_{1,2}^n}. \quad (\text{source}) \quad (3.2.17d)$$

This weak formulation can be rewritten if we apply partial integration on $d(\cdot, \cdot)$ and make use of homogeneous boundary conditions. For all $u, v \in W^\Gamma + H^1(Q_{1,2})$ there holds

$$B(u, v) = B'(u, v) := d'(u, v) + a(u, v) + b'(u, v) + N(u, v) \quad (3.2.18)$$

with

$$d'(u, v) := - \sum_{n=1}^N (u, \partial_t v + \mathbf{w} \cdot \nabla v)_{Q_{1,2}^n}, \quad (3.2.19a)$$

$$b'(u, v) := - \sum_{n=1}^{N-1} (u_-^n, \llbracket v \rrbracket^n)_{\Omega_{1,2}^n} + (u_-^N, v_-^N)_{\Omega_{1,2}^N}. \quad (3.2.19b)$$

The Space-Time-DG Nitsche-XFEM method defined in (3.2.16) allows for a solution time step by time step. By testing with a test function which has only support within one time slab, we get the time slab problem:

Find $u \in W^\Gamma$, s.t.

$$B^n(u, v) = a^n(u, v) + \hat{b}^n(u, v) + d^n(u, v) + N^n(u, v) = f^n(v) + c^n(u^{n-1}, v) \quad (3.2.20)$$

where the upwind coupling bi- and linear forms within one time step are

$$\hat{b}^n(u, v) = \sum_{i=1,2} \int_{\Omega_i^{n-1}} \beta_i u_+^{n-1} v_+^{n-1} d\mathbf{x} \quad (3.2.21)$$

and

$$c^n(w, v) = \sum_{i=1,2} \int_{\Omega_i^{n-1}} \beta_i w v_+^{n-1} d\mathbf{x}. \quad (3.2.22)$$

Hence, the computational overhead of the method is determined by the costs for the setup and the solution of the linear systems arising from (3.2.20). The efficient solution of arising linear systems is discussed in section 3.5.

Remark 3.2.4 (Mass conservation). *The Space-Time-DG Nitsche-XFEM discretization is globally mass conserving w.r.t. the discrete time points $t = t_n$. To see that test (3.2.16) with $v = \beta^{-1} \in W^\Gamma$ and use the characterization (3.2.18) to get*

$$\int_{\Omega} u_-(\cdot, t_N) d\mathbf{x} = \int_{\Omega} u^0 d\mathbf{x} + \int_Q f d\mathbf{x}dt.$$

The same mass balance holds for the true solution. For each time slab one gets a (time) local version of this mass balance:

$$\int_{\Omega} u_-(\cdot, t_n) d\mathbf{x} = \int_{\Omega} u_-(\cdot, t_{n-1}) d\mathbf{x} + \int_{Q^n} f d\mathbf{x}dt.$$

Remark 3.2.5 (Integration on space-time domains). *An implementation of the discretization method defined in (3.2.16) or (3.2.20) needs to compute integrals on (possibly cut) prisms. Especially for the spatially three dimensional case this is challenging and non-standard. A solution strategy for the numerical integration on those prisms is presented and discussed in chapter 4. In the remainder of this chapter (except for the numerical examples in section 3.4) we assume that space-time integrals can be computed exactly.*

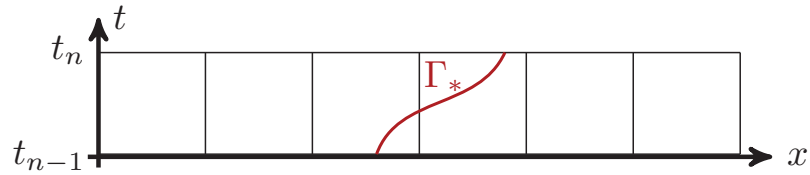


Figure 3.2.3: Sketch of space-time slab and a space-time interface Γ_*

Remark 3.2.6 (Petrov-Galerkin DG formulations). *In space-time formulations it is, in general, not necessary to choose the same test as ansatz space, cf. remark 3.2.1. In combination with XFEM it is however not clear how a well-posed Petrov-Galerkin version of the presented XFEM discretization can be achieved. We explain the problem. Consider a test space with piecewise constants in time W^0 and the ansatz space with piecewise linears in time W^1 . Both are enriched with the XFEM approach resulting in spaces $W^{\Gamma,1}$*

and $W^{\Gamma,1}$. In figure 3.2.3 the cut elements change between the time steps. The resulting XFEM enrichment for W^1 and W^0 results in six additional degrees of freedoms in the interface region for $W^{\Gamma,1}$ and only three for $W^{\Gamma,0}$. Hence, the total number of unknowns of test and ansatz space do not match. A possible remedy could be an enrichment of $W^{\Gamma,0}$ by discontinuous (in time) piecewise linear XFEM functions in the interface region. We did not investigate this further.

Remark 3.2.7 (Weightings in the average). From a computational point of view the suggested average in (3.2.14) is computationally inconvenient as in a space-time implementation of the cut elements (prisms) integration is done on space-time geometries. The evaluation of $\kappa_i(t)$, however, needs geometry information of a slice through a space-time geometry. Within each cut element $\kappa_i(t)$ has to be evaluated at several space-time integration points. A more “natural” weighting adjusts the weights to the space-time measures:

$$\kappa_i|_{\mathcal{Q}^T} := \frac{|\mathcal{Q}_i^T|}{|\mathcal{Q}^T|}. \quad (3.2.23)$$

Hence, κ_i is constant within each space-time element (prism). Note that the measures $|\mathcal{Q}_i^T|$ and $|\mathcal{Q}^T|$ have to be computed anyway, that means that this choice is computationally cheap. This choice seems to be suitable in practice. Stability, however, is proven more easily for the time-dependent weight.

Remark 3.2.8 (Non-symmetric Nitsche variants). Instead of the symmetric Nitsche formulation $N(u, v) = N_c(u, v) + N_c(v, u) + N_s(u, v)$ one can also use nonsymmetric or incomplete formulations as $N_{ns}(u, v) = N_c(u, v) - N_c(v, u) + N_s(u, v)$ or $N_{inc}(u, v) = N_c(u, v) + N_s(u, v)$. For both modifications the stability analysis discussed later also applies. For the nonsymmetric formulation the analysis is actually simpler as $N_{ns}(u, u) = N_s(u, u)$. Note that the loss of symmetry also implies a lack of adjoint consistency. In the remainder we thus only consider the symmetric formulation.

3.2.4.2 A first order space-time DG formulation

An implementation of the aforementioned discretization requires special solution strategies for the numerical integration on intersected space-time prisms $\mathcal{Q}_i^T$ and $\Gamma_*^n \cap \mathcal{Q}^T$, cf. chapter 4. A first order accurate version can be achieved in such a way that only numerical integration in space on the time levels t_n needs to be implemented. We discuss this version based on the formulation (3.2.18).

We use (time slab-) piecewise constant finite elements in time ($m = 0$ in (3.2.2)) and denote this space as $W^{\Gamma,0}$. Then for $v \in W^{\Gamma,0}$ we have $d'(u, v) = -(u, \mathbf{w} \cdot \nabla v)_{\mathcal{Q}_{1,2}^n}$ as $\partial_t v = 0$. Further we apply numerical quadrature in time with the θ -rule ($\theta \in (0, 1)$)

$$\begin{aligned} \sum_{i=1,2} \int_{t_{n-1}}^{t_n} \int_{\Omega_i(t)} f \, d\mathbf{x} \, dt &\approx I(\Omega_{1,2}^{n-1}, \Omega_{1,2}^n, \theta; f) \\ &:= \theta \Delta t \sum_{i=1,2} \int_{\Omega_i^n} f(\mathbf{x}, t_n) \, d\mathbf{x} + (1 - \theta) \Delta t \sum_{i=1,2} \int_{\Omega_i^{n-1}} f(\mathbf{x}, t_{n-1}) \, d\mathbf{x} \end{aligned} \quad (3.2.24)$$

3 Mass transport through a moving interface

and use the notation $(u, v)_{\Omega_{1,2}^\theta} := I(\Omega_{1,2}^{n-1}, \Omega_{1,2}^n, \theta; \beta uv)$. This results in the scheme

$$\begin{aligned} (u, v)_{\Omega_{1,2}^n} - (u^{n-1}, v)_{\Omega_{1,2}^{n-1}} - (u, \mathbf{w} \cdot \nabla v)_{\Omega_{1,2}^\theta} \\ + (\alpha \nabla u, \nabla v)_{\Omega_{1,2}^\theta} + N^\theta(u, v) = (f, v)_{\Omega_{1,2}^\theta} \end{aligned} \quad (3.2.25)$$

with $N^\theta(u, v) = \theta N^n(u, v) + (1-\theta)N^{n-1}(u, v)$ with $N^i(u, v) = N_c(u, v) + N_c(v, u) + N_s(u, v)$ as in (2.2.14) for $\Gamma = \Gamma(t_i)$. A similar method has been derived and used for a simple osmosis problem in the master thesis [Sch12]. Note that for the construction of $W^{\Gamma,0}$ within one time slab the interface position not only at t_n , but also at t_{n-1} is important, independent on the choice of θ . Note further, that in order to get a regular system matrix $\theta = 1$ is not allowed. This is due to the fact that for $\theta = 1$ it can happen that for some extended ansatz functions the contribution to (3.2.25) vanishes.

As addressed during the discussion of the method of lines approach in section 3.1.2 the quantity βu is continuous in time (but does not have a higher regularity). For the purpose of deriving a first order method one could also apply the method of lines on $\tilde{u} = \beta u$ in problem 1.2.1 resulting in (after the substitution $u = \beta^{-1}\tilde{u}$ and a restriction to the implicit Euler discretization)

$$\begin{aligned} \left(\frac{\beta^n u^n - \beta^{n-1} u^{n-1}}{\Delta t}, v \right)_{L^2(\Omega)} + (\mathbf{w}^n \nabla \beta^n u^n, v)_{L^2(\Omega_{1,2}^n)} \\ + (\alpha^n \nabla(\beta^n u^n), \nabla v)_{L^2(\Omega_{1,2}^n)} + N^n(u, v) = (\beta^n f^n, v)_{L^2(\Omega)} \end{aligned} \quad (3.2.26)$$

The most important difference between the formulations (3.2.25) and (3.2.26) is in the coupling between u^{n-1} and v . For the space-time method v in this coupling term is evaluated at Ω^{n-1} , for the method of lines v is evaluated at Ω^n . Both methods are supposed to be first order accurate. The difference between both methods becomes more evident if we consider mass balance. Therefore we set $v = \beta^{-1}$ in (3.2.25) and $v = 1$ in (3.2.26). For simplicity we assume $\mathbf{w}$ is constant in time and $f = 0$. We get

$$\int_{\Omega_{1,2}^n} u^n = \int_{\Omega_{1,2}^{n-1}} u^{n-1} d\mathbf{x} \quad (3.2.27)$$

for the space time method and

$$\int_{\Omega_{1,2}^n} \beta^n u^n - \int_{\Omega_{1,2}^{n-1}} \beta^{n-1} u^{n-1} d\mathbf{x} = -\Delta t \int_{\Gamma^n} \mathbf{w} \cdot \mathbf{n} \llbracket \beta u \rrbracket ds. \quad (3.2.28)$$

for the method of lines. While (3.2.27) directly describes a mass balance property for u independent of the discretization parameter Δt , (3.2.27) describes only a discrete version of the Reynolds transport theorem for the quantity βu . Hence, the method in (3.2.25) gives a (globally) conservative formulation, but (3.2.26) does not. This is because for the method in (3.2.25) the physical quantity u is conserved, whereas for the method in (3.2.26) the artificial quantity $\tilde{u} = \beta u$ fulfills an approximate balance laws only.

3.2.4.3 Convection stabilization for the Space-Time-DG Nitsche-XFEM method

We briefly discuss how standard convection stabilization ideas can be included in the Space-Time-DG Nitsche-XFEM method. For a one-phase problem a discontinuous-in-time Galerkin approach stabilized with a Least-squares formulation has been considered for a pure convection problem in [DH03, Chapter 3.10.2]. For a convection-diffusion equation on time-dependent domains a space-time Galerkin/least-squares methods in combination with a characteristic Galerkin method has been proposed in [Pir92]. The approach presented in the following is based on similar ideas. Note that for the case of piecewise linear functions (in space), discretizations using Least-squares or Streamline Diffusion stabilizations coincide.

For the stabilization we introduce a locally weighted discrete variant of $(\cdot, \cdot)_0$:

$$(u, v)_{0,h} := \sum_{n=1}^N \sum_{i=1}^2 \sum_{Q^T \in \mathcal{T}_{n,*}} \beta_i \gamma_T \int_{Q_i^T} uv \, d\mathbf{x} dt = \sum_{n=1}^N \sum_{Q^T \in \mathcal{T}_{n,*}} \gamma_T (u, v)_{0,Q^T} \quad (3.2.29)$$

We introduce the Space-Time-DG Nitsche-XFEM discretization method with SD stabilization:

Find $u_h \in W^\Gamma$ such that

$$\begin{aligned} B(u_h, v_h) + (\partial_t u_h - \operatorname{div}(\varepsilon \nabla u_h) + \mathbf{w} \cdot \nabla u_h, \partial_t v_h + \mathbf{w} \cdot \nabla v_h)_{0,h} \\ = (f, v_h)_0 + (f, \partial_t v_h + \mathbf{w} \cdot \nabla v_h)_{0,h} \quad \text{for all } v_h \in V_h^\Gamma. \end{aligned} \quad (3.2.30)$$

In the case of linear-in-space approximations $\Delta u_h = 0$ and with the space-time velocity $\mathbf{w}^* = (1, \mathbf{w})$ and the space-time gradient $\nabla^* = (\partial_t, \nabla)$, we can write

$$\begin{aligned} B(u_h, v_h) + s^{\text{SD}}(u_h, v_h) \\ = (f, v_h)_0 + (f, \mathbf{w}^* \cdot \nabla^* v_h)_{0,h} \quad \text{for all } v_h \in V_h^\Gamma \end{aligned} \quad (3.2.31)$$

with

$$s^{\text{SD}}(u, v) := (\mathbf{w}^* \cdot \nabla^* u, \mathbf{w}^* \cdot \nabla^* v)_{0,h}. \quad (3.2.32)$$

For the stabilization parameter γ_T we consider the multidimensional generalization of the choice in [KA10]:

$$\gamma_T = \left(\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2 \|\mathbf{w}\|_{\infty, T}}{h_T} \right)^2 + 9 \left(\frac{4\varepsilon}{h_T^2} \right)^2 \right)^{-\frac{1}{2}} \quad (3.2.33)$$

An alternative approach to stabilize convection dominated problems in a space-time setting has also been considered in [AMTX11]. This topic is not investigated further within this thesis.

3.3 Error analysis of the Space-Time-DG Nitsche-XFEM method

In this section we want to analyze the space-time method presented in the previous section (without Streamline Diffusion stabilization). The major difficulty with the error analysis is the fact that the regularity of the problem is anisotropic, that is the regularity in time and space direction of a solution may be different. Hence, standard isotropic approaches in the analysis as they are common for elliptic problems have their limitations. However, useful estimates can still be derived from such approaches. In [Tho97] optimal order estimates for Discontinuous Galerkin formulations in time are derived. A crucial assumption in that book is that the finite element spaces do not change in time. This is not true for our Space-Time-DG Nitsche-XFEM method and prevents that the ideas in [Tho97] can be extended easily to the problems considered here.

In section 3.3.1 we summarize important assumptions used in the error analysis in this section and recall regularity properties of the considered anisotropic Sobolev spaces.

The interpolation of space-time functions into the space-time finite element space W or the extended space-time finite element space W^Γ is not standard. For space dimension $d = 3$ there no longer holds $H^2(Q) \subset C^0(Q)$ and thus point evaluations are not possible. Hence, a standard nodal interpolation is not applicable anymore and interpolation operators with weaker regularity requirements have to be used. In section 3.3.2 we introduce a space-time interpolation operator based on the concatenation of standard spatial and temporal (quasi-)interpolations.

A first error analysis of the Space-Time-DG Nitsche-XFEM method has been presented in [LR13]. The most important results of that paper are presented in section 3.3.3. A second order in space and time error bound in the weak -1 -norm is shown.

3.3.1 Regularity statements and assumptions

We recall the definitions of the t -anisotropic Sobolev spaces introduced in section 3.1.1 and discuss a few important regularity properties for some of those spaces. Afterwards we summarize assumptions used in the error analysis.

3.3.1.1 Regularity of t -anisotropic Sobolev spaces

Under (mild) regularity assumptions on the interface Γ_* functions $u \in H_0^{2,1}(Q_1^n \cup Q_2^n)$ have well-defined traces $u|_{\Gamma_*^n}$ and $(\mathbf{n} \cdot \nabla u)|_{\Gamma_*^n}$ in $L^2(\Gamma_*^n)$, cf. [Lio57, WYW06]. The bilinear forms of the Space-Time-DG Nitsche-XFEM discretization are well-defined on $H_0^{2,1}(Q_{1,2}^n)$.

In the analysis in the following sections we need space-time trace operators and a space-time Poincare-Friedrichs inequality. Under mild smoothness conditions on Γ_* the

existence of a bounded trace operator $H^{1,0}(Q_i) \rightarrow L^2(\Gamma_*)$ follows from [Lio57]. In the remainder we assume that there exists such a trace operator that is bounded:

$$\|u_i\|_{0,\Gamma_*} \leq c(|u|_{1,Q_{1,2}} + \|u\|_{0,Q}) \quad \text{for all } u \in H_0^{1,0}(Q_{1,2}), \quad i = 1, 2 \quad (3.3.1)$$

with $u_i = u|_{Q_i}$. Furthermore, from standard results it follows that there is a (time) trace operator $H^{0,1}(Q_{1,2}) \rightarrow L^2(\Omega)$ that is bounded:

$$\|u(\cdot, t_n)\|_{0,\Omega} \leq c\|u\|_{H^{0,1}(Q_{1,2}^n)} \quad \text{for all } u \in H^{0,1}(Q_{1,2}^n), \quad n = 1, \dots, N. \quad (3.3.2)$$

With respect to the Poincare-Friedrichs inequality we first consider a fixed $t \in [0, T]$. From the Petree-Tartar theorem it follows, cf. [EG04, lemma B.63], that there exists a constant $c(t)$ such that for all $t \in I$ there holds

$$\|u\|_{0,\Omega} \leq c(t)(|u|_{1,\Omega_1 \cup \Omega_2} + \|[\![\beta u]\!]\|_{0,\Gamma(t)}) \quad \text{for all } u \in H_0^1(\Omega_1(t) \cup \Omega_2(t)). \quad (3.3.3)$$

In the remainder we assume that there exists a constant c_P such that

$$\|u\|_{0,Q} \leq c_P(|u|_{1,Q_{1,2}} + \|[\![\beta u]\!]\|_{0,\Gamma_*}) \quad \text{for all } u \in H_0^{1,0}(Q_{1,2}) \quad (3.3.4)$$

holds. Note that this follows from the result in (3.3.3) if $c(t)$ is uniformly bounded with respect to $t \in [0, T]$. We expect that such a uniform boundedness result holds if the space-time interface Γ_* is sufficiently smooth.

3.3.1.2 Assumptions

We summarize *the main assumptions that are used in the analysis*.

We restrict to the case of moderate jumps in the coefficients α, β and further only consider problems with moderate convection.

Assumption 3.3.1 (Moderate coefficients). *The coefficients α_i, β_i and $\mathbf{w}$ in the problem should be such that we have a diffusion dominated problem and that there are no singular perturbation effects caused by the Henry coefficient β . More precisely, we assume*

$$0 < c_L \leq \alpha_i, \beta_i \quad \text{and} \quad \alpha_i, \beta_i, \|\mathbf{w}\|_{L^\infty(\Omega)} \leq c_U, \quad i = 1, 2, \quad (3.3.5)$$

with constants c_L and c_U that are of order one.

For simplicity, in the analysis, we restrict to *quasi-uniform* meshes and equidistant time intervals.

Assumption 3.3.2 (Time step size and mesh size). *We assume $\Delta t = t_n - t_{n-1}$ is fixed. Further, the shape regular triangulations $\mathcal{T}_n$ are assumed to be quasi-uniform and all constants related to shape-regularity and quasi-uniformity are assumed to be uniformly bounded both with respect to the spatial mesh size h_n and with respect to Δt (i.e., with respect to N). For simplicity we further assume that the mesh size h_n is uniformly bounded with respect to the time step n , i.e. there exists h , s.t. $h \simeq h_n \simeq h_T$.*

We assume that the interface is smooth, resolved by the mesh and the following holds.

Assumption 3.3.3 (Smooth interface and trace operations). *The trace inequality (3.3.1) and the Poincare-Friedrichs inequality (3.3.4) are assumed to hold. We assume further that the space-time interface Γ_* is a smooth d -manifold and the space-time triangulation is sufficiently fine such that it can resolve Γ_* . More precisely, for each $\mathcal{Q}^T = T \times I_n$, with $T \in \mathcal{T}_n$ and $\Gamma_\omega := \mathcal{Q}^T \cap \Gamma_* \neq \emptyset$, we assume that there is a local orthogonal coordinate system $y = (z, \theta)$, $z \in \mathbb{R}^d$, $\theta \in \mathbb{R}$ such that Γ_ω is the graph of a smooth scalar function, say g , i.e. $\Gamma_\omega = \{(z, g(z)) \mid z \in U \subset \mathbb{R}^d\}$, with $\|\nabla g\|$ uniformly bounded.*

Assumption 3.3.4 ($H^2(Q_{1,2})$ -regularity). *We assume that given initial data $u^0 \in H^1(\Omega_{1,2})$ the solution u to problem 3.1.1 is in $H^2(Q_{1,2})$ and there holds*

$$\|u\|_{H^2(Q_{1,2})} \leq c \|u\|_{H^1(\Omega_{1,2})}.$$

Remark 3.3.1. *Assumption 3.3.4 is needed to apply duality arguments when proving error estimates in the weak $\|\cdot\|_{-1,T}$ -norm. It is however not directly clear if assumption 3.3.4 can be justified in general.*

Assumption 3.3.5. *For the refinement in space and time we consider one constraint on the ratio of temporal and spatial resolution within the refinement process. We assume that there exists a constant c such that there holds $h^2 \leq c\Delta t$.*

The constants denoted with c used in the results derived below are all independent of λ , $\bar{\alpha}$, h , $|\mathbf{w}|_\infty$, and of how the space-time interface Γ_* intersects the triangulation $\mathcal{T}_{n,*}$ (i.e. of the shape regularity of $\mathcal{Q}_i^T$).

3.3.2 Interpolation in space-time

The interpolation of space-time functions into the space-time finite element space W or the extended space-time finite element space W^Γ is not standard. For space dimension $d = 3$ there no longer holds $H^2(Q) \subset C^0(Q)$ and thus point evaluations are not possible. Hence, a standard nodal interpolation is not applicable anymore and interpolation operators with weaker regularity requirements have to be used.

We first discuss the interpolation on W in section 3.3.2.1 and extend it to W^Γ in section 3.3.2.2.

A nice property of the Space-Time-DG Nitsche-XFEM method is that it allows to use *non-matching* simplicial triangulations $\mathcal{T}_n$ between the time slabs I_n , $n = 1, \dots, N$, i.e. $\mathcal{T}_n \neq \mathcal{T}_m$ for $n \neq m$. Note that the interpolation applied in this section also allows for such a general setting.

3.3.2.1 Interpolation on W

We fix one time interval I_n and consider the interpolation of a function $u \in H^2(Q^n)$ into W_n . In the following we define and analyze an interpolation operator $\Pi_W : H^2(Q^n) \rightarrow$

W_n as the combination of standard spatial and a temporal interpolation operators $I^s : H^1(\Omega) \rightarrow V_h$ and $I^t : L^2(I_n) \rightarrow \mathcal{P}^k$ with $\mathcal{P}^k$ the space of polynomials up to degree k on I_n . We have to generalize both operators to deal with the space-time domain and the according function spaces.

This subsection is concerned with the space-time interpolation *without* an interface. We thus consider $\beta = 1$, such that $\|\cdot\|_S = \|\cdot\|_{0,S} = \|\cdot\|_{L^2(S)}$, $S \in \{\Omega, Q^n, Q\}$. As in the previous section W is the finite element space of (continuous in space, discontinuous in time) piecewise linear functions in space and time. For ease of presentation we restrict to the discussion of the interpolation into that space, but note, that extensions to higher order follow the same lines.

Semi-discrete spaces. For the temporal and spatial interpolation in the space-time context we introduce special tensor-product spaces, which we define next. Let X be any subspace of $L^2(\Omega)$. We use the following notation for the tensor-product between the space of linear functions in time and functions in X as

$$X \otimes \mathcal{P}^1 := \{v \in L^2(Q^n) | v(\mathbf{x}, t) = v_1(\mathbf{x}) + tv_2(\mathbf{x}); v_1, v_2 \in X\}. \quad (3.3.6)$$

For the space-time generalization of the spatial interpolation we use another tensor-product space. Let Y be a subspace of $L^2(I_n)$. We consider the space

$$V_h \otimes Y := \{v \in L^2(Q^n) | v(\mathbf{x}, t) = \sum_{i=1}^N v_i(t) \varphi_i(\mathbf{x}), v_i \in Y, N = \dim(V_h)\}. \quad (3.3.7)$$

Note that for the special cases $X = V_h$ in (3.3.6) and $Y = \mathcal{P}^1$ in (3.3.7), we have $V_h \otimes \mathcal{P}^1 = W$.

Interpolation in time. For $u \in L^2(I_n; X)$ we define the temporal interpolation $\Pi^t(u)$ as the $L^2(I_n; X)$ -projection of u into $X \otimes \mathcal{P}^1$, so that $\Pi^t(u)$ solves

$$(\Pi^t(u) - u, v)_{Q^n} = 0, \quad \text{for all } v \in X \otimes \mathcal{P}^1. \quad (3.3.8)$$

In the following we consider the general case $X = L^2(\Omega)$. One important tool in the analysis of this projection operator is the following equivalence lemma.

Lemma 3.3.1 (equivalence lemma).

Let E_1 be a Banach space, E_2, E_3 normed spaces (with $\|\cdot\|_j$ denoting the norm of E_j), and let $A \in \mathcal{L}(E_1, E_2)$, $B \in \mathcal{L}(E_1, E_3)$ such that one has:

- (a) $\|u\|_1 \simeq \|Au\|_2 + \|Bu\|_3$
- (b) B is compact.

Then one has the following properties:

- (i) The kernel of A is finite dimensional.
- (ii) The range of A is closed.
- (iii) There exists a constant C_0 such that if F is a normed space and $L \in \mathcal{L}(E_1, F)$ satisfies $Lu = 0$ whenever $Au = 0$, then one has $\|Lu\|_F \leq C_0 \|L\| \|Au\|_2$ for all $u \in E_1$.
- (iv) If G is a normed space and $M \in \mathcal{L}(E_1, G)$ satisfies $Mu \neq 0$ whenever $Au = 0$ and $u \neq 0$, then $\|u\|_1 \simeq \|Au\|_2 + \|Mu\|_G$.

Proof. See [Tar07, chapter 11]. □

The $L^2(Q^n)$ -projection on time-discrete spaces is stable.

Lemma 3.3.2 (Stability of L^2 -projection in time).

For the L^2 -projection there hold the stability results

$$\|\Pi^t u\|_{Q^n} \leq \|u\|_{Q^n}, \quad \forall u \in L^2(Q^n), \quad (3.3.9)$$

$$\|\partial_t \Pi^t u\|_{Q^n} \leq c \|\partial_t u\|_{Q^n}, \quad \forall u \in H^{0,1}(Q^n). \quad (3.3.10)$$

Proof. The proof of (3.3.9) is standard:

$$\begin{aligned} 0 &\leq \int_{I_n} (\Pi^t u - u, \Pi^t u - u)_\Omega dt = \int_{I_n} (\Pi^t u, \Pi^t u)_\Omega + (u, u)_\Omega dt - 2 \underbrace{\int_{I_n} (\Pi^t u, u)_\Omega dt}_{=\int_{I_n} (\Pi^t u, \Pi^t u)_\Omega dt} \\ &= \int_{I_n} -(\Pi^t u, \Pi^t u)_\Omega + (u, u)_\Omega dt \quad \Rightarrow \quad \|\Pi^t u\|_{Q^n} \leq \|u\|_{Q^n} \end{aligned}$$

By scaling arguments it suffices to proof (3.3.10) on the reference interval. We transform the time interval $(t_{n-1}, t_n]$ to the unit interval $\hat{I} = (0, 1]$ and consider the problem there. The corresponding transformation is denoted by $\Phi : \hat{Q} \rightarrow Q^n$ with $\hat{Q} := \Omega \times (0, 1]$ and transformed functions and spaces by $\hat{u} = u \circ \Phi$ and $\hat{W} = W \circ \Phi$. We further define $\hat{\Pi}^t = \Phi^{-1} \circ \Pi^t \circ \Phi$. For the proof we divide the L^2 -projection into $\hat{\Pi}^t = \hat{\Pi}_0^t + \hat{\Pi}_1^t$ with the L^2 -projection onto constant-in-time functions $\hat{\Pi}_0^t : L^2(\hat{Q}) \rightarrow L^2(\Omega) \otimes \mathcal{P}^0$ and $\hat{\Pi}_1^t : L^2(\hat{Q}) \rightarrow L^2(\Omega) \otimes \mathcal{P}^{0\perp}$ with $\mathcal{P}^{0\perp}$ the orthogonal complement of $\mathcal{P}^0$ in $\mathcal{P}^1$.

Then using the elementary estimate $\|\partial_t(a + bt)\|_{L^2((0,1))} \leq c\|a + bt\|_{L^2((0,1))}$ we have

$$\|\partial_t \hat{\Pi}^t \hat{u}\|_{\hat{Q}} = \|\partial_t \hat{\Pi}_1^t \hat{u}\|_{\hat{Q}} \leq c \|\hat{\Pi}^t(\hat{u} - \hat{\Pi}_0^t \hat{u})\|_{\hat{Q}} \leq c \|\hat{u} - \hat{\Pi}_0^t \hat{u}\|_{\hat{Q}} \leq c \|\partial_t \hat{u}\|_{\hat{Q}} \quad (3.3.11)$$

where in the last step we used lemma 3.3.1 (property (iii)) with $E_1 = F = H^{0,1}(\hat{Q})$, $E_2 = E_3 = L^2(\hat{Q})$, $A = \partial_t$, $B = \text{id}$, $L = \text{id} - \hat{\Pi}_0^t$. Note that continuity of L in $H^{0,1}(\hat{Q})$ is obvious due to $\partial_t \hat{\Pi}_0^t \hat{u} = 0$, $\forall \hat{u} \in L^2(\hat{Q})$. Transforming back to Q^n gives (3.3.10). $\square$

Let $\mathcal{L}^t := \text{id} - \Pi^t$ denote the temporal interpolation error. In preparation of estimating the interpolation error of the operator Π^t we give a special version of the well-known Bramble-Hilbert lemma.

Lemma 3.3.3 (Modified Bramble-Hilbert lemma).

The operator $\mathcal{L}^t : H^k(I_n, L^2(\Omega)) \rightarrow L^2(Q^n)$ is continuous for $k \geq 0$ and there holds $\mathcal{L}^t q = 0$ for $q \in L^2(\Omega) \otimes \mathcal{P}^1$. Further there holds

$$\|\mathcal{L}^t u\|_{Q^n} \leq c \|\partial_t^1 u\|_{Q^n} \quad \forall u \in H^{0,1}(Q^n), \quad (3.3.12)$$

$$\|\mathcal{L}^t u\|_{Q^n} + \|\partial_t \mathcal{L}^t u\|_{Q^n} \leq c \|\partial_t^2 u\|_{Q^n} \quad \forall u \in H^{0,2}(Q^n). \quad (3.3.13)$$

Proof. Both estimates follow with lemma 3.3.1 (property (iii)). For (3.3.12) consider $E_1 = H^{0,1}(Q^n)$, $E_2 = E_3 = F = L^2(Q^n)$, $A = \partial_t$, $B = \text{id}$, $L = \mathcal{L}^t$. The continuity of $\mathcal{L}^t$ follows with (3.3.10). Note further that $\ker(A) = L^2(\Omega) \otimes \mathcal{P}^0 \subset L^2(\Omega) \otimes \mathcal{P}^1$. For the second equation we use $E_1 = H^{0,2}(Q^n)$, $E_2 = L^2(Q^n)$, $E_3 = F = H^1(I_n, L^2(\Omega))$, $A = \partial_t^2$, $B = \text{id}$, $L = \mathcal{L}^t$. Here, we have $\ker(A) = L^2(\Omega) \otimes \mathcal{P}^1$ and continuity of $\mathcal{L}^t$ follows from (3.3.9) and (3.3.10). $\square$

This results in the following lemma.

Lemma 3.3.4 (Error estimates for temporal interpolation).

There holds

$$\|\mathcal{L}^t u\|_{Q^n} \leq c \Delta t^l \|u\|_{H^{0,l}(Q^n)}, \quad \forall u \in H^{0,l}(Q^n), \quad l = 1, 2, \quad (3.3.14a)$$

$$\|\partial_t \mathcal{L}^t u\|_{Q^n} \leq c \Delta t \|u\|_{H^{0,2}(Q^n)}, \quad \forall u \in H^{0,2}(Q^n). \quad (3.3.14b)$$

Proof. The claim follows from the preceding lemma and standard scaling arguments. Consider the transformation as in the proof of lemma 3.3.2 and define $\hat{\mathcal{L}}^t = \Phi^{-1} \circ \mathcal{L}^t \circ \Phi$. Then combining transformation rules and the results of lemma 3.3.3 we get

$$\begin{aligned} \|\mathcal{L}^t u\|_{Q^n}^2 &\leq \Delta t \|\hat{\mathcal{L}}^t \hat{u}\|_{\hat{Q}}^2 \leq c \Delta t \|\partial_t \hat{u}\|_{\hat{Q}}^2 \leq c \Delta t^2 \|\partial_t u\|_{Q^n}^2, \\ \|\mathcal{L}^t u\|_{Q^n}^2 &\leq \Delta t \|\hat{\mathcal{L}}^t \hat{u}\|_{\hat{Q}}^2 \leq c \Delta t \|\partial_t^2 \hat{u}\|_{\hat{Q}}^2 \leq c \Delta t^4 \|\partial_t^2 u\|_{Q^n}^2, \\ \|\partial_t \mathcal{L}^t u\|_{Q^n}^2 &\leq \Delta t^{-1} \|\partial_t \hat{\mathcal{L}}^t \hat{u}\|_{\hat{Q}}^2 \leq c \Delta t^{-1} \|\partial_t^2 \hat{u}\|_{\hat{Q}}^2 \leq c \Delta t^2 \|\partial_t^2 u\|_{Q^n}^2. \end{aligned}$$

$\square$

Interpolation in space. For the spatial interpolation operator different choices are possible. In this work we again consider an $L^2(\Omega)$ -projection. Possible other choices are the Clément (quasi-)interpolation operator, cf. the original paper [Cl675] or similar approaches [SZ90].

We define the projector $\Pi^s : L^2(Q^n) \rightarrow V_h \otimes Y$, so that for $u \in L^2(Q^n)$

$$(\Pi^s(u) - u, v)_{Q^n} = 0, \quad \text{for all } v \in V_h \otimes Y \quad (3.3.15)$$

In the following we consider the general case $Y = L^2(I_n)$. The introduced interpolation operator is a generalization of the standard spatial $L^2(\Omega)$ (quasi-) interpolation. For sufficiently smooth functions a spatial $L^2(\Omega)$ projection $I^s : L^2(\Omega) \rightarrow V_h$ for each $t \in I_n$ coincides with Π^s .

Lemma 3.3.5.

Let $I^s : L^2(\Omega) \rightarrow V_h$ be the spatial $L^2(\Omega)$ (quasi-) interpolation operator. Then there holds for $u \in L^2(\Omega) \otimes C^0(I_n) = C^0(I_n; L^2(\Omega))$

$$\Pi^s u = I^s u \text{ in the } L^2(Q^n)\text{-sense.} \quad (3.3.16)$$

Further for $u \in H^{0,1}(Q^n)$ there holds

$$\Pi^s \partial_t u = \partial_t \Pi^s u \in L^2(Q^n). \quad (3.3.17)$$

Proof. We define $I^s : L^2(\Omega) \rightarrow V_h$ as the $L^2(\Omega)$ projection

$$(I^s u, v_h)_\Omega = (u, v_h)_\Omega \quad \forall v_h \in V_h$$

Let $\{\varphi_i\}_{1,\dots,N}$, $N = \dim(V_h)$ be a basis of V_h and let $\{\psi_i\}_{1,\dots,N}$ be the dual basis, s.t. $(\phi_i, \psi_j)_{L^2(\Omega)} = \delta_{ij}$, $i, j = 1, \dots, N$. Then we have the following explicit description of the interpolation operator:

$$I^s u = \sum_{i=1}^N (u, \psi_i) \varphi_i \in V_h \quad (3.3.18)$$

For sufficiently smooth u (s.t. $u(\cdot, t) \in L^2(\Omega)$, i.e. $u \in L^2(\Omega) \otimes C^0(I_n)$), for instance $u \in H^{0,1}(Q^n)$, we claim that (3.3.16) holds. To see this, note that every function v_h in $V_h \otimes L^2(I_n)$ can be written as $v_h(\mathbf{x}, t) = \sum_{i=1}^N \psi_i(\mathbf{x}) a_i(t)$, $a_i \in L^2(I_n)$, $i = 1, \dots, N$ and thus we have

$$(I^s u, v_h)_{Q^n} = \sum_{i=1}^N \int_{I_n} (I^s u(\cdot, t), \psi_i)_\Omega a_i(t) dt = \sum_{i=1}^N \int_{I_n} (u(\cdot, t), \psi_i)_\Omega a_i(t) dt = (u, v_h)_{Q^n}.$$

Comparing this with (3.3.15) we conclude that (3.3.16) holds.

If further $u \in L^2(\Omega) \otimes C^1(I_n)$, there follows with (3.3.18), $I^s \partial_t u = \partial_t I^s u$ and (3.3.16) that (3.3.17) holds. Now, if $u \notin L^2(\Omega) \otimes C^1(I_n)$, but $u \in H^{0,1}(Q^n)$, we have by definition of the weak derivative $w = \partial_t u \in L^2(Q^n)$ that there holds

$$(w, v)_{Q^n} = -(u, \partial_t v)_{Q^n} \quad \forall v \in C_0^\infty(Q^n).$$

With $\hat{w} = \partial_t \Pi^s u$ it follows

$$\begin{aligned} (\hat{w}, v)_{Q^n} &= -(\Pi^s u, \partial_t v)_{Q^n} = -(u, \Pi^s \partial_t v)_{Q^n} \\ &= -(u, \partial_t \Pi^s v)_{Q^n} = (w, \Pi^s v)_{Q^n} = (\Pi^s w, v)_{Q^n} \quad \forall v \in C_0^\infty(Q^n). \end{aligned}$$

With the du Bois-Reymond lemma we conclude $\hat{w} = \Pi^s w$ a.e. in Q^n and thus $\partial_t \Pi^s u = \Pi^s \partial_t u$ (in the $L^2(Q^n)$ -sense). $\square$

The projection into the semi-discrete space is stable and has reasonable approximation properties. With $\mathcal{L}^s := \text{id} - \Pi^s$ we denote the spatial error operator corresponding to the spatial interpolation error.

Lemma 3.3.6.

For the projector $\Pi^s : L^2(Q^n) \rightarrow V_h \otimes L^2(\Omega)$ there hold the following stability and approximation results ($l = 1, 2$)

$$\|\Pi^s u\|_{Q^n} \leq \|u\|_{Q^n} \quad \forall u \in L^2(Q^n), \quad (3.3.19a)$$

$$\|\nabla \Pi^s u\|_{Q^n} \leq c \|\nabla u\|_{Q^n} \quad \forall u \in H^{1,0}(Q^n), \quad (3.3.19b)$$

$$\|\mathcal{L}^s u\|_{Q^n} \leq ch^l \|u\|_{L^2(I_n, H^l(\Omega))} \quad \forall u \in H^{l,0}(Q^n), \quad (3.3.19c)$$

$$\|\nabla \mathcal{L}^s u\|_{Q^n} \leq ch \|u\|_{L^2(I_n, H^2(\Omega))} \quad \forall u \in H^{2,0}(Q^n). \quad (3.3.19d)$$

Proof. The first estimate (3.3.19a) is a fundamental property of the L^2 -projection (cf. proof of lemma 3.3.2). We show (3.3.19b). There exists a smooth approximation $u^\varepsilon \in C^1(Q^n)$ to $u \in H^{1,0}(Q^n)$ such that $\|u^\varepsilon - u\|_{H^{1,0}(Q^n)} \rightarrow 0$ for $\varepsilon \rightarrow 0$. Then we have

$$\begin{aligned} \|\nabla \Pi^s u\|_{Q^n} &\leq \|\nabla \Pi^s(u - u^\varepsilon)\|_{Q^n} + \|\nabla \Pi^s u^\varepsilon\|_{Q^n} \\ &\leq \frac{c}{h} \|\Pi^s(u - u^\varepsilon)\|_{Q^n} + \|\nabla \Pi^s u^\varepsilon\|_{Q^n} \\ &\leq \frac{c}{h} \underbrace{\|u - u^\varepsilon\|_{Q^n}}_{\rightarrow 0} + \|\nabla \Pi^s u^\varepsilon\|_{Q^n} \end{aligned}$$

As u^ε is sufficiently smooth we have $u^\varepsilon(\cdot, t) \in H^1(\Omega)$ for every $t \in I_n$. Thus with the according stability estimate for the spatial interpolation I^s (due to the smoothness of u^ε we have $\Pi^s u^\varepsilon = I^s u^\varepsilon$, cf. lemma 3.3.5) we get

$$\begin{aligned} \|\nabla \Pi^s u^\varepsilon\|_{Q^n}^2 &= \int_{I_n} \|\nabla I^s u^\varepsilon(\cdot, t)\|_\Omega^2 dt \\ &\leq c \int_{I_n} \|\nabla u^\varepsilon(\cdot, t)\|_\Omega^2 dt = c \|\nabla u^\varepsilon\|_{Q^n}^2 \end{aligned}$$

The stability result of the spatial operator I^s can be found in the literature. For the case of quasi-uniform meshes and linear finite elements a simple proof is given in [BD81, appendix]. For the more general case (higher order and locally refined meshes) we refer to [BPS01, BY14, Kar13] and references therein. Setting $\varepsilon \rightarrow 0$ gives $\|\nabla u^\varepsilon\|_{Q^n} \rightarrow \|\nabla u\|_{Q^n}$

and thus (3.3.19b) follows. The other estimates follow the same lines using corresponding estimates for the spatial interpolator I^s . $\square$

Remark 3.3.2 (Clément interpolation). *The results in the last lemma essentially build on stability results for the spatial interpolation operator. Similar estimates can also be obtained for a space-time version of the Clément interpolation as the necessary stability estimates have also been proven for the Clément interpolation operator, see for instance [Clé75], [BF91] and [EG04].*

Interpolation in space-time. We concatenate spatial and temporal interpolation to define the space-time interpolation operator Π_W :

$$\Pi_W : L^2(Q^n) \rightarrow W_n, \quad \Pi_W(v) := \Pi^t(\Pi^s(v)), \quad \forall v \in L^2(Q^n) \quad (3.3.20)$$

Note that as Π^s and Π^t are L^2 -projections, the overall interpolation operator Π_W is an L^2 -projection on $L^2(Q^n)$.

For a function $u \in L^2(Q^n)$ with $\partial_t u \in L^2(Q^n)$, spatial interpolation and temporal derivatives commute as shown in lemma 3.3.5. Analogously, one can show that if $u \in L^2(Q^n)$ and $\nabla u \in L^2(Q^n)$ temporal interpolation and spatial derivatives commute and there holds $\nabla \Pi^t(u) = \Pi^t(\nabla u)$.

We now analyze the interpolation error of Π_W for which we also introduce the notation $\mathcal{L}_W := (\text{id} - \Pi_W)$.

Theorem 3.3.7.

For the interpolation operator Π_W as in (3.3.20) there holds the following approximation error bound ($l = 0, 1$)

$$\|u - \Pi_W u\|_{l, Q^n} \leq c(\Delta t^{2-l} + h^{2-l})\|u\|_{H^2(Q^n)}, \quad \forall u \in H^2(Q^n) \quad (3.3.21)$$

Proof. For a differential operator $D \in \{\text{id}, \nabla, \partial_t\}$ we have for $u \in H^2(Q^n)$

$$\begin{aligned} \|D\mathcal{L}_W u\|_{Q^n} &= \|D(u - \Pi^t \Pi^s u)\|_{Q^n} \\ &\leq \|D(u - \Pi^t u)\|_{Q^n} + \|D(\Pi^t u - \Pi^t \Pi^s u)\|_{Q^n} \\ &= \|D\mathcal{L}^t u\|_{Q^n} + \|D\Pi^t \mathcal{L}^s u\|_{Q^n} \\ &\leq \|D\mathcal{L}^t u\|_{Q^n} + c\|D\mathcal{L}^s u\|_{Q^n}. \end{aligned}$$

In the last step we used that for $D \in \{\text{id}, \nabla\}$ we have $D(\Pi^t w) = \Pi^t(Dw)$ for $Dw \in L^2(Q^n)$ and can apply (3.3.9). For $D = \partial_t$ we use (3.3.10). The remainder follows directly from the preceding estimates. $\square$

Note that the proof displays the anisotropic structure of the interpolation. Assume for instance we have $u \in H^{k,l}(Q^n)$. Then with the same technique we can easily show the

anisotropic result

$$\|u - \Pi_W u\|_{0,Q^n} \leq c(h^k + \Delta t^l) \|u\|_{H^{k,l}(Q^n)}, \quad \forall u \in H^{k,l}(Q^n), \quad k, l \in \{1, 2\}. \quad (3.3.22)$$

The previous estimates can also be used to bound the approximation error of the interpolation operator Π_W at discrete time levels.

Lemma 3.3.8 (Approximation error bounds at fixed time levels t_n).

For $u \in H^2(Q^n)$ there holds the estimate

$$\|(u - \Pi_W u)(\cdot, t_n)\|_\Omega \leq c(h^2 \Delta t^{-\frac{1}{2}} + \Delta t^{\frac{3}{2}}) \|u\|_{H^2(Q^n)}. \quad (3.3.23)$$

Proof. As in the proof of theorem 3.3.7 we consider the splitting $\mathcal{L}_W = \mathcal{L}^t + \Pi^t \mathcal{L}^s$ and with the triangle inequality we get

$$\|\mathcal{L}_W u(\cdot, t_n)\|_\Omega \leq \|\mathcal{L}^t u(\cdot, t_n)\|_\Omega + \|\Pi^t \mathcal{L}^s u(\cdot, t_n)\|_\Omega.$$

The argument in the last term is linear in time, a simple inverse inequality thus gives

$$\|\Pi^t \mathcal{L}^s u(\cdot, t_n)\|_\Omega \leq c \Delta t^{-\frac{1}{2}} \|\mathcal{L}^s u(\cdot, t_n)\|_{Q^n}.$$

Together with the previous bounds for the interpolation with Π^s we get

$$\|\Pi^t \mathcal{L}^s u(\cdot, t_n)\|_\Omega \leq c h^2 \Delta t^{-\frac{1}{2}} \|u\|_{H^2(Q^n)}.$$

Next, we consider the term $\|\mathcal{L}^t u(\cdot, t_n)\|_\Omega$. Transform the time interval $(t_{n-1}, t_n]$ to the unit interval $\hat{I} = (0, 1]$ and consider the problem there. The corresponding transformation is denoted by $\Phi : \hat{Q} \rightarrow Q^n$ with $\hat{Q} := \Omega \times (0, 1]$ and transformed functions and spaces by $\hat{u} = u \circ \Phi$ and $\hat{W} = W \circ \Phi$. We further define $\hat{\mathcal{L}}^t = \Phi^{-1} \circ \mathcal{L}^t \circ \Phi$. Then we have (using the continuity of the (time) trace operator in $H^{0,1}(\hat{Q})$ and (3.3.13))

$$\|\mathcal{L}^t u(\cdot, t_n)\|_\Omega = \|\hat{\mathcal{L}}^t \hat{u}(\cdot, 1)\|_\Omega \leq \|\hat{\mathcal{L}}^t \hat{u}\|_{H^{0,1}(\hat{Q})} \leq c \|\partial_t^2 \hat{u}\|_{\hat{Q}} \leq c \Delta t^{\frac{3}{2}} \|u\|_{H^2(Q^n)}.$$

□

So far we did not make use of assumption 3.3.5. If we do this the estimate (3.3.23) reads as

$$\|(u - \Pi_W u)(\cdot, t_n)\|_\Omega \leq c(h + \Delta t^{\frac{3}{2}}) \|u\|_{H^2(Q^n)}.$$

3.3.2.2 Interpolation on W^Γ

Based on the interpolation on W we can introduce the space-time XFEM interpolation into the space $W^\Gamma = \mathcal{R}_1 W + \mathcal{R}_2 W$. This is done in the same way as in the analysis of the spatial XFEM, cf. section 2.3.1.5 and [HH02, GR11]. We assume linear extension

operators $\mathcal{E}_i : H_0^2(Q_i) \rightarrow H_0^2(Q)$, $i = 1, 2$, that are bounded: $\|\mathcal{E}_i u\|_{2,Q} \leq c\|u\|_{2,Q_i}$ for all $u \in H_0^2(Q_i)$. The space-time XFEM interpolation operator $I_\Gamma : H_0^2(Q_{1,2}) \rightarrow W^\Gamma$ is given by

$$I_\Gamma := \mathcal{R}_1 \Pi_W \mathcal{E}_1 \mathcal{R}_1 + \mathcal{R}_2 \Pi_W \mathcal{E}_2 \mathcal{R}_2. \quad (3.3.24)$$

Due to the linear extension operators the results from the previous section extend naturally to the XFEM space:

Theorem 3.3.9.

For the interpolation operator I_Γ as in (3.3.24) there holds the following approximation error bound ($l = 0, 1$)

$$\|u - I_\Gamma u\|_{H^l(Q_{1,2}^n)} \leq c(\Delta t^{2-l} + h^{2-l})\|u\|_{H^2(Q_{1,2}^n)}, \quad \forall u \in H^2(Q_{1,2}^n), \quad (3.3.25)$$

$$\|(u - I_\Gamma u)(\cdot, t_n)\|_{\Omega_{1,2}} \leq c(h^2 \Delta t^{-\frac{1}{2}} + \Delta t^{\frac{3}{2}})\|u\|_{H^2(Q_{1,2}^n)}, \quad \forall u \in H^2(Q_{1,2}^n). \quad (3.3.26)$$

3.3.3 Error analysis for the mass transport problem with moving interface

The error analysis in this section divides into the definition of norms (section 3.3.3.1), the summary of important properties of the bilinear form $B(\cdot, \cdot)$ and interpolation bounds (section 3.3.3.2) and a priori bounds in the discrete norms (section 3.3.3.3) and a weaker norm (section 3.3.3.4). Most of the results are taken from [LR13]. The presentation is however slightly different, due to the use of different discrete norms. Further, the interpolation used and the assumption on the dual problem in section 3.3.3.4 are different. At the end of the section we comment on a comparison of this error analysis with the one in [Tho97] for a simpler problem.

3.3.3.1 Discrete norms

For the analysis of the discretization we define special norms. For the Nitsche bilinear form we introduce space-time variants of the norms used for the stationary case in chapter 2 (and [HH02]):

$$\|v\|_{\pm \frac{1}{2}, h, \Gamma_*^n}^2 := \int_{\Gamma_*^n} h_T^{-1} \nu v^2 ds, \quad \|v\|_{\pm \frac{1}{2}, h, \Gamma_*}^2 := \sum_{n=1}^N \|v\|_{\pm \frac{1}{2}, h, \Gamma_*^n}^2.$$

Note that due to assumption 3.3.2 we have $h_T \simeq h_n \simeq h$. Note further that $\int_{\Gamma_*^n} \nu v w ds \leq \|v\|_{\frac{1}{2}, h, \Gamma_*^n} \|w\|_{-\frac{1}{2}, h, \Gamma_*^n}$ holds. We define the important contributions to the discrete norms which are related to the Nitsche interface terms to

$$\|v\|_{\mathcal{N}}^2 := \|\llbracket \alpha \nabla v \cdot \mathbf{n} \rrbracket\|_{-\frac{1}{2}, h, \Gamma_*}^2 + \|\llbracket \beta v \rrbracket\|_{\frac{1}{2}, h, \Gamma_*}^2$$

We further define the discrete norms for $v \in W^\Gamma + H_0^{2,1}(Q)$

$$\begin{aligned} \|v\|^2 &:= |\sqrt{\alpha}v|_{1,Q_{1,2}}^2 + \|v\|_N^2 + \llbracket v \rrbracket^2, \\ \text{with } \llbracket v \rrbracket^2 &:= \sum_{n=1}^{N-1} \|\llbracket v \rrbracket^n\|_{\Omega_{1,2}^n}^2 + \|v_+^0\|_{\Omega_{1,2}^0}^2 + \|v_-^N\|_{\Omega_{1,2}^N}^2, \end{aligned} \quad (3.3.27)$$

$$\begin{aligned} \|v\|_*^2 &:= \|v\|_{1,Q_{1,2}}^2 + \|v\|_N^2 + \llbracket v \rrbracket_*^2, \\ \text{with } \llbracket v \rrbracket_*^2 &:= \sum_{n=1}^{N-1} \|v_+^n\|_{\Omega_{1,2}^n}^2 + \|v_+^0\|_{\Omega_{1,2}^0}^2 + \|v_-^N\|_{\Omega_{1,2}^N}^2. \end{aligned} \quad (3.3.28)$$

where $\|\cdot\|$ is tailored to proof stability and $\|\cdot\|_*$ to allow for continuity estimates. Note that $\|\cdot\|_{1,Q_{1,2}}^2$ contains time derivatives $\partial_t \cdot$, but $|\sqrt{\alpha} \cdot|_{1,Q_{1,2}}$ does not.

Further note that for $v_h \in W^\Gamma$ there holds (cf. the stability result in [LR13, Theorem 3.8])

$$|\sqrt{\alpha}v_h|_{1,Q_{1,2}}^2 + \|v_h\|_N^2 \simeq |\sqrt{\alpha}v_h|_{1,Q_{1,2}}^2 + \|\llbracket \beta v_h \rrbracket\|_{\frac{1}{2},h,\Gamma_*}^2 \quad (3.3.29)$$

and

$$\|v\| \leq c\|v\|_* \quad \forall u \in W^\Gamma + H_0^{2,1}(Q_{1,2}). \quad (3.3.30)$$

3.3.3.2 Consistency, stability, continuity and interpolation bounds

We quote (with adapted notation) the consistency, stability and boundedness results in the norms $\|\cdot\|$ and $\|\cdot\|_*$ from [LR13] and apply the interpolation bounds from section 3.3.2.

Lemma 3.3.10 (Consistency).

Let u be the solution of problem 3.1.1 and assume that $u \in H_0^{2,1}(Q_{1,2})$. Then

$$B(u, v_h) = f(v_h) \quad \text{for all } v_h \in W^\Gamma$$

holds, which implies for $u_h \in W^\Gamma$ the solution of (3.2.16)

$$B(u - u_h, v_h) = 0 \quad \text{for all } v_h \in W^\Gamma.$$

Proof. The consistency of the discretization has already been sketched in the derivation of the method in section 3.2.4.1. For details see [LR13, Theorem 3.4]. $\square$

Lemma 3.3.11 (Stability).

Define $g_B := \frac{1}{2} \min\{\alpha_1, \alpha_2, 1\}$. There exists a constant $c^* > 0$ independent of h_n , Δt and λ such that for all $\lambda > c^*$ the following holds:

$$B(v_h, v_h) \geq g_B \|v_h\|^2$$

for all $v_h \in W^\Gamma$.

Proof. See [LR13, Theorem 3.8]. $\square$

Remark 3.3.3 (Energy-stability). A consequence of the estimates in [LR13] is the energy-stability in the sense that $\frac{1}{2} \|u_-^n\|_{\Omega_{1,2}^n}^2 \leq \frac{1}{2} \|u_-^{n-1}\|_{\Omega_{1,2}^{n-1}}^2$ holds if there are no additional sources due to boundary conditions or volume sources f , c.f. remark 3.2.2.

Lemma 3.3.12 (Boundedness).

There exists a constant G_B , depending only on $\|\mathbf{w}\|_{\infty, Q}$ and $\alpha_i, \beta_i, i = 1, 2$, such that for all $u, v \in W^\Gamma + H_0^{2,1}(Q_{1,2})$ the following holds:

$$|B(u, v)| \leq G_B \|u\|_* \|v\| \quad (3.3.31a) \quad |B(v, u)| \leq G_B \|u\|_* \|v\| \quad (3.3.31b)$$

Proof. With only minor adaptations (3.3.31a) follows from [LR13, Theorem 3.12]. In [LR13, Theorem 3.12] the time jump term in $b(\cdot, \cdot)$ is only treated for arguments which are continuous in time. The slightly stronger norm $\|\cdot\|_*$ used in this work allows use a bound of the form $b(u, v) \leq \|u\| \|v\|_*$ also for u, v discontinuous in time. The estimate (3.3.31b) follows the same lines after rewriting $B(\cdot, \cdot)$ as $a(\cdot, \cdot) + d'(\cdot, \cdot) + b'(\cdot, \cdot) + N(\cdot, \cdot)$. $\square$

Lemma 3.3.13 (Interpolation in $\|\cdot\|$ and $\|\cdot\|_*$).

For $u \in H_0^2(Q_{1,2})$ and the interpolation operator I_Γ , there holds

$$\|u - I_\Gamma u\| \leq c(h + \Delta t) \|u\|_{2, Q_{1,2}} \text{ for all } u \in H_0^2(Q_{1,2}) \quad (3.3.32a)$$

$$\|u - I_\Gamma u\|_* \leq c(h + \Delta t) \|u\|_{2, Q_{1,2}} \text{ for all } u \in H_0^2(Q_{1,2}) \quad (3.3.32b)$$

Proof. The estimates within the space-time domain $Q_{1,2}$ and on the time levels t_n , $n = 0, \dots, N$ follow directly from the results in section 3.3.2. The approximation problem of the Nitsche interface terms can be reduced to approximation problems within the space-time domains, cf. [LR13, Theorem 3.11]. Note that we use assumption 3.3.5 to estimate $h^2 \Delta t^{-\frac{1}{2}} \leq ch$. $\square$

Remark 3.3.4. We comment on the optimality of the interpolation bounds in lemma 3.3.13. The bound in (3.3.32b) is of optimal order w.r.t. h and Δt as $\|\cdot\|_*$ contains full derivatives in space and time direction. This is not the case for the bound in (3.3.32a). As the time derivative is not part of the norm $\|\cdot\|$ the bound (3.3.32a) can be improved for sufficiently smooth functions.

3.3.3.3 Discretization error bounds in discrete norm

Putting the previous results together we get the following version of the famous Cea lemma.

Lemma 3.3.14.

Let u be a solution of problem 3.1.1 and assume that $u \in H_0^{2,1}(Q_{1,2})$. Further, let $u_h \in W^\Gamma$ be the solution of (3.2.16). There holds

$$\|u - u_h\| \leq \left(c + \frac{G_B}{g_B}\right) \inf_{v_h \in W^\Gamma} \|u - v_h\|_* \quad (3.3.33)$$

Proof. We split the error $u - u_h$ into the approximation error $e_a = u - v_h$ and the discrete error $e_d = v_h - u_h \in W^\Gamma$ for an arbitrary $v_h \in W^\Gamma$. For e_d we use lemma 3.3.10, lemma 3.3.11, and lemma 3.3.12 to get

$$\|e_d\|^2 \leq \frac{1}{g_B} B(e_d, e_d) = \frac{1}{g_B} B(e_a, e_d) \leq \frac{G_B}{g_B} \|e_a\|_* \|e_d\|. \quad (3.3.34)$$

Applying the triangle inequality for $u - u_h = e_a + e_d$ and using (3.3.30) gives the result. $\square$

Combining that result with the interpolation estimate in lemma 3.3.13 gives

Theorem 3.3.15.

Let u be a solution of (3.1.1a)-(3.1.1e) and assume that $u \in H_0^2(Q_{1,2})$. Further, let $u_h \in W^\Gamma$ be the solution of (3.2.16). There holds for a constant c , independent of h and Δt

$$\|u - u_h\| \leq c(h + \Delta t) \|u\|_{2,Q_{1,2}}. \quad (3.3.35)$$

Proof. Combine lemma 3.3.14 and lemma 3.3.13. $\square$

Remark 3.3.5. We comment on the main differences with the error bounds in [LR13]. For the error bound in theorem 3.3.15 an interpolation operator different from that in [LR13] is used. The interpolation operator is well-defined also for functions with low regularity ($L^2(Q_{1,2}^n)$) and allows for the use of grids which are non-matching across different time slabs. Further, in contrast to the estimate in [LR13], the bound is anisotropic in the with respect to h and Δt .

3.3.3.4 Duality arguments

In the following lemma we quote a result that will be used in a duality argument in theorem 3.3.17. To this end we define the homogeneous backward problem

Problem 3.3.1.

$$-\frac{\partial \hat{v}}{\partial t} - \mathbf{w} \cdot \nabla \hat{v} - \operatorname{div}(\alpha \nabla \hat{v}) = 0 \quad \text{in } \Omega_i(t), \quad i = 1, 2, \quad t \in [0, T], \quad (3.3.36a)$$

$$[[\alpha \nabla \hat{v} \cdot \mathbf{n}]] = 0 \quad \text{on } \Gamma(t), \quad (3.3.36b)$$

$$[[\beta \hat{v}]] = 0 \quad \text{on } \Gamma(t), \quad (3.3.36c)$$

$$\hat{v}(\cdot, T) = \hat{v}_T \quad \text{in } \Omega_i(T), \quad i = 1, 2, \quad (3.3.36d)$$

$$u(\cdot, t) = 0 \quad \text{on } \partial\Omega, \quad t \in [0, T]. \quad (3.3.36e)$$

Then the following adjoint consistency property holds.

Lemma 3.3.16.

Let $\hat{v}_T \in H^1(\Omega_{1,2}(T))$ with $(\hat{v}_T)|_{\partial\Omega} = 0$ be the given initial data to problem 3.3.1. Assume that problem 3.3.1 has a solution $\hat{v} \in H_0^{2,1}(Q_{1,2})$. This solution satisfies

$$B(w, \hat{v}) = (w^N, \hat{v}_T)_{0,T} \quad \text{for all } w \in W^\Gamma + H_0^{2,1}(Q_{1,2}). \quad (3.3.37)$$

Proof. See [LR13, lemma 3.14]. □

We recall the definition of the negative norm for functions $w \in L^2(\Omega_{1,2}(T))$

$$\|w\|_{-1,T} := \sup_{\hat{v} \in H_0^1(\Omega_{1,2}(T))} \frac{(w, \hat{v})_{0,T}}{\|\hat{v}_T\|_{1,T}}.$$

This norm is weaker than $\|\cdot\|_{0,T}$ and allows to prove higher order convergence using duality arguments.

Theorem 3.3.17.

Assume that (3.1.1a)-(3.1.1e) has a solution $u \in H_0^{2,1}(Q_{1,2})$. Under assumption 3.3.4 the homogeneous backward problem as in lemma 3.3.16 has a solution $\hat{v} \in H_0^2(Q_{1,2})$ that has the regularity property $\|\hat{v}\|_{2,Q_{1,2}} \leq c\|\hat{v}_T\|_{1,T}$ with a constant c independent of the initial data $\hat{v}_T \in H^1(\Omega_{1,2}^T)$. For the discretization error $u - u_h$ the following holds:

$$\|(u - u_h)(\cdot, T)\|_{-1,T} \leq c(h^2 + \Delta t^2)\|u\|_{2,Q_{1,2}}. \quad (3.3.38)$$

Proof. We aim to derive a bound of the form

$$((u - u_h)(\cdot, T), \hat{v}_T)_{0,T} \leq c(h + \Delta t) \|u - u_h\| \|\hat{v}_T\|_{1,T}$$

which holds true for every $\hat{v}_T \in H_0^1(\Omega_{1,2}(T))$. The error is denoted by $e = u - u_h$ and we set $\hat{v}_T = e(\cdot, T)$. The induced solution is denoted by $\hat{v}$. In (3.3.37) we take $w = e$. This yields, with $e_{\hat{v}} := \hat{v} - I_\Gamma \hat{v}$,

$$(e(\cdot, T), \hat{v}_T)_{0,T} = B(e, \hat{v}) = B(e, e_{\hat{v}}).$$

In the term $B(e, e_{\hat{v}})$ both arguments are in $W^\Gamma + H_0^{2,1}(Q_{1,2})$ and we can apply (3.3.31b). Further we make use of lemma (3.3.13) and the discretization error bound in theorem 3.3.15 to obtain

$$B(e, e_{\hat{v}}) \leq c \|e\| \|e_{\hat{v}}\|_* \leq c(h + \Delta t) \|e\| \|\hat{v}\|_{2,Q_{1,2}} \leq c(h + \Delta t) \|e\| \|\hat{v}_T\|_{1,T}$$

We thus have

$$\begin{aligned} \|(u - u_h)(\cdot, T)\|_{-1,T} &= \sup_{\hat{v}_T \in H_0^1(\Omega_{1,2}(T))} \frac{((u - u_h)(\cdot, T), \hat{v}_T)_{0,T}}{\|\hat{v}_T\|_{1,T}} \\ &\leq c(h + \Delta t) \|u - u_h\| \leq c(h^2 + \Delta t^2) \|u\|_{2,Q_{1,2}}. \end{aligned}$$

□

In [LR13, Theorem 3.15] a second order bound is given in the $\|\cdot\|_{0,T}$ -norm. The assumptions in that theorem, however, is that $\|\hat{v}\|_{2,Q_{1,2}} \leq c \|\hat{v}_T\|_{0,T}$ which is not realistic.

Remark 3.3.6. *We comment on the optimality of the error bounds presented in this section. In the main Theorems, theorem 3.3.15 and theorem 3.3.17 we used the regularity assumption that problem 3.1.1 has a solution $u \in H_0^2(Q_{1,2})$. We consider this to be a reasonable assumption but we do not know of any literature where this regularity issue is studied. Furthermore, we note that the error bound derived in theorem 3.3.15 for the Space-Time-DG Nitsche-XFEM discretization with piecewise linear functions (in space and time) is of optimal order w.r.t. the space variable but suboptimal w.r.t. time, since the norm $\|\cdot\|$ contains no derivatives w.r.t. t (cf. remark 3.3.4). In theorem 3.3.17 we derived a second order error bound. This result might be suboptimal, since for the standard Space-Time-DG FEM (not XFEM!) better bounds are known in the literature (in the stronger norm $\|\cdot\|_{L^2(\Omega)}$. In [Tho97, Theorem 12.7] for the Space-Time-DG FEM method with piecewise linear functions in space and time applied to the standard heat equation an error bound of the form*

$$\|(u - u_h)(\cdot, T)\|_{0,T} \leq c(h^2 + \Delta t^3) \quad (3.3.39)$$

is derived, i.e., an error bound with third order convergence w.r.t. Δt . For other polynomial degrees, say p_s and p_t w.r.t. space and time, respectively, based on the analysis in [Tho97] one expects that the bound in (3.3.39) can be generalized to $c(h^{p_s+1} + \Delta t^{2p_t+1})$. The error analysis in [Tho97] is, however, very different from the one presented in this section. An essential ingredient in the analysis in [Tho97] is the splitting

$$u - u_h = (u - R_h \tilde{u}) + (R_h \tilde{u} - u_h),$$

with $R_h : H_0^1(\Omega) \rightarrow V_h$ the Ritz projection corresponding to the stationary (i.e. elliptic) problem and $\tilde{u}(x, t)$ a suitable space-time interpolant of u . For this idea to work it is essential that the space V_h (and the Ritz projection R_h) is independent of t . This is in general not the case. We expect that the analysis in [Tho97] could be applied for the Space-Time-DG Nitsche-XFEM method in case of a stationary interface. In that case, however, simpler methods, for instance the method of lines, can be applied and analyzed. The numerical experiments in section 3.4 indicate that the convergence behavior of the Space-Time-DG Nitsche-XFEM method with respect to Δt is much better than the error analysis in this section suggests.

3.4 Numerical examples

In this section we present results of numerical experiments. Different from the experiments in [LR13], where only spatially one-dimensional situations have been considered, we consider spatially three-dimensional cases to assess the convergence behavior of the method. In this setting we restrict ourselves to piecewise linear (in time and space) finite element approximations. The considered examples and results are also contained in the publication [Leh15]. In all examples we consider the $L^2(\Omega)$ -error for different space and time resolutions. The time interval $(0, T]$ is divided into n_t time slabs of equal size. Accordingly the time step size is $\Delta t = \frac{T}{n_t}$. The spatial domain is always a cuboid which is either a cube or divided into a small number of cubes. The cubes are divided into $(n_s)^d$ smaller cubes which are then divided into tetrahedra. The error behaviour is investigated w.r.t. refinements, i.e. series of n_s and n_t .

For the computations we applied a strategy for the numerical integration on implicit space-time domains which is explained in detail in chapter 4. We mention some important properties of this strategy in advance as they will be part of the numerical investigations in this section. First note that the integration is carried out on individual space-time (prism) elements. The approximation of the space-time interface within each prism used in the implementation is piecewise planar. To reduce the error of this approximation further subdivisions of the prism can be applied in space and time. The parameter m_t for the subdivision in time describes that a prism element is divided into m_t elements of height $\Delta t/m_t$ before the planar approximation is done. Accordingly, for the subdivision in space the parameter m_s is used. The underlying tetrahedra T of the prism element $Q^T = T \times I_n$ is therefore divided into m_s^d tetrahedra.

We consider three examples. In the first example, discussed in section 3.4.1, the interface is a (curved) plane which moves through the domain. In section 3.4.2 the second example is presented where a moving sphere is considered. A more challenging example is the third example where the interface exhibits large deformations, this example is covered in section 3.4.3.

3.4.1 Moving plane

This case is the three-dimensional counterpart to the one-dimensional test case in [LR13]. The domain is the cube $\Omega = [0, 2]^3$. The “inner” phase is contained in the domain $\Omega_1(t) = \{\mathbf{x} \in \Omega : |x_1 - q(x_2, x_3) - r(t)| \leq D/2\}$, where $q : [0, 2]^2 \rightarrow [0, 2]$ is the graph describing the shape of the domain Ω_1 and $r : (0, T] \rightarrow \mathbb{R}$ the function describing the time-dependent shift of the interface in x_1 -direction. $D = \frac{2}{3}$ is the width of the domain in x_1 -direction. The complementary domain is $\Omega_2(t) = \Omega \setminus \Omega_1(t)$. A sketch of the domains is given in figure 3.4.1 for $q(x_2, x_3) = \text{const}$. The velocity field is given as $\mathbf{w} = (\frac{\partial r}{\partial t}(t), 0, 0)$. As boundary conditions we apply periodicity in all directions, $u(x_i=0) = u(x_i=2), i = 1, 2, 3$. This renders the problem essentially one-dimensional if

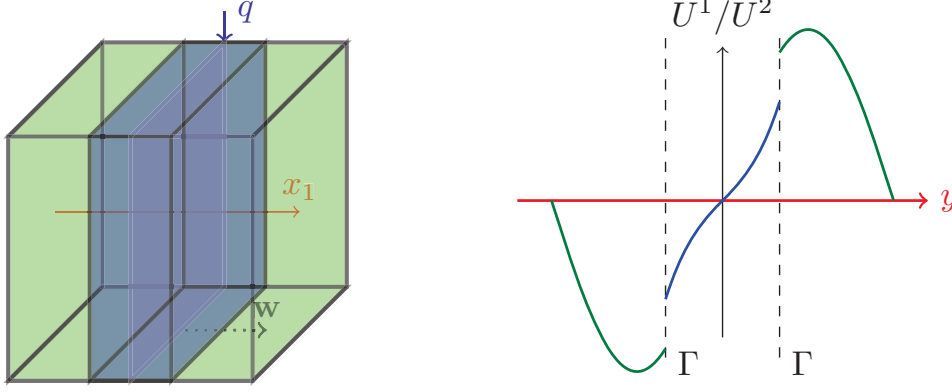


Figure 3.4.1: Sketch of the setup in section 3.4.1, Ω_1 is blue and Ω_2 is green, (left) and the time-independent part of the solution U^1/U^2 (right).

$q(x_2, x_3) = \text{const.}$

We prescribe the r.h.s. source term f , such that the solution is given by

$$u(\mathbf{x}, t) = \sin(k\pi t) \cdot U^i(x_1 - q(x_2, x_3) - r(t)), \quad \mathbf{x} \in \Omega_i(t), \quad i = 1, 2$$

with

$$U^1(y) = ay + by^3 \quad \text{and} \quad U^2(y) = \sin(\pi y) \quad (3.4.1)$$

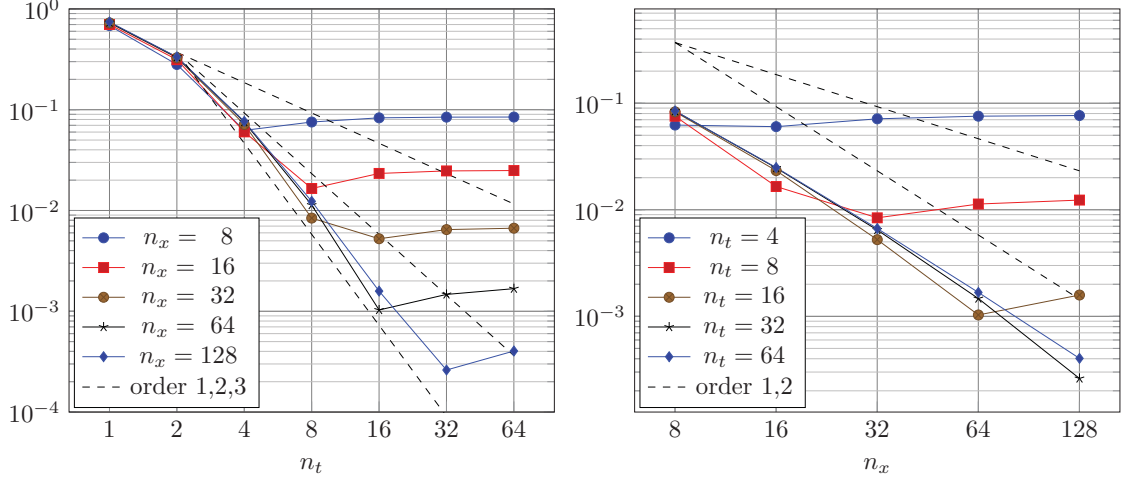
where a and b are chosen such that the interface conditions hold.

The diffusivities are $(\alpha_1, \alpha_2) = (1, 2)$ and the Henry weights $(\beta_1, \beta_2) = (1.5, 1)$, resulting in $a \approx 1.02728$ and $b \approx 6.34294$. The problem is considered in the time interval $(0, T]$ with $T = 1$.

3.4.1.1 Planar (in space and time) interface

We choose $q(x_2, x_3) = 1$ and $r(t) = 0.25t$. Hence the space-time interface is planar. Thus, the approximation of the space-time interface is exact for every $m_t, m_s \geq 1$, where m_t and m_s are the number of subdivisions in time and each space direction, respectively, cf. chapter 4. We choose $m_s = m_t = 1$. In table 3.4.1 and figure 3.4.2 we give the resulting error $\|u_h - u\|_{L^2(\Omega)}$ for different mesh and time step sizes. Similar to the results in [LR13] we observe a third order convergence w.r.t. the time step size Δt and a second order convergence w.r.t. the mesh size.

Remark 3.4.1. In [Tho97, Theorem 12.7] for the corresponding Space-Time-DG FEM method applied to the standard heat equation an error bound with third order convergence w.r.t. Δt has been derived. Note however that the analysis does not carry over for the case of the Space-Time-DG Nitsche-XFEM discretization, see also remark 3.3.6


 Figure 3.4.2: $L^2(\Omega)$ -norm for refinements in time and space for test case in section 3.4.1.1.

$n_t \setminus n_s$	8	16	32	64	128	eoc_t
1	0.6826	0.7005	0.72760	0.73301	0.742317	
2	0.2809	0.3138	0.32771	0.33260	0.334165	1.15
4	0.0624	0.0603	0.07139	0.07557	0.076649	2.12
8	0.0755	0.0165	0.00840	0.01132	0.012328	2.64
16	0.0831	0.0233	0.00524	0.00103	0.001585	2.96
32	0.0844	0.0247	0.00646	0.00147	0.000261	2.6
64	0.0846	0.0249	0.00667	0.00168	0.000403	-0.63
eoc_s		1.76	1.90	1.99	2.06	

 Table 3.4.1: $\|u_h - u\|_{L^2(\Omega(T))}$ for different refinements in time and space for the test case in section 3.4.1.1. The last column/row shows the estimated order of convergence w.r.t. time (eoc_t) / space (eoc_s) using the finest spatial/temporal resolution.

3.4.1.2 Nonlinear moving interface

We consider $q(x_2, x_3) = \frac{7}{8} + \frac{1}{4}x_2^2(2-x_2)^2$ and $r(t) = \frac{1}{4\pi} \sin(2\pi t)$ which leads to a space-time interface which is no longer planar. The geometrical approximation of the space-time interface is piecewise planar, i.e. the maximum distance between Γ_* and its approximation $\Gamma_{*,h}$ converges with second order w.r.t. increasing $n_t \cdot m_t, n_s \cdot m_s$. In table 3.4.2 the error $\|u_h - u\|_{L^2(\Omega)}$ on a fixed (fine) spatial grid with resolution $64 \times 64 \times 64$ for different numbers of time steps is listed. In order to investigate the impact of the approximation of Γ_* we performed the computation with different numbers of subdivisions m_s, m_t . The results, shown in table 3.4.2, indicate an error behaviour of the form

$$\|u_h - u\|_{L^2(\Omega)} \leq C_1 \Delta t^3 + C_2 (\Delta t / m_t)^2 + C_3(h)$$

where C_1 is independent of the approximation of Γ_* . C_2 is directly related to the interface approximation errors. If the interface approximation is exact (as in the last section) C_2 is zero. The function $C_3(h)$ describes the spatial error due to the method *and* the piecewise linear interface approximation for the numerical integration. It is thereby the part of the

n_t	1	2	4	8	16	32	64
$m_s = 1, m_t = 1$	2.50	2.89	0.547	0.137	0.0342	0.00879	0.00317
eoc _t		-0.21	2.40	2.00	2.00	1.96	1.49
$m_s = 1, m_t = 2$	2.49	0.817	0.168	0.0374	0.00878	0.00301	0.00241
eoc _t		1.61	2.28	2.17	2.09	1.54	0.32
$m_s = 1, m_t = 4$	0.520	0.481	0.0985	0.0167	0.00284	0.00219	0.00236
eoc _t		0.11	2.29	2.56	2.56	0.37	-0.11
$m_s = 1, m_t = 8$	0.491	0.412	0.0910	0.0143	0.00189	0.00212	0.00236
eoc _t		0.25	2.18	2.67	2.92	-0.17	-0.15
$m_s = 4, m_t = 8$	0.491	0.412	0.0909	0.0142	0.00179	0.00207	
eoc _t		0.25	2.18	2.68	2.99	-0.20	

Table 3.4.2: Error $\|u_h - u\|_{L^2(\Omega(T))}$ for different temporal refinements and quadrature subdivisions on a regular $64 \times 64 \times 64$ tetrahedral mesh for the test case in section 3.4.1.2.

error that can not be reduced by refinements in time. In this examples $C_3(h) \approx 0.002$. Furthermore, we observe that in this example C_3 is essentially independent of m_s .

For m_t sufficiently large, i.e. $m_t > \sqrt{C_2/(C_1 \Delta t)}$ and h sufficiently small, the first term dominates the error, such that one observes a third order in time convergence. This does not hold if m_t is too small. Especially for $m_s = m_t = 1$, the error is converging with (only) second order, due to a dominating interface approximation error.

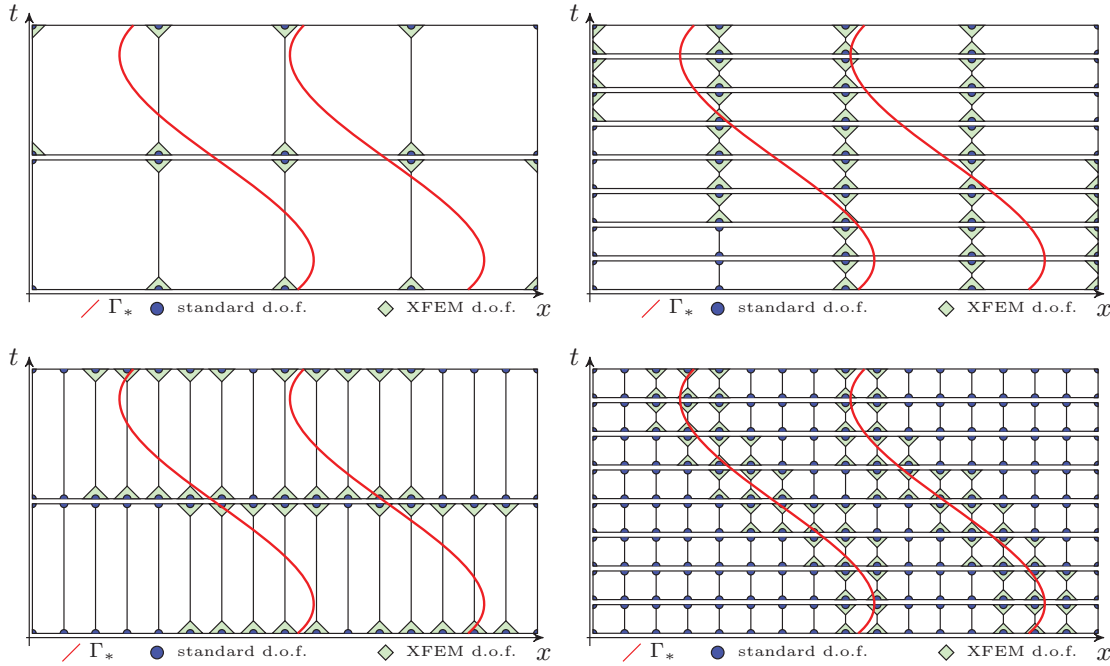


Figure 3.4.3: Sketch of different spatial and temporal resolutions for a 1D problem.

Remark 3.4.2. In order to investigate the additional effort within one time step due to additional XFEM unknowns, we consider the ratio between the maximum number of extended (XFEM) unknowns and standard (space-time) finite element unknowns within one time slab. In figure 3.4.3 a sketch of the corresponding situation is shown. If the

Std. unkn.	1024	8192	65536	524288	4194304
$n_t \setminus n_s$	8	16	32	64	128
1	736 (72%)	3648 (45%)	19328 (30%)	119808 (23%)	813056(19%)
2	656 (64%)	3008 (37%)	14336 (22%)	78848 (15%)	479232(11%)
4	656 (64%)	3008 (37%)	14336 (22%)	78848 (15%)	479232(11%)
8	656 (64%)	2880 (35%)	13056 (20%)	67072 (13%)	383488 (9%)
16	640 (63%)	2624 (32%)	11136 (17%)	52992 (10%)	275456 (7%)
32	640 (63%)	2496 (30%)	10368 (16%)	45824 (9%)	198656 (5%)
64	640 (60%)	2496 (30%)	9856 (15%)	41472 (8%)	182784 (4%)

Table 3.4.3: Number of standard (space-time) unknowns (first row) and maximal number of XFEM unknowns for one time slab In brackets the ratio between XFEM and standard unknowns is added.

interface is well resolved, the number of unknowns close to the interface increases by a factor of 2^{d-1} for one uniform (spatial) refinement whereas the overall number of unknowns increases with 2^d . Thus the ratio decreases linearly with the spatial resolution. In table 3.4.3 the corresponding numbers for this test case are given which are in agreement with the expected behaviour.

Remark 3.4.3. To decrease the (space-time) interface approximation error one can either choose smaller time steps or a larger refinement factor m_t for the construction of $\Gamma_{*,h}$. The computation with a fixed $\Delta t = \Delta \tilde{t}$ and $m_t = \tilde{m}_t > 1$ is cheaper than a computation with $\Delta t = \Delta \tilde{t}/\tilde{m}_t$ and $m_t = 1$. In figure 3.4.4 a sketch of both strategies is shown. Note that the approximation quality of the piecewise planar interface for both cases is the same. For $m_t > 1$ additional effort due to the decomposition strategy and quadrature within one time step is required. However if the interface is resolved, this is only required for a small number of elements (cf. remark 3.4.2). The number of time steps and thereby the number of linear systems that have to be solved however is reduced by a factor of $\tilde{m}_t$. Note that the solution of linear systems is typically the most time consuming part.

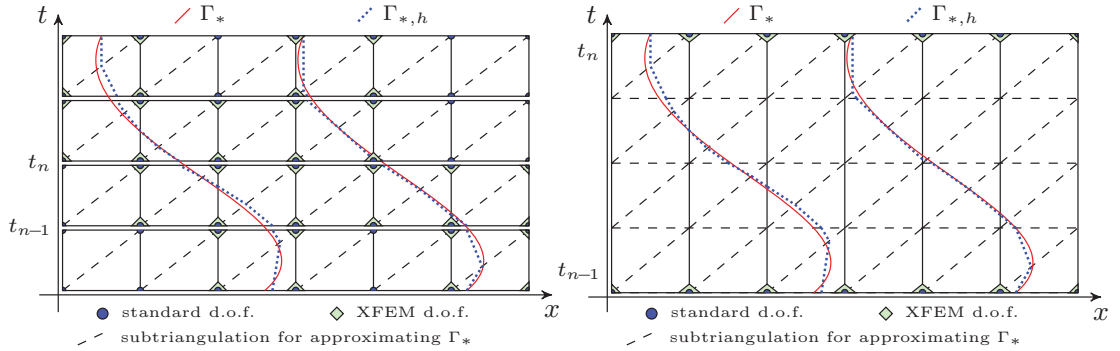


Figure 3.4.4: Sketch of geometry approximation and degrees of freedom for the case $m_t = 1$ and $\Delta t = \Delta \tilde{t}/4$ (left) and for the case $m_t = 4$ and $\Delta t = \Delta \tilde{t}$ (right).

3.4.2 Moving sphere

In this example we consider a rotational symmetric solution for a stationary sphere. The sphere is then translated with a time-dependent velocity. The time interval is $(0, T]$ with $T = 0.5$ and the domain is the cube $\Omega = [0, 2]^3$. One phase is contained in the domain

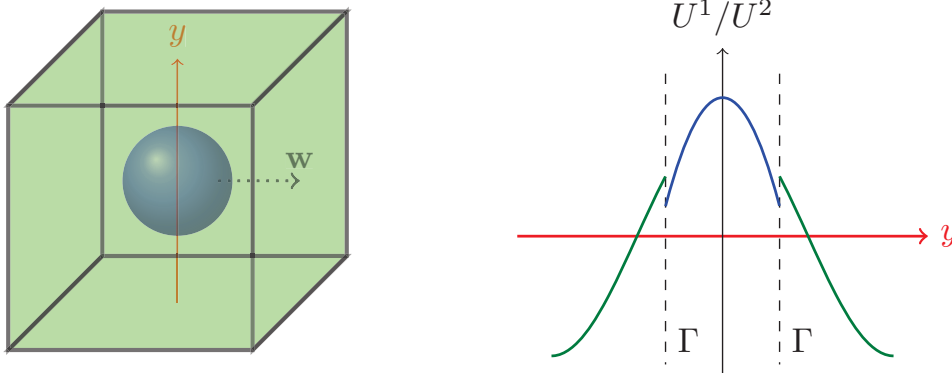


Figure 3.4.5: Sketch of geometrical setup for test case in section 3.4.2, Ω_1 is blue and Ω_2 is green, (left) and sketch of the time-independent part of the solution U^1/U^2 (right).

$\Omega_1(t) = \{\mathbf{x} \in \Omega : \|x - (\mathbf{p}_0 + r(t) \cdot \mathbf{e}_1)\| \leq R\}$, where $\mathbf{p}_0$ is the center of the initial sphere and $r(t)$ the motion of the interface in x_1 -direction, $\mathbf{e}_1$ is the corresponding unit vector. $R = \frac{1}{3}$ is the radius of the sphere. The complementary domain is $\Omega_2(t) = \Omega \setminus \Omega_1(t)$.

The velocity field $\mathbf{w}$ is given as $\mathbf{w} = (\frac{\partial r}{\partial t}(t), 0, 0)$. As boundary conditions we apply suitable Dirichlet boundary conditions everywhere.

We prescribe these boundary conditions and the r.h.s. source term f , such that the solution is given by

$$u(\mathbf{x}, t) = \sin(k\pi t) \cdot U^i(\|\mathbf{x} - (\mathbf{p}_0 + r(t) \cdot \mathbf{e}_1)\|), \quad \mathbf{x} \in \Omega_i(t), \quad i = 1, 2$$

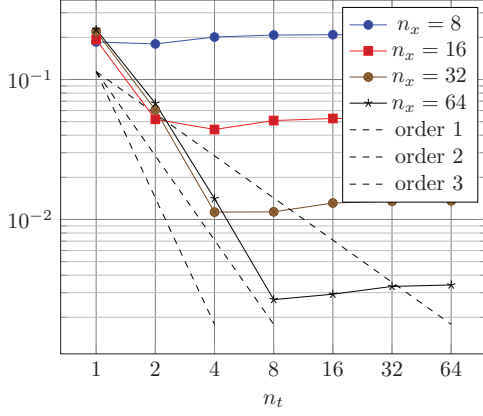
with

$$U^1(y) = a + by^2 \quad \text{and} \quad U^2(y) = \cos(\pi y), \quad (3.4.2)$$

where a and b are chosen s.t. the interface conditions hold. The diffusivities are $(\alpha_1, \alpha_2) = (10, 20)$ and the Henry weights $(\beta_1, \beta_2) = (2, 1)$ resulting in $a \approx 1.1569$ and $b \approx -8.1621$. The problem is considered in the time interval $(0, T]$ with $T = 0.5$. We choose $\mathbf{p}_0 = (0.5, 1, 1)^T$ and $r(t) = \frac{1}{4\pi} \sin(2\pi t)$. For the approximation of the space-time interface we consider $m_s = m_t = 1$. We observe an error behaviour which is of (at least) second order in time and space ($\mathcal{O}(h^2 + \Delta t^2)$) (see figure 3.4.6). For the finest spatial resolution ($n_s = 64$) we observe an order around 2.5 for the convergence in time. In contrast to the previous test cases the spatial error dominates the overall error already for coarse temporal resolutions. We expect that for finer spatial resolutions and better geometry approximations ($m_t > 1, m_s \geq 1$) one could retain the third order convergence in time.

It is also relevant to study the accuracy of the method w.r.t. the interface condition.

3 Mass transport through a moving interface



$n_t \backslash n_s$	8	16	32	64
1	0.185	0.1929	0.2202	0.22794
2	0.180	0.0520	0.0607	0.06760
4	0.201	0.0440	0.0113	0.01409
8	0.208	0.0509	0.0113	0.00268
16	0.209	0.0527	0.0131	0.00293
32	0.209	0.0530	0.0135	0.00332
64	0.209	0.0530	0.0136	0.00340

Figure 3.4.6: Convergence in $L^2(\Omega)$ -norm w.r.t. refinements in time and space as a plot and as tabulated values for test case in section 3.4.2.

Therefore, in table 3.4.4 we consider the error

$$\|\nu^{\frac{1}{2}} \llbracket \beta u_h \rrbracket\|_{L^2(\Gamma_{*,h})} = \left(\int_0^T \int_{\Gamma_h(t)} \llbracket \beta u_h \rrbracket^2 ds dt \right)^{\frac{1}{2}}$$

under space and time refinement and also observe an $\mathcal{O}(h^2 + \Delta t^2)$ behaviour.

$n_t \backslash n_s$	8	16	32	64	eoc _t
2	0.0495	0.00700	0.0198	0.0587	
4	0.0430	0.00384	0.00567	0.0227	1.37
8	0.0417	0.00253	0.00164	0.00517	2.13
16	0.0414	0.00205	0.000716	0.00117	2.14
32	0.0413	0.00190	0.000523	0.000275	2.10
64	0.0413	0.00186	0.000477	0.000131	1.07
eoc _s		4.48	1.96	1.86	

Table 3.4.4: Interface error $\|\nu^{\frac{1}{2}} \llbracket \beta u_h \rrbracket\|_{L^2(\Gamma_{*,h})}$ for different refinements in time and space for the test case in section 3.4.2. The last column/row shows the estimated order of convergence w.r.t. time (eoc_t) / space (eoc_s) using the finest spatial/temporal resolution.

3.4.3 Deforming bubble in a vortex

As a last example we consider a more complex configuration. We consider an ellipsoidal bubble which is deforming under a vortex velocity field. The domain is $\Omega = [0, 2] \times [0, 2] \times [0, 1]$ and the velocity field is given as:

$$\mathbf{w}(\mathbf{x}, t) = \mathbf{w}(\mathbf{x}) = q(r(\mathbf{x})) \cdot (x_2 - c_2, c_1 - x_1, 0)^T \quad (3.4.3)$$

where (c_1, c_2, x_3) with $c_1 = c_2 = 1$ describes the rotation axes of the vortex and

$$r(\mathbf{x}) := \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2}.$$

The angular velocity varies with a changing distance to the rotation axes. The dependency is described by $q(r)$ which is chosen as follows:

$$q(r) = \frac{\pi}{10} \cdot \begin{cases} 1 + 2^{\frac{2 \cdot 5^{10}}{3^{13}}} r^2 (0.9 - r)^3 & r \leq 0.9 \\ 10^4 (x - 0.8)^2 (x - 1)^2 & 0.9 < r < 1 \\ 0 & 1 \leq r \end{cases} \quad (3.4.4)$$

The time interval is $(0, T]$ with $T = 20$ which corresponds to one full rotation of the bubble. Due to $q(r) \neq \text{const}$ the bubble deforms during the process. In figure 3.4.7 the interface at different time levels is shown.

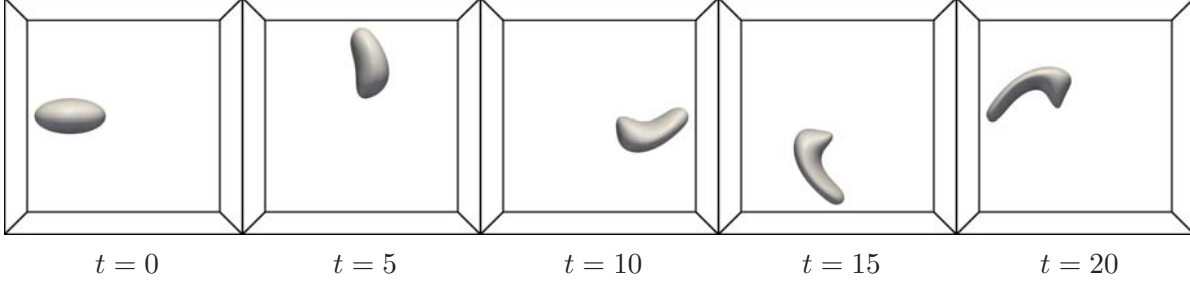
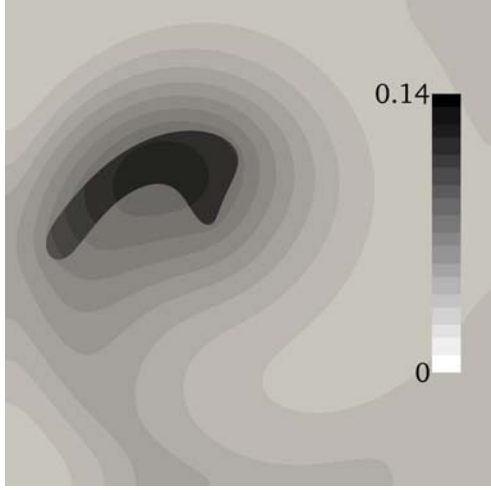


Figure 3.4.7: Interface position at several times for the test case in section 3.4.3, view from the top.

Initially we set

$$\Omega_1(0) = \{3^2(x_1 - b_1^0)^2 + 6^2(x_2 - b_2^0)^2 + 3^2(x_3 - b_3^0)^2 \leq 1\}$$

with $(b_1^0, b_2^0, b_3^0) = (0.5, 1, 0.5)$, the center of the initial bubble. The complementary initial domain is $\Omega_2(0) = \Omega \setminus \Omega_1(0)$. The interface evolves along the characteristics given by $\dot{\mathbf{x}}(t) = \mathbf{w}(\mathbf{x}, t)$ which can be calculated explicitly. The diffusivities are $(\alpha_1, \alpha_2) = (8 \cdot 10^{-3}, 4 \cdot 10^{-3})$ and the Henry weights are $(\beta_1, \beta_2) = (1, 1.75)$. As initial condition we choose $u_1(\mathbf{x}, 0) = 1$ and $u_2(\mathbf{x}, 0) = 0$. Note that $u(\cdot, 0)$ does not fulfill the interface condition (3.1.1c) which leads to a parabolic boundary layer of size $\mathcal{O}(\sqrt{\alpha t})$. For this setup we do not know the exact solution. We therefore computed a reference solution u_{ref} on a spatial fine mesh $(96 \times 96 \times 48)$ with a characteristic mesh size of $h = 1/96$ and a temporal resolution of 1024 timesteps for the whole time interval, i.e. $\Delta t_{\text{ref}} \approx 2 \cdot 10^{-2}$. This solution is denoted by u_{ref} . We only investigate the temporal convergence by varying the number of time steps n_t . For the approximation of the space-time interface we consider $m_s = 2$, $m_t = 1$. A contour plot of the solution and the convergence table is given in figure 3.4.8. We observe a convergence order in time of nearly two. The range in which we observe this order of convergence is limited by $n_t < 256$. This is due to several effects. At first the reference solution is different from the exact solution. Secondly due to the dependence of the finite element space and the approximated space-time interface on the time step size the impact of the spatial discretization errors can be different for different time step sizes. Hence $u - u_{\text{ref}}$ does contain spatial errors which do not vanish for $\Delta t \rightarrow 0$.



n_t	$\ u - u_{\text{ref}}(\cdot, T)\ _{L^2(\Omega)}$	eoc_t
32	$1.22569 \cdot 10^{-3}$	
64	$3.32611 \cdot 10^{-4}$	1.88
128	$9.65349 \cdot 10^{-5}$	1.78
256	$4.08119 \cdot 10^{-5}$	1.24

Figure 3.4.8: Numerical solution and convergence table of the test case in section 3.4.3. The contour plot (left) shows the numerical solution ($n_s = 96$, $n_t = 1024$) of the test case in section 3.4.3 at $T = 20$ in the cutting plane $z = 0.5$. The table shows the convergence in time w.r.t. a reference solution.

3.5 Preconditioning of the Space-Time-DG Nitsche-XFEM method

In this section we discuss the preconditioning of linear systems arising from Space-Time-DG Nitsche-XFEM discretizations. Analysis of the linear systems and suitable preconditioners in this case is much harder than for the elliptic case discussed in section 2.4. This is due to the non-symmetric terms stemming from the convection as well as the time derivative. We will thus only motivate a new preconditioner which we apply and the performance of which we evaluate for one test case and variations of discretization and material parameters.

Before we turn over to the discussion of more sophisticated preconditioners we investigate the performance of simple diagonal preconditioners in section 3.5.2. Then we introduce and discuss a new preconditioner in three steps. The preconditioner is based on two

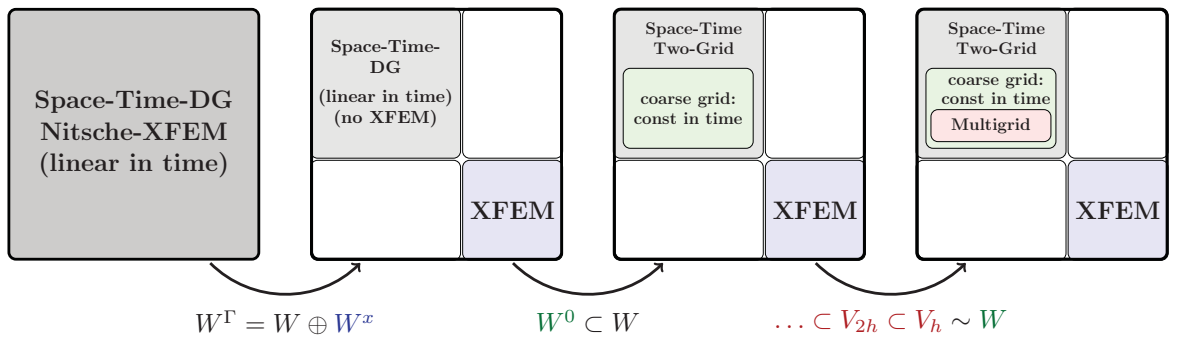


Figure 3.5.1: Sketch of precondition strategy for Space-Time-DG Nitsche-XFEM .

components. The first component is a block decomposition as in section 2.4 which divides the problem into the preconditioning of the blocks corresponding to the XFEM unknowns and the standard space-time finite element unknowns, this topic is covered in section 3.5.3. For the XFEM block we apply a simple diagonal preconditioning, see section 3.5.4. For the preconditioning of the larger block corresponding to the standard degrees of freedoms we propose a two-grid method based on a coarse grid space which consists of the functions which are constant in time. This is discussed in section 3.5.5. The efficient solution of the coarse grid problem is crucial for the performance of the two-grid method. Here, a geometrical multigrid method in space seems to be applicable. We comment on this in remark 3.5.1. In section 3.5.6 we put the components together and investigate the performance of the resulting preconditioner for the Space-Time-DG Nitsche-XFEM method. In figure 3.5.1 the structure of the preconditioning concept is illustrated. We comment on a similar strategy in remark 3.5.2.

3.5.1 Preliminaries

We recall the discrete problem and the resulting linear systems for the formulation of one time slab problem. For ease of presentation we will skip the super- and subscripts n indicating the time step in the following. We make use of the fact that there holds $u_h \in W^\Gamma \Leftrightarrow \beta^{-1}u_h \in W^\Gamma$ such that the problem in (3.2.20) can be reformulated in terms of the unknown $\tilde{u}_h = \beta u_h$. This reformulation has been essential for the elliptic problems in section 2.4 for the theoretical analysis as well as for the performance of the iterative solvers. In numerical experiments we observed that the reformulation of the Space-Time-DG Nitsche-XFEM discretization is also beneficial for the solution of linear systems, especially if the ratios of the Henry weights increases. Let $\tilde{\alpha} = \alpha/\beta$. Then, the reformulated problem reads as follows.

Find $\tilde{u}_h \in W^\Gamma$ such that

$$\begin{aligned} B(\tilde{u}_h, v_h) &= a(\tilde{u}_h, v_h) + b(\tilde{u}_h, v_h) + d(\tilde{u}_h, v_h) + N(\tilde{u}_h, v_h) \\ &= f(v_h) + c(\tilde{u}^{n-1}, v_h) \quad \forall v_h \in W^\Gamma \end{aligned} \quad (3.5.1)$$

with the following bilinear forms ($u, v \in W^\Gamma$)

$$\begin{aligned} a(u, v) &= (\tilde{\alpha} \nabla u, \nabla v)_{L^2(Q_{1,2}^n)} \\ b(u, v) &= (\beta^{-1}(\partial_t u + \mathbf{w} \cdot \nabla u), v)_{L^2(Q_{1,2}^n)} \\ d(u, v) &= (\beta^{-1}u_+^{n-1}, v_+^{n-1})_{L^2(\Omega_{1,2}^n)} \\ N(u, v) &= -(\{\tilde{\alpha} \nabla u \cdot \mathbf{n}\}, \llbracket v \rrbracket)_{\Gamma_*} - (\llbracket u \rrbracket, \{\tilde{\alpha} \nabla v \cdot \mathbf{n}\})_{\Gamma_*} + \bar{\alpha} \lambda h_T^{-1}(\llbracket u \rrbracket, \llbracket v \rrbracket)_{\Gamma_*} \end{aligned}$$

The linear forms are defined accordingly. Note that instead of the scalar product $(\cdot, \cdot)_{Q_{1,2}^n}$ the standard L^2 scalar product (without weighting) has been used.

Matrix/vector representation and block decomposition

The matrix/vector representation to the linear and bilinear forms are $\mathbf{A}_{i,j} = B(\varphi_i, \varphi_j)$ and $\mathbf{b}_j = f(\varphi_j) + c(\tilde{u}^{n-1}, \varphi_j)$ with basis functions $\varphi_i, \varphi_j \in W^\Gamma, i, j = 1, \dots, n$ and n the dimension of W^Γ . The discrete solution $\tilde{u}$ is represented by $\mathbf{u} \in \mathbb{R}^n$, such that there holds $\tilde{u} = \sum_{i=1}^n \mathbf{u}_i \varphi_i$. By construction the space W^Γ has a decomposition into standard finite element functions in W and enrichment functions in W^x . Let n_1 be the number of degrees of freedom in the W and n_x be the number of degrees of freedom in W^x , such that $n = n_1 + n_x$. We use the following notation for the splitting of a vector $\mathbf{v} \in \mathbb{R}^n$ and a matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ into blocks corresponding to a decomposition of the basis functions into standard and enrichment functions:

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_s \\ \mathbf{v}_x \end{pmatrix}, \quad \mathbf{v}_s \in \mathbb{R}^s, \mathbf{v}_x \in \mathbb{R}^x,$$

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_s & \mathbf{M}_{sx} \\ \mathbf{M}_{xs} & \mathbf{M}_x \end{pmatrix}, \quad \mathbf{M}_s \in \mathbb{R}^{n_1 \times n_1}, \mathbf{M}_{sx} \in \mathbb{R}^{n_x \times n_1}, \mathbf{M}_{xs} \in \mathbb{R}^{n_1 \times n_x}, \mathbf{M}_{xx} \in \mathbb{R}^{n_x \times n_x}.$$

Further we define the diagonal matrices to $\mathbf{A}$ and its diagonal blocks, $\mathbf{D} = \text{diag}(\mathbf{A}) \in \mathbb{R}^{n \times n}$, $\mathbf{D}_x = \text{diag}(\mathbf{A}_x) \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{D}_s = \text{diag}(\mathbf{A}_s) \in \mathbb{R}^{n_1 \times n_1}$.

Test problem

To evaluate the quality of the following preconditioning concepts we consider the problem in section 3.4.3. This problem is demanding due to the complex evolution of the interface. By varying the problem parameters $h, \Delta t, \alpha, \beta$ and λ , we can further check the sensitivity of the preconditioning concepts. We set $\Delta t = \frac{20}{n_t}$, $h = \frac{2}{n_s}$ and the default discretization parameters to $n_t = 20$, $n_s = 16$. The default parameters for β and α are $\beta = (1, 7/4)$ and $\alpha = (8, 4) \cdot 10^{-3}$ which gives the (transformed) problem parameters $\tilde{\alpha} = (56/7000, 16/7000)$. The default parameter for λ is set to $\lambda = 100$. For the spatially three-dimensional test problem we consider different resolutions in space and time and vary $n_s \in \{16, 32, 64\}$ and $n_t \in \{20, 40, 80, 160\}$. In the following investigations we consider the solution of the linear systems $\mathbf{A}\mathbf{u} = \mathbf{b}$ for all time steps within the time interval $(0, 20]$.

3.5.2 Diagonal preconditioning

The simplest choice for a preconditioner - except for not preconditioning at all - is the diagonal preconditioner. In connection with XFEM one important question is if the conditioning of the preconditioned matrix depends on the cut position. Especially if the support of XFEM functions gets very small this is of particular interest. For stationary elliptic problems and the Nitsche-XFEM method we observed and proved that a diagonal preconditioning is already sufficient to get condition number bounds independent of the cut position. We investigate the same for the non-stationary case and the Space-Time-DG

Nitsche-XFEM method. Note that within the evolution of the interface through the mesh in the test problem in section 3.4.3 the cut configurations are essentially arbitrary that means that XFEM functions with only very small support can appear at several time steps. Note further, that the system matrix in our problem is not symmetric due to the advection term *and* the time derivative which also acts as an advection term in space-time.

We consider a GMRES method which is preconditioned with a simple Gauss-Seidel method and investigate how many iterations are necessary to reduce the initial residual by a factor of 10^{-6} .

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 100$							$\alpha = (8, 4) \cdot 10^{-2}, \beta = (1, 7/4), \lambda = 100$						
$n_s \backslash n_t$	20	40	80	160			$n_s \backslash n_t$	20	40	80	160		
16	41/ 52	37/ 47	37/ 49	35/49			16	52/ 61	42/ 52	43/ 57	39/ 55		
32	108/132	61/ 82	49/ 61	43/53			32	116/143	86/101	63/ 71	58/ 71		
64	318/378	163/216	84/108	59/74			64	301/329	194/215	139/158	99/112		
$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 20$							$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 1000$						
$n_s \backslash n_t$	20	40	80	160			$n_s \backslash n_t$	20	40	80	160		
16	33/ 38	27/ 31	24/28	23/27			16	48/ 71	48/ 68	52/ 78	52/ 92		
32	75/ 82	43/ 49	31/36	27/31			32	174/ 256	114/177	84/132	77/103		
64	185/212	109/127	57/67	38/44			64	763/1433	400/762	209/367	136/218		
$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 3.5), \lambda = 100$							$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7), \lambda = 100$						
$n_s \backslash n_t$	20	40	80	160			$n_s \backslash n_t$	20	40	80	160		
16	44/ 55	38/ 50	38/ 51	36/51			16	46/ 58	41/ 53	40/ 55	37/53		
32	114/150	68/ 92	54/ 67	48/58			32	132/179	79/103	59/ 74	53/66		
64	366/435	204/257	96/125	66/82			64	444/521	252/322	112/148	73/93		

Table 3.5.1: Iteration numbers (min/max) for Gauss-Seidel preconditioned GMRES of the Space-Time-DG Nitsche-XFEM method using different discretization and material parameters

In table 3.5.1 the iteration counts for different discretization parameters and material parameters have been collected. The minimal and maximal number of iterations within all time steps are shown. We discuss the results. We observe that the ratio of minimal and maximal iteration numbers stays within reasonable bounds such that we conclude that we do not observe a severe dependency of the performance of iterative solvers on the cut position. We, however, observe a dependency on discretization and material parameters.

While for decreasing h the iteration numbers increase, a decreasing Δt leads to a decrease in the iteration numbers. Discretizations with the same quotient n_s^2/n_t have a similar number of iterations. This behavior is similar to that of linear systems arising from method

of lines discretizations for simpler parabolic problems, cf. section 2.4.6.3. The amount of Nitsche stabilization in terms of the stabilization parameter λ plays an important role for the conditioning, too. A stabilization parameter which is chosen too large, increases the number of iterations and thereby the computational costs significantly.

An increase in the diffusion parameter also leads to a slight increase in iterations. The same holds for increasing ratios of β .

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4)$						$\alpha = (8, 4) \cdot 10^{-2}, \beta = (1, 7/4)$					
$n_s \backslash n_t$	20	40	80	160		$n_s \backslash n_t$	20	40	80	160	
16	24/ 26	18/19	15/16	14/15		16	47/ 51	32/ 35	23/ 25	18/ 19	
32	56/ 60	33/37	22/24	17/18		32	105/115	80/ 86	52/ 57	38/ 41	
64	155/161	85/94	51/57	31/34		64	286/301	189/206	129/155	93/104	

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 3.5)$						$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7)$					
$n_s \backslash n_t$	20	40	80	160		$n_s \backslash n_t$	20	40	80	160	
16	24/ 26	18/20	15/16	14/15		16	24/ 27	18/ 20	15/17	14/15	
32	58/ 61	34/39	22/25	17/18		32	61/ 64	35/ 40	22/25	17/18	
64	166/174	88/99	51/59	31/35		64	171/181	91/102	52/59	31/35	

Table 3.5.2: Iteration numbers (min/max) for Gauss-Seidel preconditioned GMRES method of the standard FEM block.

In table 3.5.2 we consider the discretization of the same problems as in table 3.5.1 but without the XFEM enrichment. Note that the matrix corresponding to this discretization also coincides with the block matrix $\mathbf{A}_s$ corresponding to the standard degrees of freedom in the matrix of the Space-Time-DG Nitsche-XFEM discretization. The purpose of this comparison is that we can observe which effects are stemming from the XFEM functions and which not. We observe that the behavior of the iteration numbers is similar, especially for small λ . However, we observe that the iteration numbers for the matrix with XFEM are roughly two times higher. We also mention that the fact that the cut positions can be almost arbitrary has no significant impact on the conditioning as long as (at least) a diagonal preconditioner is used.

3.5.3 Block preconditioning

In section 2.4 we showed that the decomposition of the function in an XFEM-enriched finite element space into XFEM functions and standard functions is stable in a norm suitable for scalar elliptic problems and a Nitsche-XFEM discretization. This result implies that for the Nitsche-XFEM discretization a block preconditioner which uses exact inverses for the standard functions and the XFEM functions leads to a condition number which is bounded independent of the position of Γ and the mesh size h . We want to investigate if this is also true for the Space-Time-DG Nitsche-XFEM method.

For this purpose we use a block preconditioner with exact inverses and consider the same parameter variations as in the previous section. The preconditioner is

$$\mathbf{C}_B = \begin{pmatrix} \mathbf{A}_s & 0 \\ 0 & \mathbf{A}_x \end{pmatrix}.$$

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 100$					$\alpha = (8, 4) \cdot 10^{-2}, \beta = (1, 7/4), \lambda = 100$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	11/15	10/12	11/13	13/16	16	10/13	9/10	8/10	9/11
32	15/18	11/13	10/12	11/12	32	13/17	11/14	10/12	9/11
64	22/27	15/19	11/14	10/11	64	20/27	14/21	11/16	10/12

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = \mathbf{20}$					$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = \mathbf{1000}$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	17/22	14/17	14/17	15/19	16	7/ 8	6/ 7	8/ 9	9/11
32	22/27	16/20	14/17	14/16	32	8/10	7/ 8	7/ 8	8/ 8
64	31/40	22/30	16/22	14/17	64	14/17	10/11	8/10	8/ 8

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (\mathbf{1}, \mathbf{3.5}), \lambda = 100$					$\alpha = (8, 4) \cdot 10^{-3}, \beta = (\mathbf{1}, \mathbf{7}), \lambda = 100$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	11/13	10/12	12/13	13/17	16	11/13	11/13	12/15	15/18
32	15/19	12/14	11/12	12/13	32	16/21	13/15	12/14	13/15
64	24/30	16/20	13/15	12/13	64	28/35	18/22	15/17	13/15

Table 3.5.3: Iteration numbers (min/max) for block-preconditioned GMRES method.

In table 3.5.3 we observe that the number of iterations needed for the block preconditioned matrix is small, but not independent of the discretization parameters h and Δt . In contrast to the simpler case of an elliptic interface problem the iteration counts of the block preconditioned solver are no longer independent of h but increase with decreasing $h/\sqrt{\Delta t}$. The increase, however, is only mild. The dependency on the diffusion parameter seems to be very mild while the results for varying λ show a behavior which is similar to what has been observed in section 2.4. For a larger stabilization parameter λ the angle between the space of enrichment functions and the standard space increases which is in favor of the block preconditioning and lowers the iteration counts.

As long as the ratios of the diffusion and Henry parameters are moderate the block preconditioning seems to be effective. The efficiency of this decomposition depends on the efficiency of the solvers for the individual blocks. In the following we discuss the preconditioning of the blocks corresponding to the XFEM unknowns and the standard unknowns.

3.5.4 Diagonal preconditioning for XFEM block

Similar to what we did for the stationary case we try to manage the solution of the XFEM block with a Jacobi-preconditioner. To characterize the quality of the diagonal preconditioning we consider the solution of the XFEM block up to a relative accuracy of 10^{-6} .

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 100$					$\alpha = (8, 4) \cdot 10^{-2}, \beta = (1, 7/4), \lambda = 100$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	42/ 54	40/ 51	43/54	41/52	16	41/ 54	42/ 54	49/62	53/66
32	76/ 95	60/ 75	53/66	56/67	32	74/ 94	60/ 78	57/71	63/76
64	116/143	89/112	70/90	62/77	64	111/144	88/115	70/91	64/82

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 20$					$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 1000$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	28/33	26/32	25/30	23/27	16	50/ 70	52/ 72	61/ 79	64/ 91
32	42/52	33/39	31/37	29/34	32	116/ 174	100/ 138	98/124	99/125
64	61/75	44/55	35/43	35/42	64	>1K/>1K	215/>1K	168/221	149/179

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 3.5), \lambda = 100$					$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7), \lambda = 100$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	42/ 54	42/ 53	44/55	42/53	16	45/ 55	42/ 54	45/56	43/54
32	78/ 96	62/ 77	55/67	56/68	32	80/ 99	61/ 79	55/68	56/70
64	122/152	91/116	72/91	63/78	64	126/171	95/118	73/92	65/80

Table 3.5.4: Iteration numbers (min/max) for Gauss-Seidel preconditioned GMRES of the XFEM block of the Space-Time-DG Nitsche-XFEM method using different discretization and material parameters. Entries with “>1K” indicate that the solver did not converge within 1000 steps.

In table 3.5.4 the results are shown. We observe that the iteration numbers behave similar for the XFEM block as the diagonal preconditioning for the whole system with respect to the material parameters and h and Δt . With respect to λ we observe a similar behavior as for the stationary case: for a small λ the diagonal preconditioning performs reasonably, but for a large λ iteration numbers increase dramatically. Note however that the number of XFEM degrees of freedom are always the smaller portion of unknowns such that a non-optimal behavior of the XFEM block preconditioning is not as important as a good preconditioning for the block of standard (space-time) FEM unknowns. This is what we consider next.

3.5.5 Preconditioning for space-time finite element block

The preconditioning of the space-time finite element block with extended finite elements is a challenging task. This topic is only rarely discussed in the literature and if, it is

discussed for the case of constant coefficients, i.e. for equal diffusion parameters α and equal Henry coefficients β . For instance, interesting preconditioning ideas for parabolic problems with time-independent coefficients are discussed in [WB14]. Note however that in our case the (transformed) diffusion coefficients are not equal and vary in time due to the interface motion.

A very simple preconditioner is the diagonal preconditioner. Together with a diagonal preconditioning of the XFEM block this results in a simple diagonal preconditioner of the whole system which we discussed before. Instead we consider a space-time version of a two-grid approach which we present and discuss next.

Space-Time Two-Grid strategy

The block $\mathbf{A}_s$ corresponding to the standard space-time finite element space W represents a (discontinuous) space-time discretization of the problem with a standard (space-time) finite element space. Note that due to the reformulation ($u \leftrightarrow \beta u$) the Nitsche bilinear form is zero for every standard basis function, $N(v, w) = 0$ if $v \in W$ and $w \in W$.

Instead of the preconditioning technique presented in [WB14] we consider a somewhat simpler approach. The basis for the space-time finite element space W is constructed by taking one basis element $\phi_j(\mathbf{x}) \in V_h$ multiplied with a basis element $\psi(t)$ of $\mathcal{P}^1(I_n)$. In our construction of the finite element space we used a standard nodal basis with the basis functions $\psi_0(t) = \frac{t_n - t}{\Delta t}$, $\psi_1(t) = \frac{t - t_{n-1}}{\Delta t}$.

A natural coarse grid space in this setting is the subspace of functions which are constant in time at the current time slab $W^0 = \{u, u(\mathbf{x}, t) = v(\mathbf{x}) \text{ for a } v \in V_h\} \subset W$. Let n_0 be the dimension of W^0 such that $n_1 = 2n_0$. We define the restriction and prolongation operations $R : W \rightarrow W^0$ and $P : W^0 \rightarrow W$ with matrix representations $\mathbf{R} \in \mathbb{R}^{n_0 \times n_1}$ and $\mathbf{P}^{n_1 \times n_0}$ where we have $\mathbf{R} = \mathbf{P}^T$. The coarse grid matrix is defined as $\mathbf{A}_c = \mathbf{R}\mathbf{A}_s\mathbf{P} \in \mathbb{R}^{n_0 \times n_0}$. This matrix corresponds to a discretization with constant-in-time functions as in section 3.2.4.2 (without quadrature in time). Note that the terms including the time derivative $\partial_t u$ vanish for $u \in W^0$ and the structure of the corresponding bilinear form is very similar to that of a backward Euler discretization.

The two-grid space-time preconditioner consists of the following steps:

1. Pre-smoothing: $\mathbf{u} \leftarrow \mathbf{u} + \mathbf{M}^{\text{sm}}(\mathbf{b} - \mathbf{A}_s\mathbf{u})$
2. Restriction of defect: $\mathbf{d}_c \leftarrow \mathbf{R}(\mathbf{b} - \mathbf{A}_s\mathbf{u})$
3. Solution of “coarse grid” problem: $\mathbf{A}_c\mathbf{c}_c = \mathbf{d}_c$
4. Prolongation and application of correction: $\mathbf{u} \leftarrow \mathbf{u} + \mathbf{P}\mathbf{c}_c$
5. Post-smoothing: $\mathbf{u} \leftarrow \mathbf{u} + \mathbf{M}^{\text{sm}}(\mathbf{b} - \mathbf{A}_s\mathbf{u})$

Note that $\mathbf{u}, \mathbf{b} \in \mathbb{R}^{n_1}$ and $\mathbf{d}_c, \mathbf{c}_c \in \mathbb{R}^{n_0}$. For the smoother $\mathbf{M}^{\text{sm}}$ we use one step of a Jacobi-smoother.

We apply this preconditioner to the space-time finite element discretization (without XFEM and without Nitsche) and consider the same test case and parameter variations as before.

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 100$					$\alpha = (8, 4) \cdot 10^{-2}, \beta = (1, 7/4), \lambda = 100$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	10/11	10/12	11/12	9/11	16	11/14	9/10	8/9	8/9
32	13/15	10/11	10/11	9/11	32	21/27	15/20	12/15	9/11
64	27/31	15/18	10/11	9/10	64	45/55	29/40	21/29	16/21

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 3.5), \lambda = 100$					$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7), \lambda = 100$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	11/12	11/12	11/11	9/11	16	11/12	11/12	11/11	9/11
32	14/15	11/11	10/11	9/10	32	14/16	11/11	10/11	9/10
64	28/31	15/18	11/11	10/10	64	27/31	16/17	11/11	10/10

Table 3.5.5: Iteration numbers of GMRES preconditioned with two-grid preconditioner for the standard finite element block.

In table 3.5.5 one can observe the in general good performance of this preconditioner for the block of standard space-time unknowns. Similar to the results of the block preconditioner before, the results are not independent of the mesh size h and the diffusion parameter α , but the dependencies seem to be less severe than for the diagonal preconditioner in table 3.5.2. One would expect that the number of maximal iterations of the two-grid preconditioner is essentially independent of the mesh size h . This however does not seem to be true. At this point, we do not have a good explanation for this.

3.5.6 A new preconditioner for the Space-Time-DG Nitsche-XFEM method

At last, we combine the block preconditioning of standard and XFEM functions and the two-grid strategy for the standard FEM block. We use a GMRES method with a block preconditioning corresponding to the standard and the XFEM blocks. The exact solution of the diagonal blocks is replaced by approximate solutions obtained with preconditioned iterative solvers. The solution of the XFEM block is obtained using a diagonally preconditioned GMRES method until the initial residual is reduced by a factor of 0.1 which typically takes only a small number of iterations. The solution of the standard block is replaced by a GMRES method preconditioned with the space-time two-grid strategy explained before. Also here the GMRES methods is used until the initial residual is reduced by a factor of 0.1.

In table 3.5.6 the number of outer iterations are shown corresponding to the block decomposition. Furthermore the maximum number of inner iterations for the GMRES method of the two-grid preconditioned standard block and the diagonally preconditioned

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 100$					$\alpha = (8, 4) \cdot 10^{-2}, \beta = (1, 7/4), \lambda = 100$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	15(2/11)	12(3/11)	14(3/12)	17(6/12)	16	14(3/12)	11(3/12)	11(3/13)	11(2/15)
32	19(3/20)	14(3/14)	12(2/12)	13(2/14)	32	18(6/19)	15(4/17)	13(4/14)	12(3/16)
64	28(6/31)	20(4/21)	15(3/17)	12(3/15)	64	29(11/27)	22(9/22)	17(7/19)	13(5/15)

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 20$					$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7/4), \lambda = 1000$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	22(2/ 7)	18(3/ 7)	17(3/7)	20(6/6)	16	9(2/18)	8(3/18)	10(3/20)	12(6/24)
32	28(3/10)	21(2/ 8)	17(2/7)	16(2/7)	32	12(3/38)	10(3/31)	9(3/28)	9(2/29)
64	42(6/16)	31(3/11)	22(3/9)	17(2/7)	64	20(6/74)	14(4/55)	11(3/43)	10(3/36)

$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 3.5), \lambda = 100$					$\alpha = (8, 4) \cdot 10^{-3}, \beta = (1, 7), \lambda = 100$				
$n_s \backslash n_t$	20	40	80	160	$n_s \backslash n_t$	20	40	80	160
16	14(3/12)	13(2/11)	14(3/13)	17(6/12)	16	14(3/13)	13(3/12)	15(3/14)	18(4/13)
32	20(3/21)	15(3/16)	13(2/13)	14(2/15)	32	22(3/22)	17(2/17)	15(3/13)	15(2/15)
64	33(6/31)	22(4/22)	16(3/18)	13(2/14)	64	37(6/37)	25(4/22)	19(3/18)	16(2/14)

Table 3.5.6: Maximum outer iteration numbers (block solver) and maximum iteration numbers for the two-grid preconditioned solve for the standard block.

XFEM block is shown in the brackets. We observe that the number of iterations is similar to the exact block preconditioning discussed before, although the numbers are slightly higher. The number of inner iterations stay within very reasonable bounds except for the case $\lambda = 1000$ where the diagonal scaling for the XFEM block require many additional iterations. The results indicate that this overall preconditioner is suitable for the considered test case and the considered parameter range. In the next section we summarize and discuss the results for this and the diagonal preconditioner.

3.5.7 Discussion of results

We briefly summarize and discuss the results of the numerical study carried out in this section. We performed tests for simple and more sophisticated preconditioning strategies for the Space-Time-DG Nitsche-XFEM method. In the previous investigations we observed the following behavior:

- As soon as we apply a diagonal preconditioning the results appear to be essentially independent on the cut position. Furthermore, for λ not too large the results are comparable to a discretization without XFEM function. We conclude that all preconditioners discussed here are *robust w.r.t. the interface position*.
- For the simple diagonal preconditioner we observe that the conditioning of the problem depends on the ratio $h^2/\Delta t$. This behavior has also been observed

for method of lines discretizations of other parabolic problem, see for instance section 2.4.6.3. Also the more sophisticated preconditioners display a dependency on the ratio of h and Δt .

- The block decomposition into XFEM and standard space-time finite element space gives good results. In contrast to the elliptic case in section 2.4, however, the results are *not robust* w.r.t. the mesh size.
- The same statement essentially also holds for the two-grid space-time preconditioning. This preconditioning strategy shows some significant improvement over diagonal preconditioning. Nevertheless, the results are also not completely robust w.r.t. the mesh size.

Remark 3.5.1. *We briefly comment on the computational efficiency of the combined method in section 3.5.6. The performance of the proposed preconditioning strategy essentially depends on the efficiency of the solver for the coarse grid problem (related to the matrix $\mathbf{A}_c$). As mentioned before, this matrix block corresponding to standard unknowns is similar to a system matrix of a backward Euler method and thus should allow for an efficient use of a spatial multigrid preconditioner (at least in the diffusion dominated case). We expect that the inner most iterations which correspond to the solution of the lowest order standard block can be replaced by one (or a small number) of multigrid steps. This would essentially render the costs for the coarse grid solution linear in the number unknowns.*

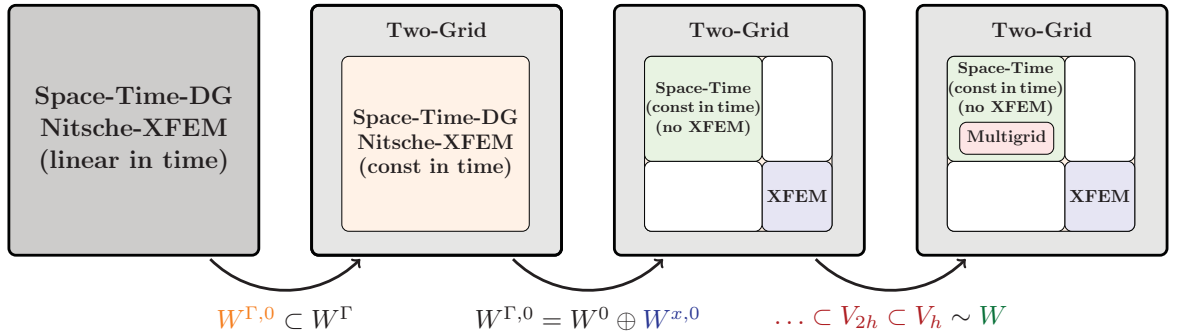


Figure 3.5.2: Sketch of an alternative precondition strategy for Space-Time-DG Nitsche-XFEM

Remark 3.5.2 (Open problems). *As depicted in figure 3.5.1 the main idea of the presented preconditioner is based on a decomposition of W^{Γ} into W and W^x . Then, the tensor product structure of the space-time basis functions is used to construct a two-grid preconditioner. One could also change the order of these preconditioning concepts as is depicted in figure 3.5.2. Here, the two-grid preconditioner is used on the outer level. Then, the inner system of the coarse grid block is similar to a backward Euler method with a Nitsche-XFEM discretization. One could thus try a block preconditioning of this problem using the decomposition of $W^{\Gamma,0}$ into W^0 and $W^{x,0}$. Finally on the space W^0 , which can be identified with V_h , a geometrical multigrid can be applied.*

CHAPTER 4

Numerical integration on implicitly defined domains

Throughout the previous chapters we assumed that integration on the subdomains Ω_i (or Q_i) and on the interface Γ (or Γ_*) can be done exactly. This, however, is typically not true. In many applications the interface is described implicitly via some phase indicator function. The indicator can be a volume fraction (e.g. used in the Volume-of-Fluid (VoF) method) or an artificial function for the description of the interface location (e.g. used in level set method, cf. section 5.1.2). In the following we assume that the zero level of such a level set function defines the interface Γ . As an exact integration on Γ or the subdomains Ω_i is practically not feasible we have to apply suitable numerical integration strategies.

In this section we consider a strategy which approximates the implicit interface in a way which allows for an explicit representation of the approximated interface and subdomains. The approximation then allows to apply suitable (standard) quadrature rules. As standard quadrature rules can be applied, positive weights can be ensured which circumvents stability issues stemming from the geometry approximation of the interface. The complexity of this task very much depends on the dimension of the interface. While conceptionally, the strategy of how to approximate the interface is very similar in two and three dimensions and for the stationary and the space-time interfaces, the details of the construction of an approximation can be of varying complexity. Especially the construction of an approximated space-time interface $\Gamma_{*,h}$ for the spatially three dimensional case is involved.

For the case of a stationary interface and spatial dimension $d \leq 3$ the same or very similar strategies have been presented in [MKOW12, Mül14] and [GR11, Section 7.3]. In this chapter we present numerical strategies for the numerical integration in the spatially two- and three-dimensional case in space and space-time. The essential new aspect in this work with respect to the literature is the space-time setting for the spatially three-dimensional case.

A drawback of the considered approach is that typically only piecewise planar approximations are constructed and thus only second order accurate numerical integration is obtained. An alternative strategy which tries to overcome this problem has been proposed in [MKO13]. We briefly comment on this approach. The idea is to devise a strategy which automatically generates quadrature rules for implicitly given domains (and interfaces) by means of fitting sufficiently many moments for which the exact evaluation of the integrals can be constructed. These approaches are supposed to achieve arbitrarily high order accuracy for (piecewise) sufficiently smooth functions. The integration rules resulting from such a strategy in [MKO13] have the disadvantage that they can not guarantee positiveness which can lead to mass matrices with negative eigenvalues and similar effects. Further no theoretical analysis exists for this method and an extension into the four-dimensional (space-time) case has not been investigated in the literature.

Outline of this chapter

First, in section 4.1 we present the basic strategy of how we construct approximations of the interface and the subdomains. This is done for the stationary case and the space-time case. Next, in section 4.2 we briefly summarize which integral types are necessary to implement the Nitsche-XFEM method (or the Space-Time-DG Nitsche-XFEM method) in order to collect the requirements that the numerical integration strategy has to meet. The final step within this strategy is the decomposition of a simplex (stationary case) or a prism (space-time case) intersected by a (hyper-)plane into simplices. This step is specifically discussed in section 4.3 and section 4.4. While in section 4.3 the total dimension $d_{\text{tot}} = d(+1)$ is assumed to be smaller or equal to three, section 4.4 discusses the more involved four dimensional situation ($d_{\text{tot}} = 3+1 = 4$). At the end the approximation strategy provides a decomposition into one, two, three or four dimensional simplices which are either completely on the interface or not intersected by the interface. On those simplices quadrature has to be applied to evaluate integrals. For one, two and three dimensional simplices the problem of quadrature is standard. The quadrature on four dimensional simplices (pentatopes), however, is less common. We comment on quadrature rules on pentatopes (4D simplices) and technical details for the handling of the weighting factor (measure ratio) $\nu(\mathbf{s}), \mathbf{s} \in \Gamma_* \subset \mathbb{R}^4$ in section 4.5.

4.1 Approximation of implicitly defined domains

For the numerical integration on the (space-time) interface and the (space-time) subdomains we first need to construct a suitable approximation of the (space-time) interface. This is done on each element T (or $\mathcal{Q}^T$) separately. In the following we assume that a smooth scalar function ϕ , denoted as the level set function (cf. section 5.1.2) is given on each element T (or $\mathcal{Q}^T$) such that $\Gamma = \{\mathbf{x} \in T, \phi(\mathbf{x}) = 0\}$ (or $\Gamma_* = \{(\mathbf{x}, t) \in \mathcal{Q}^T, \phi(\mathbf{x}, t) = 0\}$).

We first discuss the approximation of the interface needed for the treatment of the stationary case in section 4.1.1. For the space-time case an according strategy to construct a suitable approximation is presented in section 4.1.2. In section 4.1.3 we comment on properties and possible enhancements of the presented approach. In the subsequent sections the resulting approximation of the (space-time) interface will be denoted as Γ_h ($\Gamma_{*,h}$) and the (space-time) domains as $\Omega_{i,h}$ ($Q_{i,h}$).

4.1.1 Approximation of implicit space domains

For the case of a stationary interface a solution strategy for the approximation of implicit domains has been discussed in [GR11, Chapter 7.3]. We briefly recall the idea.

We consider an element with characteristic length h and apply regular subdivisions dividing every edge into m sub-edges which results in a decomposition into m^d sub-simplices. On each sub-simplex the level set function is evaluated linearly (e.g. by interpolation on the vertices) resulting in a planar zero-level on that sub-simplex. The sub-simplex is then divided into the corresponding two (convex) polygons. Hence, a continuous piecewise planar approximation of the interface is constructed. To apply quadrature on the approximated sub-domains we divide the polygons further into simplices on which standard quadrature formulas can then be applied. The decomposition is explained in section 4.3.

4.1.2 Approximation of implicit space-time domains

We discuss the construction of an approximation of the space-time interface. This is done similarly to the ideas presented in the previous section.

We consider the prism Q^T with a characteristic spatial length h of T (for instance the diameter) and the time step size $\Delta t = t_n - t_{n-1}$. We apply regular subdivisions in time and space. Each edge of T is divided into m_s parts of equal length and the time interval is divided into m_t parts (see figure 4.1.1). We get $m_t \cdot m_s^d$ smaller prisms $\{Q_k\}$ with spatial resolution h/m_s and temporal resolution $\Delta t/m_t$.

Each (smaller) prism Q_k is subdivided into $d + 1$ ($d+1$)-simplices $\{P_l\}$ (For details we refer to section 4.3.3 for $d = 2$ and section 4.4 for $d = 3$). On P_l the level-set function is interpolated as a linear function in space-time (by simply evaluating the vertex values only). As the level-set function is now represented as a linear function on each simplex, the according approximation of the zero-level of the level-set function is piecewise planar (and continuous within Q^T).

If Q^T is intersected some simplices P_j within the decomposition $Q^T = \{P_j\}$ are intersected by a planar approximation of the interface. Using the simplex and the (hyper-) plane one can find a decomposition of P_j into simplices $\{P_j^{(m)}\}$ which are no longer intersected and

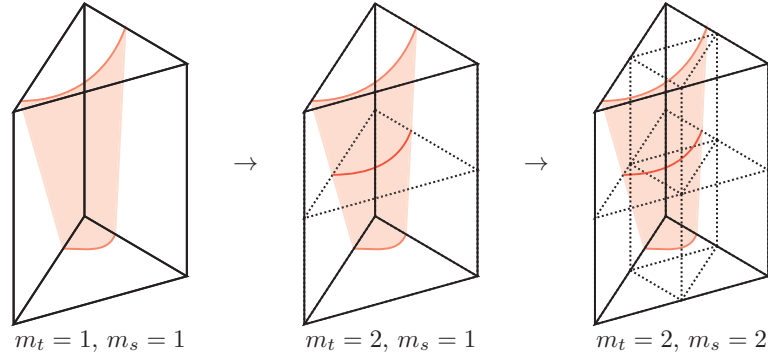


Figure 4.1.1: An intersected prism in $d + 1$ dimensions, with $d = 2$. The original prism (left), the prisms after uniform subdivision in time (middle) and after subdivision in space and time (right).

form a decomposition of $\mathcal{P}_j$, $\mathcal{P}_j = \bigcup_m \mathcal{P}_j^{(m)}$ (cf. section 4.3.3 for $d = 2$ and section 4.4.4 for $d = 3$). Furthermore the plane intersecting one simplex $\mathcal{P}_j$ can also be decomposed into uncut d -dimensional simplices. As these decompositions are neither obvious nor standard for the case $d = 3$ a solution strategy is presented in detail in section 4.4. With this strategy one obtains an explicit decomposition of $\mathcal{Q}_i^T$ into uncut $(d+1)$ -dimensional simplices and of $\Gamma_{*,h}$ into d -dimensional simplices.

4.1.3 Remarks on piecewise planar approximation of implicitly defined domains

We comment on properties and possible modifications of the strategy.

Remark 4.1.1 (Approximation quality). *For smooth (space-time) interfaces both, the piecewise planar approximations in space and space-time, result in an approximation of the interface which is second order accurate in h/m_s and $\Delta t/m_t$.*

Remark 4.1.2 (Adaptive strategy). *Instead of a uniform subdivision one can easily devise a strategy to perform the piecewise planar approximation in an adaptive manner. We explain this for the stationary case. The extension to the space-time setting is straight-forward. We divide the simplex into 2^d sub-simplices first and check on each of those if the level set function has a zero level inside it. If this is the case we refine the sub-simplex further, otherwise not. This is repeated until the edge length of the smallest sub-simplices reaches a desired size. For the same final edge length the adaptive and the uniform refinement strategy give the same approximate interface. However the number of simplices for the subdomains can be significantly reduced with the adaptive strategy. In figure 4.1.2 the adaptive strategy for the starfish example in section 2.5.2 is shown for one, two and three levels of adaptive refinements. Note that in this figure the subdivision of the (convex) polygons into simplices (cf. section 4.3) is already performed.*

Remark 4.1.3 (Subdivision into simplices). *The strategy to decompose every upcoming (uncut) geometry into simplices simplifies the handling of the resulting structure. Only on simplices quadrature rules are applied. This approach is, however, not necessary.*

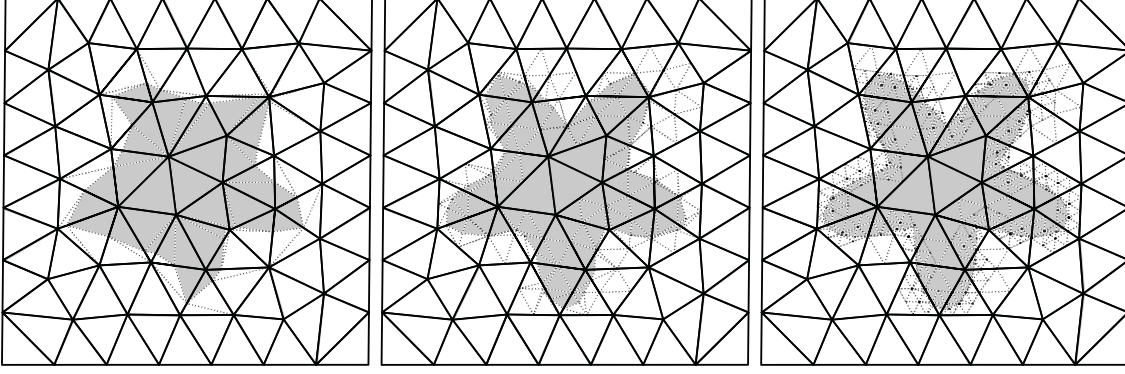


Figure 4.1.2: Construction of an approximate interface for the starfish problem in section 4.3 using the adaptive strategy in remark 4.1.2 for three levels of refinements (on each element).

Quadrature rules could also be defined on the convex polygons arising from the intersection which would typically result in less quadrature points. This is, however, more technical, especially for the spatially three dimensional space-time case.

Remark 4.1.4 (Small angles). *The resulting simplices in the decompositions can have arbitrary small angles. Note that this does not lead to stability problems as we are using the decomposition only for the purpose of numerical integration.*

4.2 Integral types

As we need to calculate (approximations of) integrals of different kinds, we categorize these integrals before we discuss their numerical treatment in section 4.3 and section 4.4. We distinguish those integrals in terms of the sets S we are integrating on. The cases are denoted as **case** ($\mathbf{m}, \mathbf{n}, \mathbf{o}$) where $\mathbf{m}$ is the dimension of S , $\mathbf{n}$ is the co-dimension of S and $\mathbf{o} \in \{\mathbf{c}, \mathbf{n}\}$ describes if the set S is cut by the approximate (space-time) interface Γ_h ($\Gamma_{*,h}$) ($\mathbf{o}=\mathbf{c}$) or not ($\mathbf{o}=\mathbf{n}$).

4.2.1 Stationary interface

In this part we list all integral types that are needed to implement the Nitsche-XFEM discretization in (2.2.41). We recall the notation $T_i = T \cap \Omega_{i,h}$.

4.2.1.1 d -dimensional measure, co-dimension 0

Integrals appearing on each element for $a(\cdot, \cdot)$, $b(\cdot, \cdot)$, $s^{\text{SD}}(\cdot, \cdot)$ and $f(\cdot)$ are integrals on d -dimensional objects like

$$\int_{T_i} f \, d\mathbf{x}$$

We distinguish two different situations: The simplex T is not intersected by the (approximated) interface. We consider this as **case (d,0,n)**. If on the other hand the simplex is intersected by the (approximated) interface, the handling of geometry T_i is more involved. This is denoted by **case (d,0,c)**.

4.2.1.2 $d-1$ -dimensional measure, co-dimension 1

For the space-time integrals stemming from the Nitsche stabilization bilinear form $N(\cdot, \cdot)$ on each element we get terms like

$$\int_{\Gamma_T} f \, d\mathbf{s}.$$

where $\Gamma_T = \Gamma_h \cap T$. Some terms (in $N_c(\cdot, \cdot)$) also depend on the normal direction $\mathbf{n}_\Gamma$. These integrals only appear on elements that are intersected. The measure is $d-1$ -dimensional on the manifold Γ_h with co-dimension 1. This case is denoted as **case (d-1,1,c)**.

4.2.2 Space-time interface

In this part we list all integral types that are need to implement the Space-Time-DG Nitsche-XFEM discretization in its form of a time-slab problem in (3.2.20). We use the notation for a prism $\mathcal{Q}^T = T \times I_n$. Accordingly we define $\mathcal{Q}_i^T = \mathcal{Q}^T \cap Q_{i,h}$.

4.2.2.1 $d+1$ -dimensional measure, co-dimension 0

Integrals appearing on each (prism) element for $a^n(\cdot, \cdot)$, $d^n(\cdot, \cdot)$ and $f(\cdot)$ in (3.2.20) or $s^{\text{SD}}(\cdot, \cdot)$ in (3.2.32) are integrals on $d+1$ -dimensional objects like

$$\int_{\mathcal{Q}_i^T} f \, d\mathbf{x} = \int_{t_{n-1}}^{t_n} \int_{T_i(t)} f \, d\mathbf{x} \, dt, \quad \text{with } T_i(t) = T \cap \Omega_{i,h}(t)$$

We distinguish two different situations: The prism $\mathcal{Q}_i^T$ is not intersected by the (approximated) interface, i.e. the prism is completely in one phase and thus the volume to integrate on is the prism itself. We consider this as **case (d+1,0,n)** where numerical integration can exploit the tensor product structure. If on the other hand the prism $\mathcal{Q}_i^T$ is intersected by the (approximated) interface, the geometry $\mathcal{Q}_i^T$ is much more difficult to handle. In that case $d+1$ -dimensional quadrature on subsimplices has to be applied. This is denoted by **case (d+1,0,c)**.

4.2.2.2 d -dimensional measure, co-dimension 0

The integrals in the element contributions of $\hat{b}^n(\cdot, \cdot)$ and $c^n(\bar{u}; \cdot)$ in (3.2.20) have the form

$$\int_{T_i(t_{n-1})} f d\mathbf{x}$$

and thus are d -dimensional measures. Also here, we distinguish the case of a one phase element (i.e. an element which is not intersected), denoted by **case** (d,0,n) and the case of an intersected element, **case** (d,0,c). The problem is essentially the same as for the stationary problem in section 4.2.1.1.

4.2.2.3 d -dimensional measure, co-dimension 1

For the space-time integrals stemming from the Nitsche stabilization bilinear form $N^n(\cdot, \cdot)$ on each element we get terms like

$$\int_{\Gamma_*^T} \nu(\mathbf{s}) \cdot d\mathbf{s}.$$

where $\Gamma_*^T = \Gamma_{*,h} \cap Q^T$. Some terms also depend on the normal direction $\mathbf{n}$. These integrals only appear on elements that are intersected. The measure is d -dimensional on the manifold $\Gamma_{*,h}$ with co-dimension 1. This case is denoted as **case** (d,1,c).

4.2.3 Summary of cases

In table 4.2.1 all relevant cases for the Nitsche-XFEM and the Space-Time-DG Nitsche-XFEM method are summarized and references to subsequent sections where the corresponding numerical treatments are discussed are given.

case	2D	3D	ST2D	ST3D	num. treatment
case(2,0,n)	X	–	X *	–	standard rules
case(2,0,c)	X	–	X *	–	see section 4.3.1
case(3,0,n)	–	X	X	X *	standard rules
case(3,0,c)	–	X	X	X *	see section 4.3.2 (+ section 4.3.3)
case(4,0,n)	–	–	–	X	standard rules
case(4,0,c)	–	–	–	X	see section 4.4.2-4.4.4
case(1,1,c)	X	–	–	–	see section 4.3.1
case(2,1,c)	–	X	X	–	see section 4.3.2 (+ section 4.3.3)
case(3,1,c)	–	–	–	X	see section 4.4.2-4.4.4

Table 4.2.1: Overview on integral types appearing in the Nitsche-XFEM method (2D / 3D) and the Space-Time-DG Nitsche-XFEM method (ST2D / ST3D). Marks with an asterisk label terms stemming from the DG coupling of the space-time method.

4.3 A strategy to decompose intersected 3-simplices or 3-prisms into simplices

We briefly explain how the subdivision of the (convex) polygons into simplices generated in the strategy introduced in section 4.1.1 and section 4.1.2 can be done for the case that $d_{\text{tot}} \leq 3$, i.e. for a stationary problem with $d = 2, 3$ or a space-time problem with $d = 2$. The much simpler case $d = 1$ is not discussed.

4.3.1 2D stationary case

For the two-dimensional case the simplex T is a triangle $K = T$. The approximated interface is a straight line inside this triangle. We denote the coordinates of the vertices of the current element as $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$, such that $K = \text{conv}(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)$. There exists $\mathbf{c}^1, \mathbf{c}^2 \in \partial K$ such that there holds $\Gamma_h = \text{conv}(\mathbf{c}^1, \mathbf{c}^2)$. Without loss of generality we assume $\mathbf{c}^1 \in \text{conv}(\mathbf{x}^1, \mathbf{x}^3)$, $\mathbf{c}^2 \in \text{conv}(\mathbf{x}^2, \mathbf{x}^3)$ and $\mathbf{x}^1, \mathbf{x}^2 \in \Omega_{1,h}$. Then we have

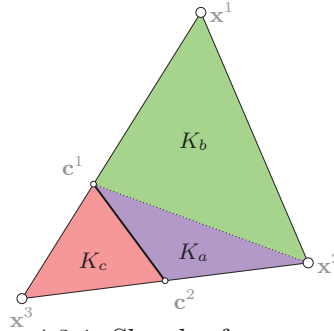


Figure 4.3.1: Sketch of a cut triangle.

$K_1 = \Omega_{1,h} \cap K = K_a \cup K_b$ with $K_a = \text{conv}(\mathbf{x}^1, \mathbf{x}^2, \mathbf{c}^1)$ and $K_b = \text{conv}(\mathbf{x}^2, \mathbf{c}^1, \mathbf{c}^2)$. Accordingly we have $K_2 = \Omega_{2,h} \cap K = K_c = \text{conv}(\mathbf{c}^1, \mathbf{c}^2, \mathbf{x}^3)$. The situation is sketched in figure 4.3.1. Note that for the cases where one vertex is intersected or an edge coincides with Γ_h some of the resulting simplices can have measure zero which is, however, not a problem.

4.3.2 3D stationary case

In three dimensions the simplex is a tetrahedron T and the approximated interface is a plane inside this tetrahedron. Let $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4$ be the vertices of T such that $T = \text{conv}(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4)$. The approximated interface Γ_h can either be a triangle or a quadrilateral. We distinguish both cases.

Case 1: Interface Γ_h is a triangle. We consider the case where we can characterize the approximate interface as $\Gamma_h = \text{conv}(\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3)$ with $\mathbf{c}^i \in \text{conv}(\mathbf{x}^i, \mathbf{x}^4)$, $i = 1, 2, 3$ and assume (w.l.o.g.) $\mathbf{x}^4 \in \Omega_{2,h}$. Then we have $T \cap \Omega_{1,h} = \text{conv}(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3)$ (a deformed prism) and $T \cap \Omega_{2,h} = \text{conv}(\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3, \mathbf{x}^4)$ (a tetrahedron $T_d = \text{conv}(\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3, \mathbf{x}^4)$). We decompose the prism into three tetrahedra $T \cap \Omega_{1,h} = T_a \cup T_b \cup T_c$

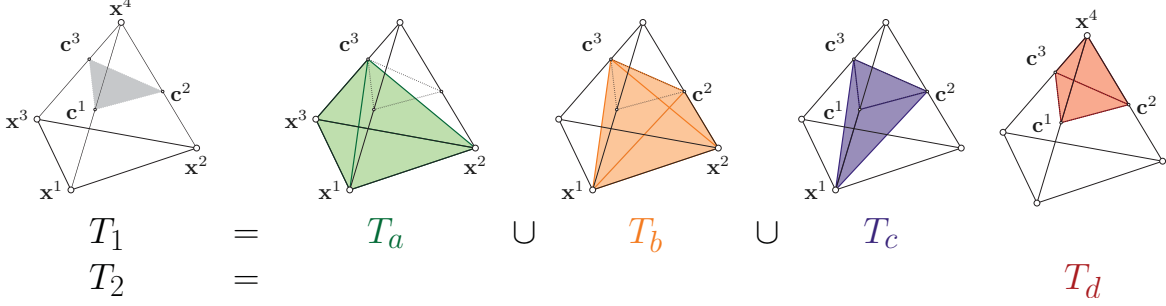


Figure 4.3.2: Sketch of decomposition of tetrahedron cut by a triangle.

with

$$\begin{aligned} T_a &= \text{conv}(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{c}^3), \\ T_b &= \text{conv}(\mathbf{x}^1, \mathbf{x}^2, \mathbf{c}^2, \mathbf{c}^3), \\ T_c &= \text{conv}(\mathbf{x}^1, \mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3). \end{aligned}$$

The decomposition is sketched in figure 4.3.2.

Case 2: Interface Γ_h is a quadrilateral. We consider the case where the approximate interface is a quadrilateral. The vertices are labeled such that $\mathbf{x}_1$ and $\mathbf{x}_2$ are in a different domain as $\mathbf{x}_3$ and $\mathbf{x}_4$. For ease of presentation we introduce the notation $\mathbf{y}^1 = \mathbf{x}^3$ and $\mathbf{y}^2 = \mathbf{x}^4$. Then, we can characterize the approximate interface as $\Gamma_h = \text{conv}(\mathbf{z}^{11}, \mathbf{z}^{12}, \mathbf{z}^{21}, \mathbf{z}^{22})$ with $\mathbf{z}^{ij} \in \text{conv}(\mathbf{x}^i, \mathbf{y}^j)$, $i, j = 1, 2$. We assume (w.l.o.g.) $\mathbf{y}^1, \mathbf{y}^2 \in \Omega_{2,h}$. Both domains $T_1 = \text{conv}(\mathbf{x}^1, \mathbf{z}^{11}, \mathbf{z}^{12}, \mathbf{x}^2, \mathbf{z}^{21}, \mathbf{z}^{22})$ and $T_2 = \text{conv}(\mathbf{y}^1, \mathbf{z}^{11}, \mathbf{z}^{12}, \mathbf{y}^2, \mathbf{z}^{21}, \mathbf{z}^{22})$ are deformed prisms and similar to the decomposition of the deformed prism in case 1 we can devise a decomposition into simplices. We get a subdivision $T_1 = T_a^x \cup T_b^x \cup T_c^x$ with

$$\begin{aligned} T_a^x &= \text{conv}(\mathbf{x}^1, \mathbf{z}^{11}, \mathbf{z}^{12}, \mathbf{z}^{22}), \\ T_b^x &= \text{conv}(\mathbf{x}^1, \mathbf{z}^{11}, \mathbf{z}^{21}, \mathbf{z}^{22}), \\ T_c^x &= \text{conv}(\mathbf{x}^1, \mathbf{x}^2, \mathbf{z}^{21}, \mathbf{z}^{22}), \end{aligned}$$

and $T_2 = T_a^y \cup T_b^y \cup T_c^y$ with

$$\begin{aligned} T_a^y &= \text{conv}(\mathbf{y}^1, \mathbf{z}^{11}, \mathbf{z}^{12}, \mathbf{z}^{22}), \\ T_b^y &= \text{conv}(\mathbf{y}^1, \mathbf{z}^{11}, \mathbf{z}^{21}, \mathbf{z}^{22}), \\ T_c^y &= \text{conv}(\mathbf{y}^1, \mathbf{y}^2, \mathbf{z}^{21}, \mathbf{z}^{22}), \end{aligned}$$

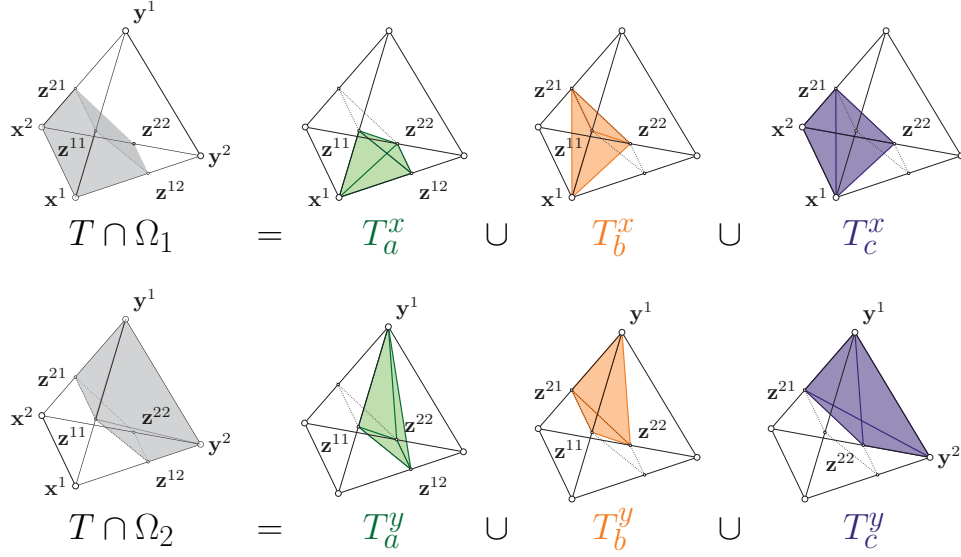


Figure 4.3.3: Sketch of decomposition of tetrahedron cut by a quadrilateral.

cf. figure 4.3.3. The decomposition of the interface into simplices is achieved with the triangles $\text{conv}(\mathbf{z}^{11}, \mathbf{z}^{12}, \mathbf{z}^{22})$ and $\text{conv}(\mathbf{z}^{11}, \mathbf{z}^{21}, \mathbf{z}^{22})$.

Note that for the cases where one vertex or a complete edge is intersected or a face coincides with Γ_h some of the resulting simplices can also have measure zero, which is not a problem.

4.3.3 (2+1)D space-time case

If the space-time method is applied for a spatially two-dimensional problem one has to deal with the decomposition of a prism element intersected by an approximate interface which is represented by the vertex values of the level set function. Therefore we divide the prism $\mathcal{Q}^T = K \times (t_{n-1}, t_n]$ into three tetrahedra and proceed as in section 4.3.2. The prism can be represented as $\mathcal{Q}^T = \text{conv}(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3)$ with $(\mathbf{x}^i)_3 = t_{n-1}$ and $(\mathbf{y}^i)_3 = t_n$, $i = 1, 2, 3$ and thus we can use the decomposition into $K_1 = \text{conv}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{y}_3)$, $K_2 = \text{conv}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_2, \mathbf{y}_3)$ and $K_3 = \text{conv}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$ such that $\mathcal{Q}^T = K_1 \cup K_2 \cup K_3$. On those tetrahedra we interpolate the level set function resulting in a linear level set function and thus a planar approximate interface. Each tetrahedron (if intersected) is then subdivided into tetrahedra according to the rules in section 4.3.2.

4.4 A strategy to decompose intersected 4-prisms into pentatopes

In this section we introduce a decomposition strategy that allows for a decomposition of four dimensional prisms into pentatopes as needed for the cases $(d+1, 0, c)$ and $(d, 1, c)$ in the Space-Time-DG Nitsche-XFEM method. This approach is new and has been published in [Leh15].

Firstly, we introduce the definitions of relevant four dimensional geometries in section 4.4.1. The decomposition of a 4-prism into four pentatopes is presented in section 4.4.2. Note that this decomposition is already needed to construct (via interpolation of the level-set function) the piecewise planar space-time interface as described in section 4.1.

In section 4.4.4 a strategy is presented that allows us to decompose a pentatope which is intersected by a hyperplane (representing an approximation of the space-time interface) into pentatopes which are not intersected. Figure 4.4.1 sketches the algorithmic structure of the decomposition strategy. In this algorithm we need a particular geometrical object, that we call *hypertriangle*, which can be decomposed into six pentatopes following the decomposition rule in section 4.4.3.

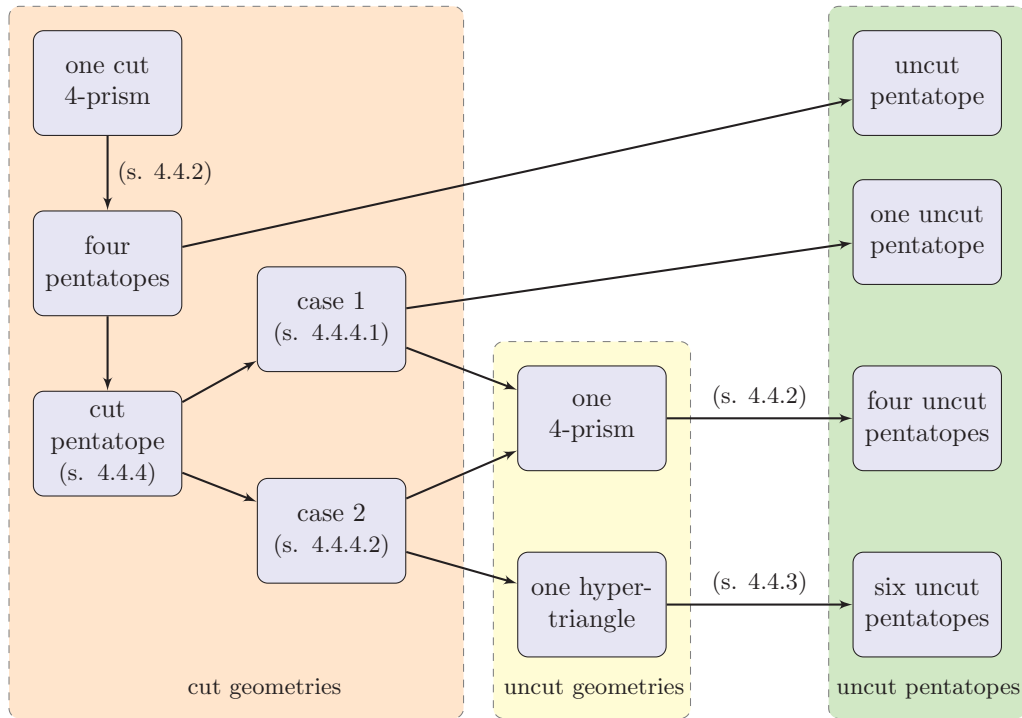


Figure 4.4.1: Algorithmic structure of the decomposition strategy proposed in section 4.4.

4.4.1 Definition of simple geometries in four dimensions

By $\mathbf{e}^i \in \mathbb{R}^n$ we denote the i -th unit vector with $(\mathbf{e}^i)_j = \delta_{i,j}$ for $i = 1, \dots, n$ and $\mathbf{e}^0 := 0$.

Definition 4.4.1 (4-simplex / pentatope). Let $\mathbf{x}^i \in \mathbb{R}^4$ for $i = 1, \dots, 5$ and $\mathbf{d}^{i,j} := \mathbf{x}^i - \mathbf{x}^j$. Iff the vectors $\mathbf{d}^{i,1}$ for $i = 2, \dots, 5$ are linearly independent, we call the convex hull $\mathcal{P} = \text{conv}(\{\mathbf{x}^i\}_{i=1,\dots,5})$ the 4-simplex or pentatope.

Remark 4.4.1 (reference pentatope). Every pentatope $\mathcal{P}$ can be represented as an affine transformation applied to the reference pentatope $\hat{\mathcal{P}} = \text{conv}(\{\mathbf{e}^i\}_{i=0,\dots,4})$. The transformation has the form

$$\Phi : \hat{\mathcal{P}} \rightarrow \mathcal{P}, (\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4) \rightarrow \sum_{i=1}^5 \hat{\lambda}_i(\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4) \mathbf{x}^i,$$

where $\hat{\lambda}_i(\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4)$ is the barycentric coordinate of $\hat{\mathcal{P}}$ with respect to the vertex $\mathbf{e}^{i-1}$.

Definition 4.4.2 (4-prism). Let $\mathbf{x}^i \in \mathbb{R}^4$ for $i = 1, \dots, 4$ and $\mathbf{y} \in \mathbb{R}^4$. Iff $\{\mathbf{x}^i\}_{i=1,\dots,4}$ defines a 3-simplex (tetrahedron) $T = \text{conv}(\{\mathbf{x}^i\}_{i=1,\dots,4})$ and $\mathbf{y}$ is linearly independent of $\{\mathbf{d}^{i,1}\}_{i=2,\dots,4}$, with $\mathbf{d}^{i,j} := \mathbf{x}^i - \mathbf{x}^j$, the set

$$\mathcal{Q} = \text{conv}(\{\mathbf{x}^i\}_{i=1,\dots,4}, \{\mathbf{x}^i + \mathbf{y}\}_{i=1,\dots,4}) = \{\mathbf{x} + \alpha \mathbf{y}, \mathbf{x} \in \text{conv}(\{\mathbf{x}^i\}_{i=1,\dots,4}), \alpha \in [0, 1]\}$$

is called 4-prism.

Remark 4.4.2 (reference 4-prism). Every 4-prism can be represented as an affine linear transformation applied to the reference 4-prism $\hat{\mathcal{Q}} = \text{conv}(\{\mathbf{e}^i\}_{i=0,\dots,3}, \{\mathbf{e}^i + \mathbf{e}^4\}_{i=0,\dots,3})$. The transformation has the form

$$\Phi : \hat{\mathcal{Q}} \rightarrow \mathcal{Q}, (\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4) \rightarrow \sum_{i=1}^4 \hat{\mu}_i(\hat{x}_1, \hat{x}_2, \hat{x}_3) \mathbf{x}^i + \hat{x}_4 \mathbf{y},$$

where $\hat{\mu}_i(\hat{x}_1, \hat{x}_2, \hat{x}_3)$ are the barycentric coordinates of the reference tetrahedron $\hat{T} = \text{conv}(\{\mathbf{e}^i\}_{i=0,\dots,3})$.

The next geometry is a little bit more complex. It later occurs as one part of a pentatope cut by a hyperplane.

Definition 4.4.3 (hypertriangle). We define the reference hypertriangle as

$$\begin{aligned} \hat{\mathcal{H}} &:= \{(x_1, x_2, x_3, x_4) \in \mathbb{R}_+^4, x_1 + x_2 \leq 1, x_3 + x_4 \leq 1\} \\ &= \text{conv}(\{\hat{\mathbf{x}}^{i,j}\}_{i=1,\dots,3,j=1,\dots,3}) = \hat{\mathcal{K}} \times \hat{\mathcal{K}} \end{aligned}$$

where $\hat{\mathcal{K}} \subset \mathbb{R}^2$ denotes the reference triangle $\hat{\mathcal{K}} = \text{conv}(\{\chi^1, \chi^2, \chi^3\}) \subset \mathbb{R}^2$ with $\chi^i = \mathbf{e}^{i-1} \in \mathbb{R}^2$, $i = 1, 2, 3$ and $\hat{\mathbf{x}}^{i,j} = (\chi^i, \chi^j) \in \mathbb{R}^4$, $i, j = 1, 2, 3$. Now, let $\mathbf{x}^{i,j} \in \mathbb{R}^4$, $i, j = 1, 2, 3$. The convex hull $\mathcal{H} = \text{conv}(\{\mathbf{x}^{i,j}\}_{i,j=1,2,3})$ is called a hypertriangle iff there exists a transformation

$$\Phi : \hat{\mathcal{H}} \rightarrow \mathcal{H}, (\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4) \rightarrow \sum_{i=1}^3 \sum_{j=1}^3 \hat{\rho}_i(\hat{x}_1, \hat{x}_2) \hat{\rho}_j(\hat{x}_3, \hat{x}_4) \mathbf{x}^{i,j},$$

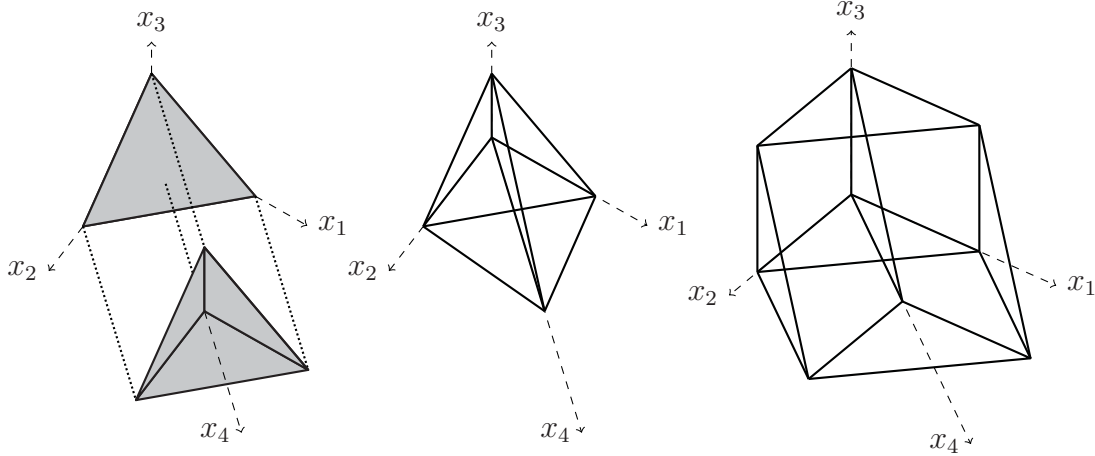


Figure 4.4.2: Sketch of reference geometries. Reference 4-prism $\hat{Q}$ (left), reference pentatope $\hat{P}$ (center) and reference hypertriangle $\hat{H}$ (right). The dotted line in the left picture are parallel to the x_4 -axes and connect the tetrahedra at $x_4 = 0$ and $x_4 = 1$.

where $\hat{\rho}_i(\hat{x}_1, \hat{x}_2)$ is the barycentric coordinate of the reference triangle $\hat{K}$ corresponding to the vertex χ^i . There holds $\Phi(\hat{\mathcal{H}}) = \mathcal{H}$.

Remark 4.4.3 (Sketches). The sketches in figure 4.4.2, figure 4.4.3 and figure 4.4.4 in this section show two dimensional parallel projections of four dimensional objects. Straight lines in the sketch represent a line (in four dimensions) between two vertices. Note that the preimage of a point in the two dimensional sketch of the parallel projection is a two-dimensional set.

4.4.2 Decomposition of a 4-prism into four pentatopes

We consider an arbitrary prism element $\mathcal{Q}^T = T \times I_n$ with a tetrahedral element T and a time interval I_n . For each $\mathcal{Q}^T$ there exists a linear transformation Φ mapping from the reference 4-prism $\hat{Q}$ to $\mathcal{Q}^T$ which is of the form $\Phi(\hat{\mathbf{x}}, \hat{t}) = (\Phi_x(\hat{\mathbf{x}}), \Phi_t(\hat{t}))^T$ with the time transformation $\Phi_t(\hat{t}) = \hat{t} \cdot t_n + (1 - \hat{t}) \cdot t_{n-1}$ and the space transformation $\Phi_x(\hat{\mathbf{x}})$ mapping from the reference tetrahedron $\hat{T}$ to T .

It is sufficient to consider the decomposition of the reference 4-prism $\hat{Q}$ into four pentatopes as applying Φ to each pentatope of this decomposition results in a valid decomposition of $\mathcal{Q}$ into four pentatopes. With $\mathbf{x}^i := \mathbf{e}^{i-1}$ and $\mathbf{y}^i := \mathbf{e}^{i-1} + \mathbf{e}^4$ for $i = 1, \dots, 4$ for the reference 4-prism there holds $\hat{Q} = \text{conv}(\{\mathbf{x}^i\}_{i=1,\dots,4}, \{\mathbf{y}^i\}_{i=1,\dots,4})$. We decompose $\hat{Q}$ into four pentatopes $\hat{P}_1, \hat{P}_2, \hat{P}_3, \hat{P}_4$, which are defined as follows:

$$\begin{aligned} \hat{P}_1 &:= \text{conv}(\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4, \mathbf{y}^4\}), \quad \hat{P}_2 := \text{conv}(\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{y}^3, \mathbf{y}^4\}) \\ \hat{P}_3 &:= \text{conv}(\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4\}), \quad \hat{P}_4 := \text{conv}(\{\mathbf{x}^1, \mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4\}) \end{aligned}$$

A sketch of those can be found in figure 4.4.3. To see that this is a suitable decomposition, we give the following characterization of the pentatopes $\hat{P}_i$ in terms of constrained sets

and their partial sums $\hat{\mathcal{B}}_i = \bigcup_{j=1}^i \hat{\mathcal{P}}_j$:

$$\begin{aligned}\hat{\mathcal{P}}_1 &= \{\mathbf{x} \in \hat{Q}, x_3 \geq x_4\}, \\ \hat{\mathcal{P}}_2 &= \{\mathbf{x} \in \hat{Q}, x_3 \leq x_4, x_3 + x_2 \geq x_4\}, & \hat{\mathcal{B}}_2 &= \{\mathbf{x} \in \hat{Q}, x_3 + x_2 \geq x_4\}, \\ \hat{\mathcal{P}}_3 &= \{\mathbf{x} \in \hat{Q}, x_1 + x_2 + x_3 \geq x_4, x_3 + x_2 \leq x_4\}, & \hat{\mathcal{B}}_3 &= \{\mathbf{x} \in \hat{Q}, x_1 + x_2 + x_3 \geq x_4\}, \\ \hat{\mathcal{P}}_4 &= \{\mathbf{x} \in \hat{Q}, x_1 + x_2 + x_3 \leq x_4\}, & \hat{\mathcal{B}}_4 &= \{\mathbf{x} \in \hat{Q}\}.\end{aligned}$$

One can easily show that the pentatopes are disjoint (except for a part with measure

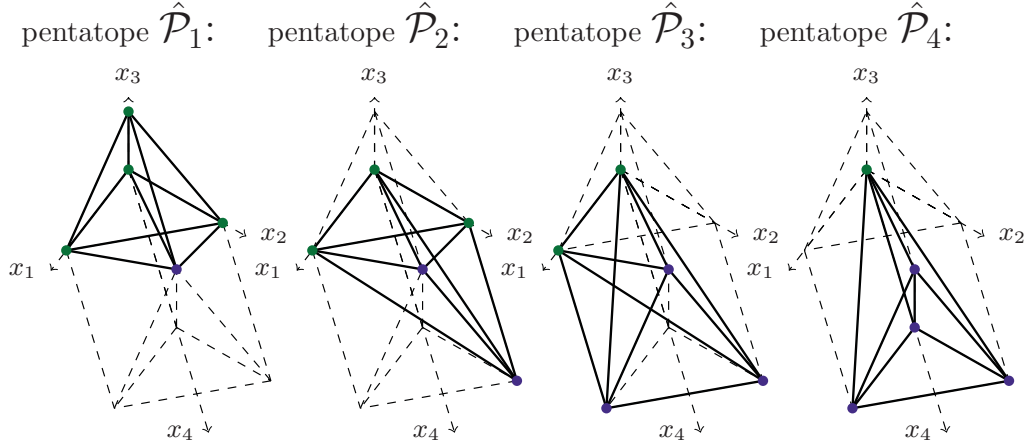


Figure 4.4.3: Sketch of pentatopes $\hat{\mathcal{P}}_1, \hat{\mathcal{P}}_2, \hat{\mathcal{P}}_3, \hat{\mathcal{P}}_4$ which form a valid decomposition of the reference 4-prism $\hat{Q}$.

zero) and sum up to the reference prism: $\bigcup \hat{\mathcal{P}}_i = \hat{Q}$. Note further that the measure of all pentatopes is the same, i.e. $\text{meas}_4(\mathcal{P}_i) = 1/24$.

4.4.3 Decomposing the reference hypertriangle

Let $\mathbf{u}^i = \hat{\mathbf{x}}^{1,i}$, $\mathbf{v}^i = \hat{\mathbf{x}}^{2,i}$, $\mathbf{w}^i = \hat{\mathbf{x}}^{3,i}$, $i = 1, \dots, 3$ with $\hat{\mathbf{x}}^{i,j}$ as in definition 4.4.3. We decompose $\hat{\mathcal{H}}$ into six pentatopes which are defined as follows:

$$\begin{aligned}\hat{\mathcal{D}}_u &= \text{conv}(\{\underline{\mathbf{u}}^1, \underline{\mathbf{u}}^2, \underline{\mathbf{u}}^3, \mathbf{v}^2, \mathbf{w}^3\}), & \hat{\mathcal{D}}_v &= \text{conv}(\{\mathbf{u}^1, \underline{\mathbf{v}}^1, \underline{\mathbf{v}}^2, \underline{\mathbf{v}}^3, \mathbf{w}^3\}), \\ \hat{\mathcal{D}}_w &= \text{conv}(\{\mathbf{u}^1, \mathbf{v}^2, \underline{\mathbf{w}}^1, \underline{\mathbf{w}}^2, \underline{\mathbf{w}}^3\}), & \hat{\mathcal{D}}_1 &= \text{conv}(\{\underline{\mathbf{u}}^1, \underline{\mathbf{v}}^1, \mathbf{v}^2, \underline{\mathbf{w}}^1, \mathbf{w}^3\}), \\ \hat{\mathcal{D}}_2 &= \text{conv}(\{\mathbf{u}^1, \underline{\mathbf{u}}^2, \underline{\mathbf{v}}^2, \underline{\mathbf{w}}^2, \mathbf{w}^3\}), & \hat{\mathcal{D}}_3 &= \text{conv}(\{\mathbf{u}^1, \underline{\mathbf{u}}^3, \mathbf{v}^2, \underline{\mathbf{v}}^3, \underline{\mathbf{w}}^3\}).\end{aligned}$$

Note that there is a simple structure behind this decomposition. We define the “diagonal triangle” as $\hat{K}_{\text{diag}} = \text{conv}(\mathbf{u}^1, \mathbf{v}^2, \mathbf{w}^3) = \text{conv}(\hat{\mathbf{x}}^{1,1}, \hat{\mathbf{x}}^{2,2}, \hat{\mathbf{x}}^{3,3})$. To the three vertices of $\hat{K}_{\text{diag}}$ we add the missing vertices (underlined) of one of the following six triangles and form the convex hull.

$$\begin{aligned}\hat{K}_u &= \text{conv}(\{\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3\}), & \hat{K}_v &= \text{conv}(\{\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3\}), & \hat{K}_w &= \text{conv}(\{\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3\}), \\ \hat{K}_1 &= \text{conv}(\{\mathbf{u}^1, \mathbf{v}^1, \mathbf{w}^1\}), & \hat{K}_2 &= \text{conv}(\{\mathbf{u}^2, \mathbf{v}^2, \mathbf{w}^2\}), & \hat{K}_3 &= \text{conv}(\{\mathbf{u}^3, \mathbf{v}^3, \mathbf{w}^3\}).\end{aligned}$$

4.4 A strategy to decompose intersected 4-prisms into pentatopes

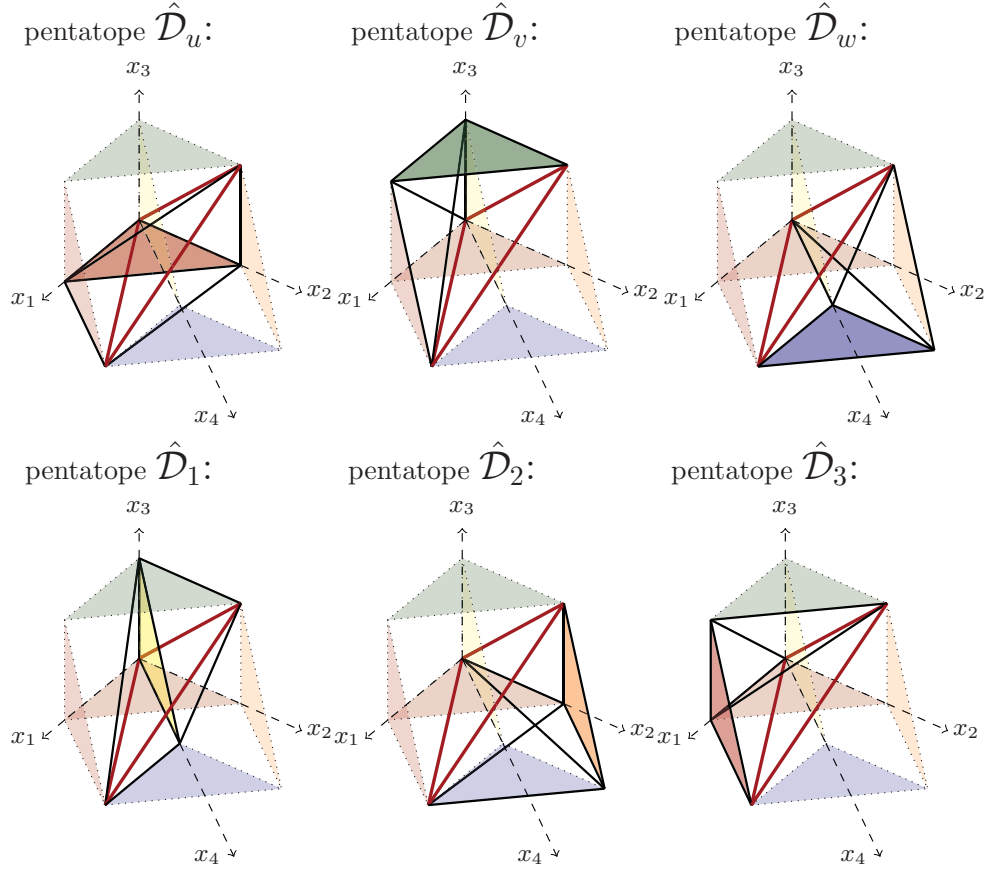


Figure 4.4.4: Sketch of pentatopes $\hat{\mathcal{D}}_u, \hat{\mathcal{D}}_v, \hat{\mathcal{D}}_w, \hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2, \hat{\mathcal{D}}_3$ which form a valid decomposition of the reference hypertriangle $\hat{\mathcal{H}}$. The edges of $\hat{K}_{\text{diag}}$ are highlighted in red, whereas the triangles $\hat{K}_u, \hat{K}_v, \hat{K}_w, \hat{K}_1, \hat{K}_2, \hat{K}_3$ are filled with the corresponding color. The triangle corresponding to each pentatope is highlighted.

A sketch of those pentatopes is given in figure 4.4.4. Also here, one can show that the pentatopes are disjoint (except for a part with measure zero), and sum up to $\hat{\mathcal{H}}$. To this end we divide the hypertriangle according to three binary decisions and define

$$\begin{aligned} \hat{\mathcal{H}}^{i,j,k} &:= \hat{\mathcal{H}} \cap \{(-1)^i x_2 \leq (-1)^i x_4\} \cap \{(-1)^j x_1 \leq (-1)^j x_3\} \\ &\quad \cap \{(-1)^k (x_1 + x_2) \leq (-1)^k (x_3 + x_4)\}, \quad i, j, k = 0, 1 \end{aligned}$$

Note that $\hat{\mathcal{H}}^{1,1,0}$ and $\hat{\mathcal{H}}^{0,0,1}$ are sets with measure zero. All other sets can be identified with a pentatope from the decomposition:

$$\hat{\mathcal{H}}^{1,1,1} = \hat{\mathcal{D}}_u, \quad \hat{\mathcal{H}}^{1,0,0} = \hat{\mathcal{D}}_v, \quad \hat{\mathcal{H}}^{0,1,0} = \hat{\mathcal{D}}_w, \quad \hat{\mathcal{H}}^{0,0,0} = \hat{\mathcal{D}}_1, \quad \hat{\mathcal{H}}^{0,1,1} = \hat{\mathcal{D}}_2, \quad \hat{\mathcal{H}}^{1,0,1} = \hat{\mathcal{D}}_3.$$

4.4.4 Decomposition of a pentatope intersected by the space-time interface into uncut pentatopes

We assume that the space-time interface is approximated in a piecewise planar fashion, s.t. within each pentatope the space-time interface is a (hyper-)plane. This plane divides a pentatope into two parts. Note that due to the pentatope being a convex set each of the two parts will still be convex. We now consider a pentatope $\mathcal{P}$ which is cut by the plane $\mathcal{G} = \{\mathbf{x} \in \mathbb{R}^4 : \mathbf{x} \cdot \mathbf{n}_{\mathcal{G}} = c\}$ which represents the local approximation of the space-time interface. Each vertex $\mathbf{v}$ is marked corresponding to one of the two half spaces. Vertices with $\mathbf{v} \cdot \mathbf{n}_{\mathcal{G}} < c$ are marked with a plus (+), all others with a minus (-). Note that this classification includes the cases where the space-time interface hits vertices ($\mathbf{v} \cdot \mathbf{n}_{\mathcal{G}} = c$). We thus can only have two non-trivial situations:

- case 1: One vertex has a sign that is different from all the others or
- case 2: Two vertices have a sign that is different from the other three vertices.

In the following we will consider these cases separately and construct a decomposition of the parts into pentatopes. Without loss of generality we assume that the vertices in the smaller group of vertices are those marked with a plus (+).

4.4.4.1 Case 1: Decomposition into one pentatope and one 4-prism

We consider the case where one vertex of a pentatope, say $\mathbf{x}^5$, is marked with a plus (+). All other vertices ($\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4$) are marked with a minus (-). The cutting points of the hyperplane $\mathcal{G}$ with the edges are $\mathbf{b}^1 := \overline{\mathbf{x}^1\mathbf{x}^5} \cap \mathcal{G}, \mathbf{b}^2 := \overline{\mathbf{x}^2\mathbf{x}^5} \cap \mathcal{G}, \mathbf{b}^3 := \overline{\mathbf{x}^3\mathbf{x}^5} \cap \mathcal{G}, \mathbf{b}^4 := \overline{\mathbf{x}^4\mathbf{x}^5} \cap \mathcal{G}$. The geometry containing the separated vertex is the pentatope $\mathcal{P}^+ := \text{conv}(\{\mathbf{b}^1, \mathbf{b}^2, \mathbf{b}^3, \mathbf{b}^4, \mathbf{x}^5\})$ while the remainder is $\mathcal{Q}^- := \text{conv}(\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4, \mathbf{b}^1, \mathbf{b}^2, \mathbf{b}^3, \mathbf{b}^4\})$. Consider the mapping

$$\Phi : \hat{\mathcal{Q}} \rightarrow \mathcal{Q}^-, (\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4) \rightarrow \sum_{i=1}^4 \mu_i(\hat{x}_1, \hat{x}_2, \hat{x}_3)(\hat{x}_4 \mathbf{b}^i + (1 - \hat{x}_4) \mathbf{x}^i)$$

with $\hat{\mu}(\hat{x}_1, \hat{x}_2, \hat{x}_3)$ the barycentric coordinates of the reference tetrahedron $\hat{T}$. The decomposition of the reference 4-prism $\hat{\mathcal{Q}}$ into the four pentatopes $\hat{\mathcal{P}}_i$, $i = 1, \dots, 4$ as described in section 4.4.2 can be used as a triangulation of $\hat{\mathcal{Q}}$. Let Φ_h be the (pentatope-) piecewise linear interpolation of Φ at the vertices of this triangulation. Then Φ_h is an isomorphism between $\hat{\mathcal{Q}}$ and $\mathcal{Q}^-$. This is due to the fact that with the linearity of Φ_h on each pentatope $\hat{\mathcal{P}}_i$ the mapping Φ_h is an isomorphism between $\hat{\mathcal{P}}_i$ and its image $\Phi_h(\hat{\mathcal{P}}_i)$ which is again a pentatope. As the pentatopes $\hat{\mathcal{P}}_i$ or $\Phi_h(\hat{\mathcal{P}}_i)$, respectively, are non-overlapping it follows that Φ_h is an isomorphism. Thus the decomposition rule for the reference 4-prism can also be applied here and we get a valid decomposition by taking

the four pentatopes

$$\begin{aligned}\mathcal{P}_1 &= \Phi_h(\hat{\mathcal{P}}_1) = \text{conv}(\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4, \mathbf{b}^4\}), \\ \mathcal{P}_2 &= \Phi_h(\hat{\mathcal{P}}_2) = \text{conv}(\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{b}^3, \mathbf{b}^4\}), \\ \mathcal{P}_3 &= \Phi_h(\hat{\mathcal{P}}_3) = \text{conv}(\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{b}^2, \mathbf{b}^3, \mathbf{b}^4\}), \\ \mathcal{P}_4 &= \Phi_h(\hat{\mathcal{P}}_4) = \text{conv}(\{\mathbf{x}^1, \mathbf{b}^1, \mathbf{b}^2, \mathbf{b}^3, \mathbf{b}^4\}).\end{aligned}$$

Decomposition of the space-time interface into tetrahedra for case 1. The triangulation of the interface is trivially obtained with the tetrahedron

$$\mathcal{P} \cap \mathcal{G} = \mathcal{I} = \text{conv}(\{\mathbf{b}^1, \mathbf{b}^2, \mathbf{b}^3, \mathbf{b}^4\}).$$

4.4.4.2 Case 2: Decomposition into one 4-prism and one hypertriangle

Let us consider the case where two vertices of a pentatope are marked with a plus (+), these are (w.l.o.g.) vertices $\mathbf{x}^4$ and $\mathbf{x}^5$. All other vertices ($\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$) are marked with a minus (-). The cutting points of the hyperplane $\mathcal{G}$ with the edges are $\mathbf{c}^1 := \overline{\mathbf{x}^1\mathbf{x}^4} \cap \mathcal{G}$, $\mathbf{c}^2 := \overline{\mathbf{x}^2\mathbf{x}^4} \cap \mathcal{G}$, $\mathbf{c}^3 := \overline{\mathbf{x}^3\mathbf{x}^4} \cap \mathcal{G}$, $\mathbf{d}^1 := \overline{\mathbf{x}^1\mathbf{x}^5} \cap \mathcal{G}$, $\mathbf{d}^2 := \overline{\mathbf{x}^2\mathbf{x}^5} \cap \mathcal{G}$, $\mathbf{d}^3 := \overline{\mathbf{x}^3\mathbf{x}^5} \cap \mathcal{G}$. Thus we have to decompose the two parts $\mathcal{H}^-$ and $\mathcal{Q}^+$ into pentatopes with

$$\begin{aligned}\mathcal{H}^- &:= \text{conv}(\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3, \mathbf{d}^1, \mathbf{d}^2, \mathbf{d}^3\}), \\ \mathcal{Q}^+ &:= \text{conv}(\{\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3, \mathbf{d}^1, \mathbf{d}^2, \mathbf{d}^3, \mathbf{x}^4, \mathbf{x}^5\}).\end{aligned}$$

Let us start with the decomposition of $\mathcal{H}^-$. Consider the mapping

$$\Phi : \hat{\mathcal{H}} \rightarrow \mathcal{H}^-, (\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4) \rightarrow \sum_{i=1}^3 \sum_{j=1}^3 \rho_i(\hat{x}_1, \hat{x}_2) \rho_j(\hat{x}_3, \hat{x}_4) \mathbf{q}^{i,j}$$

with $\mathbf{q}^{i,1} = \mathbf{x}^i$, $\mathbf{q}^{i,2} = \mathbf{c}^i$ and $\mathbf{q}^{i,3} = \mathbf{d}^i$ where $\rho_i(\hat{x}_1, \hat{x}_2)$ are the barycentric coordinates of the reference triangle $\hat{K} \subset \mathbb{R}^2$. Following section 4.4.3, we have a triangulation of $\hat{\mathcal{H}}$ into pentatopes $\{\hat{\mathcal{D}}_i\}$. With the same arguments as in case 1 one can show that the (pentatope-) piecewise linear interpolation Φ_h of Φ is an isomorphism between $\hat{\mathcal{H}}$ and $\mathcal{H}^-$ and each image $\Phi_h(\hat{\mathcal{D}}_i)$ is again a pentatope. Therefore we can apply the decomposition of the reference hypertriangle $\hat{\mathcal{H}}$ into pentatopes to get the six pentatopes

$$\begin{aligned}\mathcal{D}_u &= \Phi_h(\hat{\mathcal{D}}_u) = \text{conv}(\{\underline{\mathbf{x}}^1, \underline{\mathbf{x}}^2, \underline{\mathbf{x}}^3, \mathbf{c}^2, \mathbf{d}^3\}), \\ \mathcal{D}_v &= \Phi_h(\hat{\mathcal{D}}_v) = \text{conv}(\{\mathbf{x}^1, \underline{\mathbf{c}}^1, \underline{\mathbf{c}}^2, \underline{\mathbf{c}}^3, \mathbf{d}^3\}), \\ \mathcal{D}_w &= \Phi_h(\hat{\mathcal{D}}_w) = \text{conv}(\{\mathbf{x}^1, \mathbf{c}^2, \underline{\mathbf{d}}^1, \underline{\mathbf{d}}^2, \underline{\mathbf{d}}^3\}), \\ \mathcal{D}_1 &= \Phi_h(\hat{\mathcal{D}}_1) = \text{conv}(\{\underline{\mathbf{x}}^1, \underline{\mathbf{c}}^1, \mathbf{c}^2, \underline{\mathbf{d}}^1, \mathbf{d}^3\}), \\ \mathcal{D}_2 &= \Phi_h(\hat{\mathcal{D}}_2) = \text{conv}(\{\mathbf{x}^1, \underline{\mathbf{x}}^2, \underline{\mathbf{c}}^2, \underline{\mathbf{d}}^2, \mathbf{d}^3\}), \\ \mathcal{D}_3 &= \Phi_h(\hat{\mathcal{D}}_3) = \text{conv}(\{\mathbf{x}^1, \underline{\mathbf{x}}^3, \mathbf{c}^2, \underline{\mathbf{c}}^3, \underline{\mathbf{d}}^3\}).\end{aligned}$$

We now turn over to $\mathcal{Q}^+$. For notational convenience define $\mathbf{c}^4 := \mathbf{x}^4$ and $\mathbf{d}^4 := \mathbf{x}^5$. Thus $\mathcal{Q}^+ = \text{conv}(\{\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3, \mathbf{c}^4, \mathbf{d}^1, \mathbf{d}^2, \mathbf{d}^3, \mathbf{d}^4\})$. Now the structure is similar to the situation for $\mathcal{Q}^-$ in Case 1 and we can apply the same procedure and get a valid decomposition $\bigcup \mathcal{P}_i = \mathcal{Q}^+$ with

$$\begin{aligned}\mathcal{P}_1 &= \Phi_h^{\mathcal{Q}}(\hat{\mathcal{P}}_1) = \text{conv}(\{\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3, \mathbf{x}^4, \mathbf{x}^5\}), \\ \mathcal{P}_2 &= \Phi_h^{\mathcal{Q}}(\hat{\mathcal{P}}_2) = \text{conv}(\{\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3, \mathbf{d}^3, \mathbf{x}^5\}), \\ \mathcal{P}_3 &= \Phi_h^{\mathcal{Q}}(\hat{\mathcal{P}}_3) = \text{conv}(\{\mathbf{c}^1, \mathbf{c}^2, \mathbf{d}^2, \mathbf{d}^3, \mathbf{x}^5\}), \\ \mathcal{P}_4 &= \Phi_h^{\mathcal{Q}}(\hat{\mathcal{P}}_4) = \text{conv}(\{\mathbf{c}^1, \mathbf{d}^1, \mathbf{d}^2, \mathbf{d}^3, \mathbf{x}^5\}),\end{aligned}$$

with $\Phi_h^{\mathcal{Q}}$ the corresponding piecewise linear transformation for the 4-prism.

Decomposition of the space-time interface into tetrahedra for case 2. With similar techniques as done for the four dimensional volume, we can proceed with the triangulation of the interface which is isomorph to a 3-prism resulting in tetrahedra $\mathcal{I}_i$, $i = 1, 2, 3$:

$$\mathcal{I}_1 = \text{conv}(\{\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3, \mathbf{d}^3\}), \quad \mathcal{I}_2 = \text{conv}(\{\mathbf{c}^1, \mathbf{c}^2, \mathbf{d}^2, \mathbf{d}^3\}), \quad \mathcal{I}_3 = \text{conv}(\{\mathbf{c}^1, \mathbf{d}^1, \mathbf{d}^2, \mathbf{d}^3\})$$

4.5 Details of the numerical integration for the Space-Time-DG Nitsche-XFEM method

We briefly address the problem of quadrature on four dimensional simplices (pentatopes) and the computation of the weighting factor (measure ratio) $\nu(\mathbf{s}), \mathbf{s} \in \Gamma_{*,h} \subset \mathbb{R}^4$ needed for the implementation of the Space-Time-DG Nitsche-XFEM method.

4.5.1 Quadrature on 4D simplices (pentatopes)

Quadrature rules of high order for simplices can be found in standard references (see e.g. [Str73]) if the dimension d_s of a simplex is smaller than three. For $d_s = 4$, i.e. the simplex is four-dimensional (a pentatope) this is no longer standard. In the literature only a few integration rules can be found (see eg. [Beh08] and [Str73]). We briefly review lower order quadrature rules on pentatopes and discuss how to achieve higher order rules using tetrahedron rules, Duffy transformation and 1D Gauss-Jacobi integration rules. Integration rules are given for the reference pentatope $\hat{\mathcal{P}}$.

4.5.1.1 First order rule

There holds $\int_{\hat{\mathcal{P}}} 1 d\hat{\mathbf{x}} = 1/24$ and $\int_{\hat{\mathcal{P}}} q(\hat{\mathbf{x}}) d\hat{\mathbf{x}} = 1/120$ for $q(\hat{\mathbf{x}}) \in \{\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4\}$, s.t. the following rule is obviously exact for all polynomials up to degree one:

$$I^1(f) = 1/120 \sum_{i=0}^4 f(\mathbf{e}^i)$$

4.5.1.2 Third order rule

A third order rule, taken from [Beh08], is as follows:

$$I^3(f) = 1/120 \sum_{i=1}^5 f(\mathbf{x}^i) \text{ with } \hat{\lambda}_j(\mathbf{x}^i) = \alpha \text{ for } i \neq j \text{ and } \hat{\lambda}_j(\mathbf{x}^i) = \beta \text{ for } i = j$$

where $\hat{\lambda}_j$ is the barycentric coordinate of vertex j in the reference pentatope and the coefficients are $\alpha = 0.118350341907227374$ and $\beta = 0.526598632371090503$.

4.5.1.3 Higher order rules using the Duffy transformation

A more general approach to derive integration rules for pentatopes is based on the Duffy transformation [Duf82]. Let $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \hat{y}_3) \in \mathbb{R}^3$, and $\hat{\mathbf{x}} = (\hat{\mathbf{y}}, t) \in \mathbb{R}^4$. The problem to compute $\int_{\hat{\mathcal{P}}} f(\hat{\mathbf{x}}) d\hat{\mathbf{x}} = \int_{\hat{\mathcal{P}}} f(\hat{\mathbf{y}}, t) d(\hat{\mathbf{y}}, t)$ can be transformed using the transformation $(\hat{\mathbf{y}}, t) \rightarrow (1/(1-t)\hat{\mathbf{y}}, t) = (\tilde{\mathbf{y}}, t)$ (see also figure 4.5.1 for a sketch):

$$\begin{aligned} \int_{\hat{\mathcal{P}}} f(\hat{\mathbf{y}}, t) d(\hat{\mathbf{y}}, t) &= \int_0^1 \int_0^{1-t} \int_0^{1-t-\hat{y}_1} \int_0^{1-t-\hat{y}_1-\hat{y}_2} f(\hat{\mathbf{y}}, t) d\hat{y}_3 d\hat{y}_2 d\hat{y}_1 dt \\ &[\tilde{\mathbf{y}} = 1/(1-t)\hat{\mathbf{y}}] \\ &= \int_0^1 (1-t)^3 \int_0^1 \int_0^{1-\tilde{y}_1} \int_0^{1-\tilde{y}_1-\tilde{y}_2} f((1-t)\tilde{\mathbf{y}}, t) d\tilde{y}_3 d\tilde{y}_2 d\tilde{y}_1 dt \\ &= \int_0^1 (1-t)^3 \int_{\hat{\mathcal{T}}} \tilde{f}(\tilde{\mathbf{y}}, t) d\tilde{\mathbf{y}} dt = \int_0^1 (1-t)^3 \tilde{g}(t) dt \end{aligned}$$

with $\tilde{f}(\tilde{\mathbf{y}}, t) = f((1-t)\tilde{\mathbf{y}}, t)$ and $\tilde{g}(t) = \int_{\hat{\mathcal{T}}} \tilde{f}(\tilde{\mathbf{y}}, t) d\tilde{\mathbf{y}}$. In this form one can apply a one-dimensional integration rule of the form

$$\int_0^1 (1-t)^3 \tilde{g}(t) dt \approx \sum_{k=0}^N \omega_k \tilde{g}(t_k)$$

where ω_k and t_k are weights and points of the corresponding quadrature rule. In order to approximate $\tilde{g}(t_k)$ at every (time) integration point t_k a standard 3D quadrature rule can be applied. Assume this 3D quadrature rule has order q accuracy. The highest order for the pentatope rule at lowest costs is achieved if a Gauss-Jacobi rule (corresponding

to the weight $(1 - t)^3$ of order q is used for the numerical integration w.r.t. t . The resulting quadrature rule has positive weights, but is not symmetric. In principle also the quadrature rule for the tetrahedron can be derived from lower dimensional quadrature rules applying the idea recursively. This generic procedure generates quadrature rules which have slightly more points than symmetric Gauss rules on the simplex. For that reason, a combination of known symmetric Gauss rules for the tetrahedron and the Gauss-Jacobi rule gives a good compromise between simplicity of the rule and efficiency in terms of integration points.

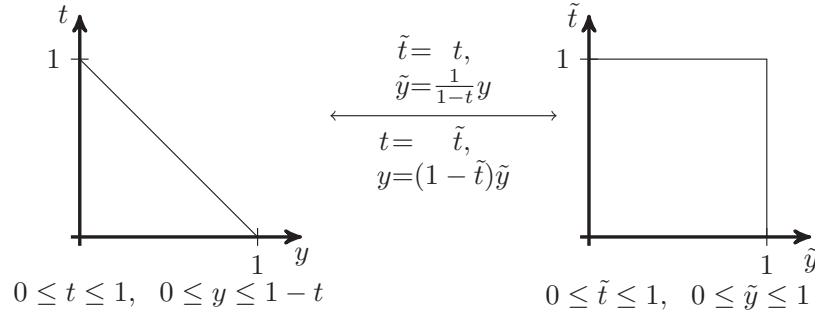


Figure 4.5.1: Sketch of the Duffy transformation for $d = 1$

4.5.2 Computation of ν

We comment on the computation of the weighting factor $\nu(\mathbf{s})$, $\mathbf{s} \in \Gamma_*$. The weighting factor $\nu(\mathbf{s})$ in the Nitsche XFEM-DG method can be computed using the space-time normal $\mathbf{n}^*$ of the space-time interface. One can show that there holds

$$\nu(\mathbf{s}) = (1 + (\mathbf{w} \cdot \mathbf{n})^2)^{-\frac{1}{2}} = \|(n_1, \dots, n_d)^T\|, \quad \mathbf{s} \in \Gamma_*$$

with $\mathbf{n}^* = (n_1, \dots, n_{d+1})^T$ the space-time normal at the interface.

As we use a piecewise planar approximation of the space-time interface consisting of d -simplices in $d + 1$ dimensions we have to compute a normal to the d -simplex. It is known that for $d = 2$ one can use the standard cross-product to compute the normal. In the next section we quote a generalized cross-product which allows to do the same if $d = 3$.

Computing normals to tetrahedra in 4 dimensions

In [Hol91] a generalization of the cross-product is given. Given three vectors $\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3 \in \mathbb{R}^4$ one can compute the cross-product $\mathbf{v} = \mathbf{X}(\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3)$, s.t.

- $\mathbf{X}(\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3) = 0$ iff $\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3$ are linear dependent.
- Iff $\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3$ are linear independent then for $\mathbf{v} = \mathbf{X}(\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3)$, there holds: $\mathbf{v} \perp \mathbf{u}^i$, $i = 1, \dots, 3$.

- $\alpha \mathbf{X}(\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3) = \mathbf{X}(\alpha \mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3) = \mathbf{X}(\mathbf{u}^1, \alpha \mathbf{u}^2, \mathbf{u}^3) = \mathbf{X}(\mathbf{u}^1, \mathbf{u}^2, \alpha \mathbf{u}^3)$, $\alpha \in \mathbb{R}$
- $\mathbf{X}(\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3) = \text{sign}(\pi) \mathbf{X}(\mathbf{u}^{\pi(1)}, \mathbf{u}^{\pi(2)}, \mathbf{u}^{\pi(3)})$, where π is a permutation, i.e. changing the order of the arguments switches the sign.

This cross-product can be used to compute normals to tetrahedra. The computation is given below:

Given $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^4$. Compute $\mathbf{z} = \mathbf{X}(\mathbf{u}, \mathbf{v}, \mathbf{w}) \in \mathbb{R}^4$ as follows:

$$\begin{aligned}
 a_{1,2} &= u_1 \cdot v_2 - u_2 \cdot v_1, & z_1 &= w_2 \cdot a_{3,4} - w_3 \cdot a_{2,4} + w_4 \cdot a_{2,3}, \\
 a_{1,3} &= u_1 \cdot v_3 - u_3 \cdot v_1, & z_2 &= -w_1 \cdot a_{3,4} + w_3 \cdot a_{1,4} - w_4 \cdot a_{1,3}, \\
 a_{1,4} &= u_1 \cdot v_4 - u_4 \cdot v_1, & z_3 &= w_1 \cdot a_{2,4} - w_2 \cdot a_{1,4} + w_4 \cdot a_{1,2}, \\
 a_{2,3} &= u_2 \cdot v_3 - u_3 \cdot v_2, & z_4 &= -w_1 \cdot a_{2,3} + w_2 \cdot a_{1,3} - w_3 \cdot a_{1,2}, \\
 a_{2,4} &= u_2 \cdot v_4 - u_4 \cdot v_2, \\
 a_{3,4} &= u_3 \cdot v_4 - u_4 \cdot v_3,
 \end{aligned}$$

CHAPTER 5

Two-phase flow simulations with mass transport

In the last decades different methods for the solution of flow problems involving two fluid phases have been developed, including the level set [OS88, SSO94, Set99], Volume of Fluid [NW76, HN81], Arbitrary Lagrangian-Eulerian [Beh01, DHPRF04] and diffuse interface methods [AMW98, Jac99]. In this chapter we couple the methods for the mass transport problem presented and analyzed in the previous chapters with a solver for the fluid dynamics.

Outline of this chapter

In section 5.1 we introduce a very common *sharp interface* model for the fluid dynamics of two-phase flows. In recent years, at the Chair for Numerical Mathematics, RWTH Aachen, numerical methods have been developed to solve this model numerically, cf. the internet homepage of the two-phase flow solver DROPS [DRO14]. We discuss the most important features and properties of these methods in section 5.2. The methods are implemented in the software package DROPS, which has also been developed at the Chair for Numerical Mathematics. The discussion is held very brief. For a more extensive overview we refer the interested reader to [GR11]. The methods for the simulation of the fluid dynamics of a two-phase flow have been tested and compared to experiments and other numerical codes between scientific groups participating in the Priority Program SPP 1506 “Transport Processes at Fluidic Interfaces” (cf. the internet homepage [SPP14]). In section 5.3 we present recent results for a benchmark problem which reveal a good agreement of the numerical prediction and experiments with respect to integral and local quantities. We conclude the chapter with results of the simulation of a complex and challenging (one-way) coupled two-phase flow problem involving mass transport in section 5.4.

5.1 Model for fluid dynamics in two-phase flows

The fluid dynamics of a two-phase flow problem with a moving interface can be decomposed into two simpler problems and their coupling. First, assuming the interface motion is known, a Navier-Stokes equation on deforming domains is considered in section 5.1.1. Second, a suitable formulation of the problem where the fluid velocity is assumed to be known and the interface motion has to be determined is described in section 5.1.2. The coupling of both models results in a suitable model for the fluid dynamics in a two-phase flow problem which is summarized in section 5.1.3.

5.1.1 Two-phase Navier-Stokes model

We briefly introduce a standard *sharp interface* model which describes the behavior of two-phase flows. Within the (time-dependent) domains Ω_i , $i = 1, 2$ we consider a standard model for a viscous incompressible Newtonian fluid and then add suitable conditions at the interface. We restrict to isothermal conditions and assume that there is no change of phase. As the phases are viscous the velocity is continuous at the interface, such that

$$[[\mathbf{w}]] = 0 \text{ on } \Gamma(t). \quad (5.1.1)$$

The second condition, related to the momentum balance at the interfaces is more complex. We consider a standard model from the literature [Scr60, BKZ92, GR11]. The jump in the normal stress $\sigma \cdot \mathbf{n}$ along the interface $\Gamma(t)$ is proportional to the local mean curvature $\kappa(x, t)$

$$[[\sigma \cdot \mathbf{n}]] = \tau \kappa \mathbf{n}, \quad x \in \Gamma(t), \quad (5.1.2)$$

where the mean curvature is defined as

$$\kappa(x, t) = -\operatorname{div}(\mathbf{n}(x)), \quad x \in \Gamma(t). \quad (5.1.3)$$

The proportionality constant τ is called the surface tension coefficient. In the definition of the curvature the orientation can be different in other literature, in our definition a convex interior of Γ results in a *negative* κ . The stress tensor takes the form of a *Newtonian* fluid:

$$\sigma = 2\mu D(\mathbf{w}) - p\mathbf{I} \quad (5.1.4)$$

with $D(\mathbf{w}) := \frac{1}{2}\nabla\mathbf{w} + \frac{1}{2}(\nabla\mathbf{w})^T$, $\mathbf{I}$ the identity matrix, μ the dynamic viscosity and p the pressure.

Remark 5.1.1 (Variable surface tension coefficient). *Besides on the geometrical configuration (mean curvature), the surface tension force can also depend on local species concentrations at the interface or the local temperature. This can be modeled by changing from a constant τ in (5.1.2) to a function $\tau = \tau(c, c_\Gamma, T)$, where c is the concentration of a certain species dissolved in the fluid phase at the interface, c_Γ the concentration of surfactants (surface active agents) on the interface and T the temperature. In many*

applications the value of $\tau(c, c_\Gamma, T)$ is significantly lower than the corresponding “clean interface” value τ , which can lead to so-called Marangoni instabilities. In this work we do not consider these effects and consider a constant surface tension coefficient τ .

For now, we assume the interface motion is known. This gives the following standard model.

Problem 5.1.1.

Given suitable boundary conditions and initial values for $\mathbf{w}$, find $\mathbf{w}(\mathbf{x}, t)$ and $p(\mathbf{x}, t)$, s.t.

$$\begin{aligned} \rho_i(\partial_t \mathbf{w} + (\mathbf{w} \cdot \nabla) \mathbf{w}) - \operatorname{div}(2\mu_i D(\mathbf{w})) + \nabla p &= \rho_i \mathbf{g} && \text{in } \Omega_i(t), \ i = 1, 2 \\ \operatorname{div}(\mathbf{w}) &= 0 && \text{in } \Omega_i(t), \ i = 1, 2, \\ \llbracket \sigma \cdot \mathbf{n} \rrbracket &= \tau \kappa \mathbf{n}, \quad \llbracket \mathbf{w} \rrbracket = 0, && \text{on } \Gamma(t). \end{aligned}$$

Note that densities and viscosities depend on the position of the interface.

5.1.2 Model for the evolution of the interface

There are two popular techniques to represent the motion of a sharp interface. Most methods for sharp interface models of two-phase flows fall into one of those two classes: interface tracking or interface capturing methods.

In interface tracking methods, points on the interface (grid points or artificial marker points) are transported explicitly with the flow field. This method has the advantage that an explicit description of the interface can be preserved. The major drawback of this method is that the distribution of control points (grid points or marker points) at the interface will typically deviate significantly from a uniform distribution and redistribution gets necessary. If grid points of a mesh are used as control points, this means that an automated remeshing procedure has to take place after several time steps which is typically challenging and computationally expensive. Especially difficult to handle are situations where the topology of the domains changes, for instance collisions of droplets.

Interface capturing methods such as the Volume of Fluid (VoF) and the level set method were developed to circumvent problems with topology changes and frequent remeshing. These methods use an implicit description of the interface, typically in an Eulerian framework. For that an auxiliary indicator field is introduced. The transport of this field is described by a linear hyperbolic PDE. For an overview of methods we refer to [Loc13, Chapter 2]. In DROPS the level set method is used. We discuss the main idea and the most important properties in the following.

The level set equation

We introduce the scalar level set function $\phi = \phi(\mathbf{x}, t)$. This scalar function characterizes the position of the interface and the domains Ω_i in the following way. There holds

$$\phi(\mathbf{x}, t) \begin{cases} < 0, & \mathbf{x} \in \Omega_1(t), \\ > 0, & \mathbf{x} \in \Omega_2(t), \\ = 0, & \mathbf{x} \in \Gamma(t). \end{cases} \quad (5.1.6)$$

Thus, given a sufficiently smooth level set function ϕ , the interface is determined implicitly by the zero-level of that level set function.

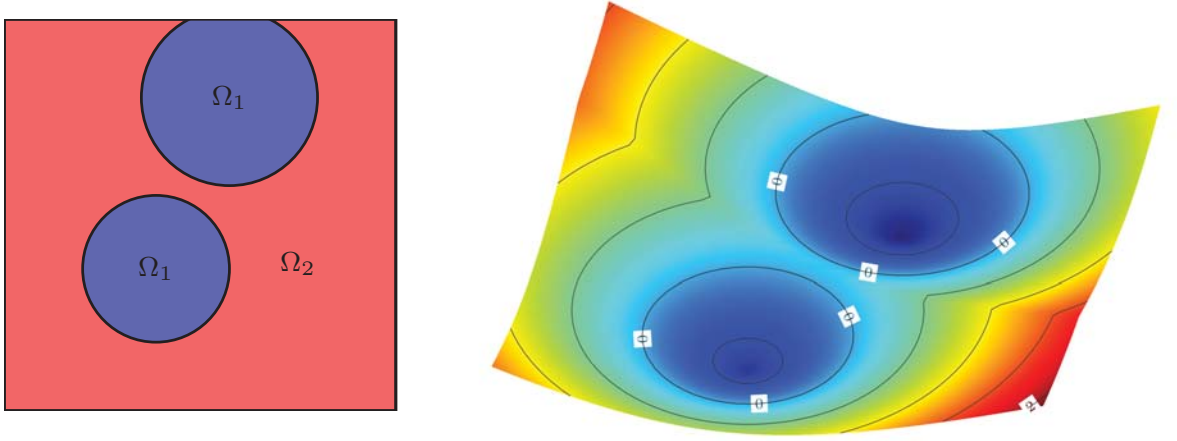


Figure 5.1.1: Sketch of domains Ω_1 and Ω_2 (left) and elevation plot of a corresponding level set function satisfying the signed distance property (right).

Another desirable property of the level set function is the signed-distance property, which means that the scalar value of ϕ at a point $\mathbf{x}$ does not only indicate the domain in which $\mathbf{x}$ lies but further the absolute value of ϕ defines the (approximate) distance of that point to the interface $\Gamma(t)$. ϕ is called a signed distance function to Γ iff $|\phi(\mathbf{x}, t)| = |\text{dist}(\mathbf{x}, \Gamma(t))|$ and (5.1.6) holds. If ϕ only approximately fulfills this property, i.e. $|\phi(\mathbf{x}, t)| \approx |\text{dist}(\mathbf{x}, \Gamma(t))|$ the level set function ϕ is called an approximate signed distance function. In figure 5.1.1 a level set function which fulfills the signed distance property exactly is shown for a simple two-dimensional configuration.

We assume that the interface motion is completely determined by the velocity field $\mathbf{w}$ of the flow problem. In that case the interface motion can be described by the following linear hyperbolic level set problem.

Problem 5.1.2.

Given suitable initial values ϕ_0 and boundary conditions ϕ_D on the inflow part of the boundary $\partial\Omega_-(t) := \{\mathbf{x} \in \partial\Omega, \mathbf{w}(\mathbf{x}, t) \cdot \mathbf{n} < 0\}$, find $\phi(\mathbf{x}, t)$ such that

$$\partial_t \phi + \mathbf{w} \cdot \nabla \phi = 0 \quad \text{in } \Omega, \quad \phi(\mathbf{x}, 0) = \phi_0 \text{ in } \Omega, \quad \phi = \phi_D \text{ on } \partial\Omega_-(t). \quad (5.1.7)$$

Let ϕ be the exact solution of problem 5.1.2. Problem 5.1.2 advects the initial values $\phi(\cdot, 0) = \phi_0$ with the velocity $\mathbf{w}$. Especially the zero-level (the interface Γ) thus follows the flow $\mathbf{w}$. Note, that the signed distance property of an initial level set function ϕ_0 is typically not preserved by the solution of problem 5.1.2.

5.1.3 Two-phase flows model

Note that in problem 5.1.1 it is assumed that $\Gamma(t)$ is known in order to determine the velocity $\mathbf{w}$ (and the pressure). In problem 5.1.2 the opposite is the case, we assume $\mathbf{w}$ is known in order to determine the level set function ϕ and thereby Γ . A problem formulation for the two-phase flow problem with a moving interface can be obtained by a combination of the level set problem, problem 5.1.2, and the two-phase Navier-Stokes equations in problem 5.1.1. Note that the resulting coupling between problem 5.1.2 and problem 5.1.1 is highly non-linear. This combination has also been considered in [CHMO96, SAB+99, PS01, TE00, GR11] and reads as

Problem 5.1.3.

Given suitable initial and boundary conditions for $\mathbf{w}$ and ϕ , find $\mathbf{w}(\mathbf{x}, t), \phi(\mathbf{x}, t), p(\mathbf{x}, t)$ such that

$$\begin{aligned} \rho(\phi)(\partial_t \mathbf{w} + (\mathbf{w} \cdot \nabla) \mathbf{w}) - \operatorname{div}(2\mu(\phi)D(\mathbf{w})) + \nabla p &= \rho(\phi)\mathbf{g} && \text{in } \Omega_i, \ i = 1, 2, \\ \operatorname{div}(\mathbf{w}) &= 0 && \text{in } \Omega_i, \ i = 1, 2, \\ \partial_t \phi + \mathbf{w} \cdot \nabla \phi &= 0 && \text{in } \Omega, \\ \llbracket \sigma \cdot \mathbf{n} \rrbracket &= -\tau \kappa \cdot \mathbf{n}, \quad \llbracket \mathbf{w} \rrbracket = 0, && \text{on } \Gamma(t). \end{aligned}$$

which is complemented by suitable initial and boundary conditions for the velocity $\mathbf{w}$ and the level set function ϕ . Note that here $\rho(\phi)$ and $\mu(\phi)$ are step functions:

$$\rho(\phi) = \begin{cases} \rho_1, & \phi > 0, \\ \rho_2, & \phi \leq 0, \end{cases}, \quad \mu(\phi) = \begin{cases} \mu_1, & \phi > 0, \\ \mu_2, & \phi \leq 0. \end{cases}$$

A weak formulation of problem 5.1.3 is discussed in [GR11, Chapter 6.3].

5.2 Numerical methods for solving two-phase flow problems implemented in DROPS

We briefly summarize the methods used for solving two-phase flow problems in the form of problem 5.1.3. In section 5.2.1 we explain how the interface motion is discretized based on a known velocity field, while in section 5.2.2 we explain the discretization of the two-phase Navier-Stokes equations under the assumption of a known interface motion. The most important features of the methods implemented in software package DROPS are discussed in section 5.2.2. The coupling of the equations in the discretization of problem 5.1.3 is highly non-linear. Usually an iterative decoupling strategy is used to solve for $\mathbf{v}$, p and ϕ . This is discussed in detail in [GR11, Chapter 9.1].

In this section we only present the main ideas. For details we refer to [GRR06, GR07b, GR07a, Gro08] and for an elaborate overview to [GR11].

5.2.1 Discretization of the level set equation

To discretize linear hyperbolic transport equations many established methods exist. In the framework of finite element methods the Streamline Diffusion (SD) or Streamline-Upwind-Petrov-Galerkin (SUPG) method is very natural. We thus consider a discretization of the level set equation problem 5.1.2 using continuous finite elements and a Streamline Diffusion stabilization as in cf. [Loc13], [GR11, Chapter 7.2] and [RST08]. In [Loc13] a comparison between this discretization and a discretization based on a Discontinuous Galerkin method with discontinuous piecewise quadratic finite elements and an upwind formulation is carried out.

We briefly explain the Streamline Diffusion discretization where the level set field ϕ is discretized with the space of continuous piecewise quadratic functions P_2 and consider the semi-discretization in space:

Find $\phi_h \in P_2$, such that

$$(\partial_t \phi_h, v_h)_\Omega + (\mathbf{w} \cdot \nabla \phi_h, v_h)_\Omega + (\partial_t \phi_h + \mathbf{w} \cdot \nabla \phi_h, \mathbf{w} \cdot \nabla v_h)_{0,h} = 0, \quad \forall v_h \in P_2 \quad (5.2.1)$$

where $(\cdot, \cdot)_\Omega$ is the standard $L^2(\Omega)$ scalar product and $(\cdot, \cdot)_{0,h}$ is defined as $(u, v)_{0,h} = \sum_{T \in \mathcal{T}_h} \gamma_T(u, v)_T$ with the stabilization parameter $\gamma_T = \frac{h}{\|\mathbf{w}\|_{\infty, T}}$. This can be combined with a method of lines discretization in time.

For an elaborate discussion of the discretization in space and time we refer to [GR11, Chapter 7.22] and [Loc13].

In practice, besides the space and time discretizations of the level set function there often is a need for further (numerical) corrections. We briefly comment on two important ones

Remark 5.2.1 (Volume correction). *If the velocity field is incompressible then the volume of the domain Ω_1 is preserved exactly for the true solution ϕ of problem 5.1.2. Discretization errors will in general lead to a violation of volume conservation. This is independent of the discretization method as the volume is not a “conserved quantity” in problem 5.1.2. However, this effect decreases for decreasing mesh sizes. To avoid nonphysical shrinkage or growth of the domain Ω_1 a volume correction is usually applied. For instance with a proper (global) shift of the level set function ϕ the volume balance can be corrected, cf. [GR11, Chapter 7.4.2].*

Remark 5.2.2 (Re-Initialization). *Another issue is related to the (approximate) signed distance property. The exact solution ϕ to problem 5.1.2 does not preserve the (approximate) signed distance property which can lead to large and small gradients close to the interface. This again influences the stability and accuracy of the zero-level. It is thus often necessary to restore the (signed) distance property approximately whenever gradients get too large or too small. To restore the (approximate) signed distance property different methods exist. We do not discuss this topic, but refer to [GR11, Chapter 7.4] where the problem of re-initialization is discussed.*

5.2.2 XFEM discretization for the solution of the two-phase Navier-Stokes equations

We briefly present the methods used in DROPS to discretize the two-phase Navier-Stokes problem. Spatial discretizations in DROPS are based on a multilevel hierarchy of nested simplicial grids. The multilevel hierarchy allows for adaptive refinements and coarsenings. On these grids the standard Hood-Taylor P_2 - P_1 stable velocity-pressure pair is used for the discretization of the flow variables, i.e. the velocity is discretized with continuous piecewise quadratics and the pressure with continuous piecewise linear functions.

For the discretization of two-phase problems DROPS has essentially two special components. Due to the interface condition the pressure will in general be discontinuous across the interface and the velocity has kinks across the interface. To account for the jump in the pressure, a discretization using the XFEM for the pressure field is used. To this end the pressure space of continuous piecewise linear functions is enriched with discontinuous functions as presented in section 2.2.1. A discontinuity in the viscosities can further lead to a discontinuity in the normal derivative of the velocity. A stable XFEM enrichment to capture those kinks in the velocity is the topic of ongoing research. In the version of DROPS used in this work no special treatment for potential kinks in the velocity field is used. An appropriate numerical evaluation of the curvature κ for a given level set field ϕ is important for obtaining an accurate resolution of surface tension forces. The discretization of the surface tension force uses a modified Laplace-Beltrami characterization of the mean curvature which has been presented and analyzed in [GR07b], see also [GR11, Chapter 7.6].

In the context of two-phase flows the topic of time discretization for the fluid dynamics and

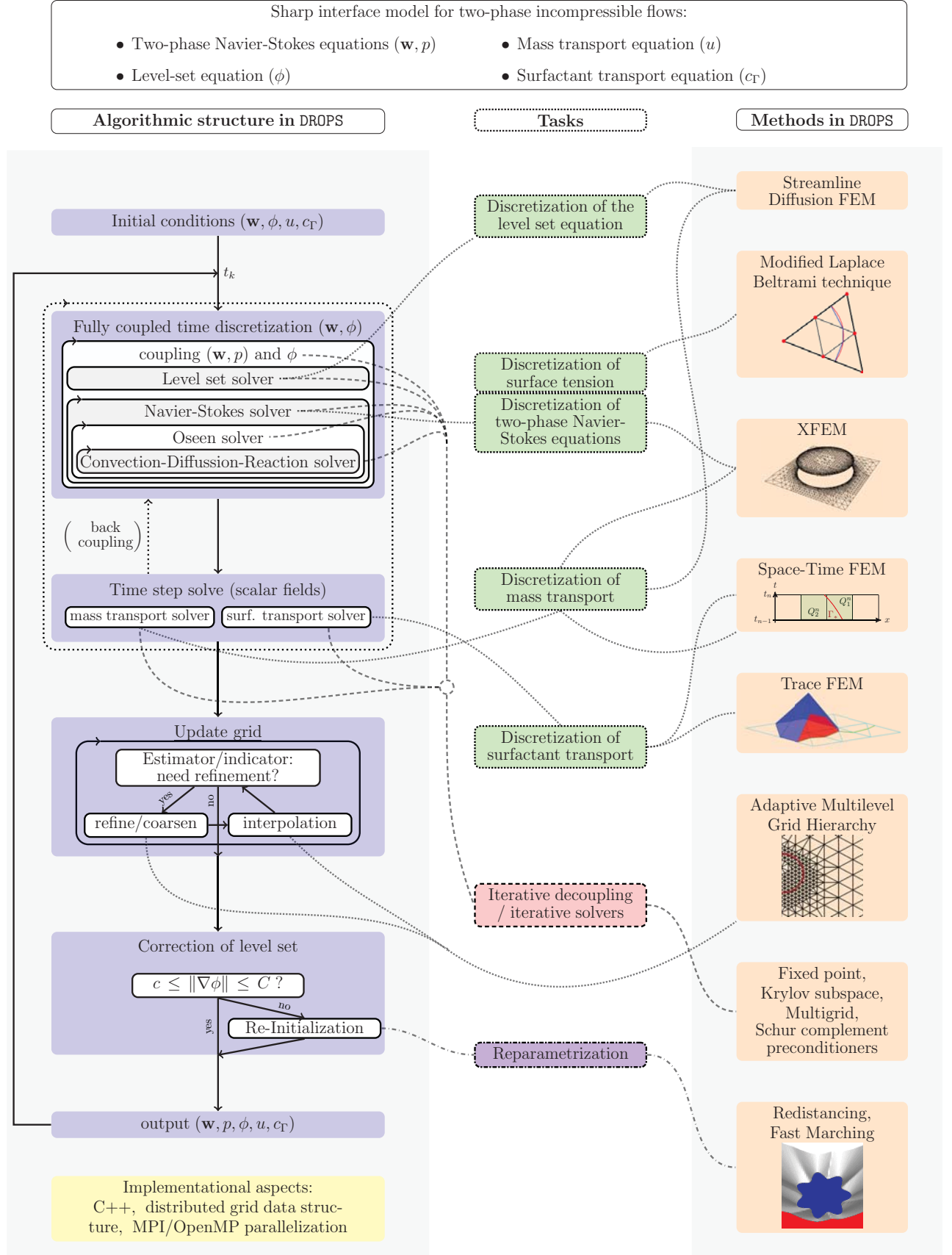


Figure 5.2.1: Algorithmic structure (left) and some important components of DROPS.

the development of robust and efficient iterative solvers for the arising linear systems is very difficult. Solution strategies for these problems have been developed and implemented in DROPS. In figure 5.2.1 some of the most important ingredients that are used in DROPS to simulate two-phase flows (including surfactant and mass transport) are illustrated. The left column in this figure shows the algorithmic structure of the time stepping within DROPS. The coupling of the equations in the discretization of problem 5.1.3 is highly non-linear. Usually an iterative decoupling strategy is used to solve for $\mathbf{w}$, p and ϕ . This is discussed in detail in [GR11, Chapter 9.1]. It is worth noting that the coupling of the different equations leads to nested loops to resolve the non-linearities. The elements in the second column represent the essential tasks, for instance the discretization of the Navier-Stokes equations or the solution of linear systems. In the last column some of the methods used to solve the tasks are listed. The dashed lines in the figure represent the corresponding connections between the algorithmic components, the tasks and the methods. Details on all the aspects illustrated in figure 5.2.1 can be found in [GR11].

5.3 Benchmark problem with complex two-phase fluid dynamics

The comparison of experiments and numerical simulations or the comparison between numerical codes to assess the quality of models, numerical methods and numerical codes for complex interfacial two-phase flows is a very challenging task.

For the purpose of validating and comparing different numerical codes, benchmarks based on simplified problems with artificial parameters and configurations have been considered. Two popular examples are the benchmark problems in [HTK⁺09] and [Zal79]. For DROPS a code-to-code comparison similar to that in [HTK⁺09] has also been considered in [AEG⁺14]. A comparison between DROPS and experimental results based on the rise velocity of rising droplets in liquid-liquid systems has been carried out in [BGG⁺10].

Within the Priority Program SPP 1506 “Transport Processes at Fluidic Interfaces” by the German Research Foundation DFG (cf. the internet homepage [SPP14]) *Taylor bubble flows* have been considered as a suitable and demanding *gas-liquid* two-phase flow system to compare *local quantities* between *experiments* and *numerical simulations* using different models, numerical methods and numerical codes. In [ABH⁺13] and [MBL⁺14] benchmark problems for 2D and 3D Taylor flows have been defined and different numerical codes have been compared to high-resolution experimental data based on X-ray data of the bubble shape. A summary of the results in [ABH⁺13] and [MBL⁺14] is given in [ALM⁺13]. In these papers different numerical approaches, interface capturing (Volume of Fluid, level set) and interface tracking (ALE method), and different models (sharp interface models and diffuse interface models) have been used and compared. The benchmark problems allow for the validation of 2D and 3D codes. We briefly discuss the relevance

and the physics of Taylor flows in section 5.3.1 and describe the concrete physical setting of the case in [MBL⁺14] in section 5.3.2. In section 5.3.3 we comment on the different numerical codes. The discussion of the case setup and the simulation parameters is given in section 5.3.4. The results of the comparison in [MBL⁺14] are summarized in section 5.3.5. These results demonstrate the quality of the numerical methods applied in DROPS for the simulation of the *fluid dynamics* in two-phase flows.

5.3.1 Physics of Taylor flows

In Taylor flows the flow of elongated gas bubbles in capillary channels is considered. The gas bubbles typically exhibit bullet shape (cf. figure 5.3.1). One distinguishes between *Taylor bubbles* which are *single* gaseous bubbles in a narrow channel and *Taylor flow* (also: bubble train flow) where several subsequent bubbles are separated by a liquid slug. Although the bubbles fill the channel almost completely the walls of the channel are not wetted (cf. figure 5.3.1).

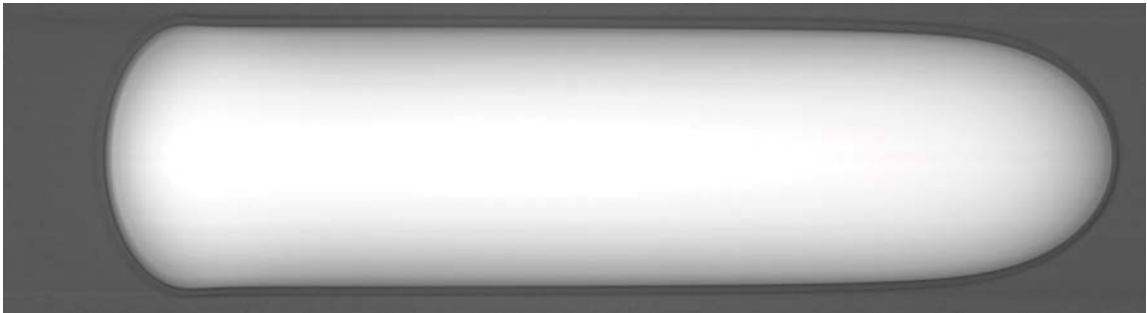


Figure 5.3.1: X-ray projections of a Taylor bubble in a square capillary. The image is taken from [MBL⁺14].

Taylor flows appear in many chemical applications in the field of micro-fluidics, for instance as multiphase monolith microreactors [KKMH05], heat-exchanger reactors [RLGC11] or fuel cells [BS09].

In view of those applications important features of Taylor flows are the high ratio between interface area and volume (the specific exchange area), the small liquid film between gaseous phase and wall and the high mixing rate due to recirculations between bubbles. In [AG08, GFH10] recent reviews on Taylor flow can be found. Viscous and surface tension effects are typically the dominating forces in Taylor flows.

5.3.2 Description of the benchmark problem

In [ABH⁺13] and [MBL⁺14] cases have been considered where surface tension effects are predominant. To characterize the ratio between viscous and surface tension forces the

	liquid phase ($\Omega_2 = \Omega_L$)	disperse phase ($\Omega_1 = \Omega_B$)
density ρ [kg/m^3]	1195.6	1.3
dynamic viscosity μ [$Pa\cdot s$]	28.54×10^{-3}	20×10^{-6}
surface tension τ [N/m]	66.69×10^{-3}	
bubble volume V_B [m^3]	17.5×10^{-9}	
channel width D [m]	1.979×10^{-3}	
bubble velocity U_B [m/s]	0.20557	
gravity g [m/s^2]	9.81	

 Table 5.3.1: Parameters for Taylor Bubble setting in [MBL⁺14]

dimensionless Capillary number is used:

$$Ca = \mu_L U_B / \tau.$$

Here $\mu_L = \mu_2$ is the dynamic viscosity of the bulk fluid, U_B is the (average) bubble velocity and τ the surface tension coefficient. The setting in [MBL⁺14] considers an upward-rising *pressure-driven* flow, where gravity effects do not play an important role. As the substance system is water-glycerol and air the ratios in the densities and viscosities are in the order of 1000. The applied pressure difference is adjusted such that the resulting flow is *laminar*, which means that viscous forces play an important role. To quantify this, the ratio between inertia and viscous forces is introduced, which is described by the Reynolds number

$$Re = \rho_L U_B D / \mu_L,$$

where $\rho_L = \rho_2$ is the density of the bulk fluid and D is the hydraulic diameter of the channel. The fact that the flow is laminar results in a *quasi-stationary* rise configuration which significantly facilitates the comparison between simulations and experiments.

The substance system in [MBL⁺14] consists of 76.9% (volume) glycerol (Sigma Aldrich 49770) and 23.1% (volume) deionized water for the liquid phase at a temperature of 27.9°C and air as gaseous phase. This results in the material parameters summarized in table 5.3.1 and the dimensionless numbers $Ca \approx 0.088$ and $Re \approx 17$.

5.3.3 Methods compared in the benchmark problem

In [MBL⁺14] a comparison between four numerical codes *and* experimental data has been carried out. We give a short overview on the experiments and numerical codes that have been used to obtain the results.

Experiments. The experiments have been carried out in the Institute of Fluid Dynamics, Helmholtz-Zentrum Dresden-Rossendorf. In the experiments Taylor bubbles

are repeatedly injected into a square borosilicate glass capillary of length $90D \approx 18\text{cm}$. The experimental setup allows for a continuous operation where one Taylor bubble after another is rising. The distance between the bubbles, however, is sufficiently large, such that there are no interactions between the bubbles. The visualization experiments have been carried out at the synchrotron radiation source ANKA (Karlsruhe Institute of Technology). Using a high speed camera with 36000 frames per second radiographic images were taken. The images were taken after a distance of $70D \approx 14\text{cm}$ which ensured that the quasi-stationary rise configuration of the Taylor flow had developed. The camera was slowly rotated to obtain images from different angles. The three-dimensional bubble shape has then been reconstructed by superposition of (many) images of different angles and *different bubbles*. Details on the experimental setup and the data processing can be found in [BdSRBH14].

FS3D. The in-house code FS3D is a two-phase flow solver which applies a Volume of Fluid method for interface capturing and is being developed at the Institute of Aerospace Thermodynamics (ITLR), University Stuttgart and the Center of Smart Interfaces, TU Darmstadt, see also the FS3D homepage [FS314]. The solver is based on a Finite Volume discretization on Cartesian staggered grids for the Navier-Stokes equations and the volume fraction which is the indicator field used to describe the interface position in the VoF method. The sharp interface is constructed using a Piecewise Linear Interface Calculation (PLIC) method. Details on the PLIC method can be found for instance in [PP04]. FS3D is parallelized using MPI and OpenMP. For further details we refer to [Rie04] and the corresponding references in [MBL⁺14].

TURBIT-VoF is another in-house code which is developed at the “Institut für Katalyseforschung und -technologie”, at the Karlsruhe Institute of Technology (KIT), see also the TURBIT-VoF homepage [TUR14]. Similar to FS3D it uses a Finite Volume discretization with a Volume of Fluid method for interface capturing. The interface is also computed using the PLIC method with an algorithm different from the one applied in FS3D. The numerical treatment of surface tension forces is also different from the one in FS3D, see the discussion in [MBL⁺14, Section 3.2]. For details on the numerical methods used in TURBIT-VoF we refer the interested reader to [Sab00] and the references in [MBL⁺14].

OpenFOAM/interTrackFoam. OpenFOAM (Open Field Operation And Manipulation) is an open source library for computational continuum mechanics with a variety of solvers, see also the OpenFOAM homepage [Ope14]. The solver used in [MBL⁺14] is the interface tracking method *interTrackFoam* which uses a Finite Volume discretization on a polygonal mesh which is *aligned* to the interface. We will abbreviate the solver as OF/iTF. For the details we refer to [MBL⁺14], the references therein and [TJ12].

The four numerical codes considered for this benchmark problem, FS3D, TURBIT-VoF,

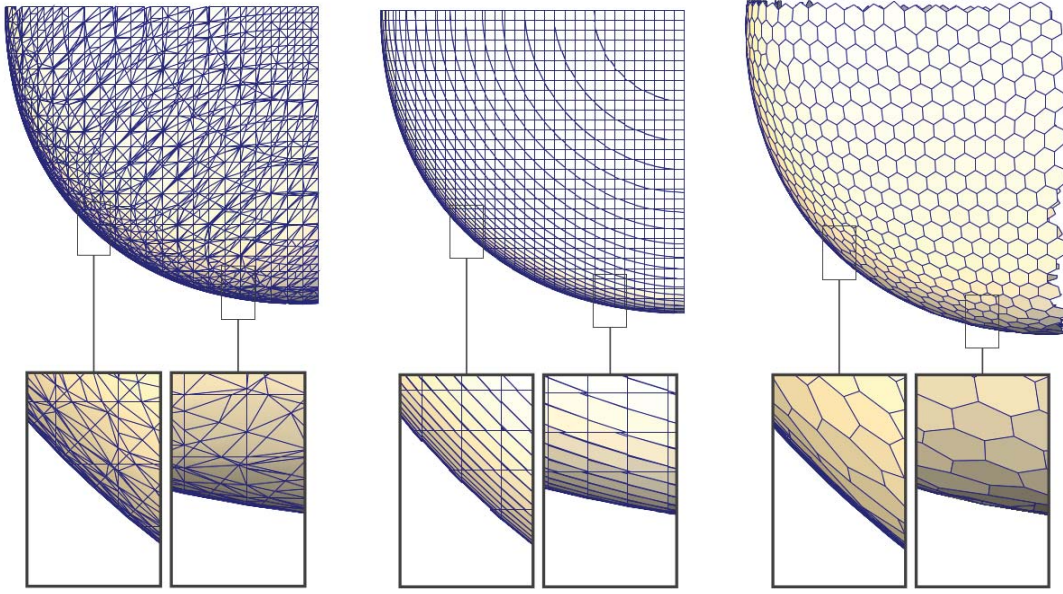


Figure 5.3.2: Rear view on the Taylor bubble in [MBL⁺14] for simulations with DROPS(left), FS3D(center) and OF/iTF(right). The graphic is taken from [MBL⁺14].

OF/iTF and DROPS, are very different in many aspects. However, they are all based on the same sharp interface formulation (cf. problem 5.1.3) of the problem. We briefly highlight important differences between the codes. The discretization of the bulk equations is already different. Here DROPS uses a finite element method whereas all other codes use some version of a Finite Volume discretization. The interface representation is also of different structure. While the OF/iTF discretization has an explicit representation of the interface in terms of cell faces, DROPS, FS3D and TURBIT-VoF use an indicator field to represent the interface implicitly. DROPS uses a level set method for this representation whereas FS3D and TURBIT-VoF use the volume fraction of the VoF method for that. In figure 5.3.2 this difference is depicted by showing the representation of the interface for the methods DROPS, FS3D and OF/iTF. Note that a representation of the interface obtained with TURBIT-VoF would be very similar to that of FS3D.

5.3.4 Case setup in DROPS

For the simulation of the Taylor bubble benchmark problem with DROPS a periodic unit cell of length $6D$ is introduced. With this unit cell configuration the distance between two bubbles is sufficiently large such that we can assume that the interaction between the bubbles is negligible. This assumption has been justified by a comparison with simulation results on a unit cell with length $10D$. As boundary conditions no-slip boundary conditions have been applied at the channel walls. On the top and bottom of the channel periodicity conditions have been applied. In order to render the periodicity physically meaningful the pressure is decomposed into a linearly decreasing part in

the vertical direction z , $-\bar{p}z$, where $\bar{p}$ is a constant, and a periodic part $\tilde{p}$, which is the unknown pressure field in the simulation, such that $p = \tilde{p} - \bar{p}z$. Thus there holds $\nabla p = \nabla \tilde{p} - \nabla(\bar{p}z)$ where the latter contribution is shifted to the r.h.s. of problem 5.1.3 and acts as a volume force $(0, 0, \bar{p})$ in the Navier-Stokes equations. The constant $\bar{p}$ is used to control the bubble rise velocity. We adjust $\bar{p}$ such that the experimentally determined bubble velocity U_B is matched sufficiently accurate in our simulation.

As initial conditions for the interface geometry we used a cylinder combined with two half spheres of the same radius where the radius has been determined such that the bubble volume determined in the experiments is matched. The initial condition for the velocity was obtained by solving a stationary two-phase Stokes problem on this geometry. The unsteady simulation was continued until bubble length, minimal film thickness and the bubble rise velocity were stationary within a certain tolerance.

Simulation parameters

For the simulation in DROPS we used a uniform mesh of size $5 \times 5 \times 30$ on the coarsest level with two additional adaptive refinements towards the interface. The average number of velocity/pressure unknowns was around 321000/26000. For the time discretization we used a version of the backward Euler method with time step size $\Delta t = 8 \times 10^{-5}$. Figure 5.3.3 illustrates the mesh and the magnitude of the relative velocity of the numerical solution obtained with DROPS. It is important to note that in DROPS and TURBIT-VoF the pressure difference (which corresponds to the linear decreasing pressure part $\bar{p}$) is used to match the bubble rise velocity. FS3D and OF/iTF do not use a periodic pressure in the simulation setup. The setup is different such that there the average fluid velocity in the slug (region between two bubbles) has to be prescribed. This is adjusted such that the experimental bubble rise velocity is matched. More details about this can be found in [MBL⁺14, Section 4.1].

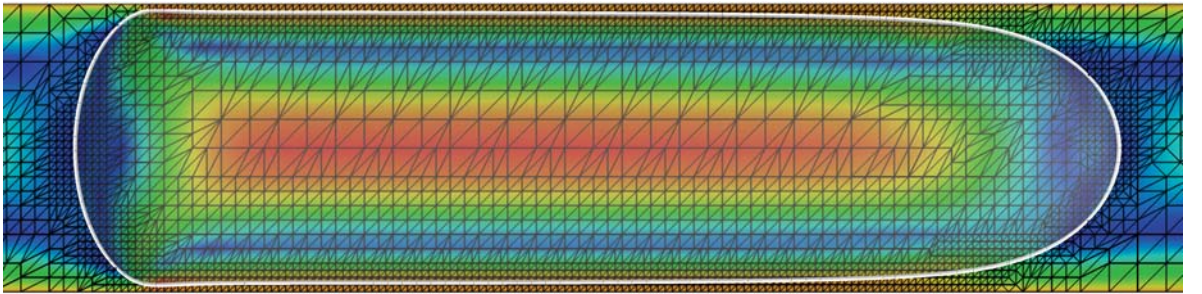


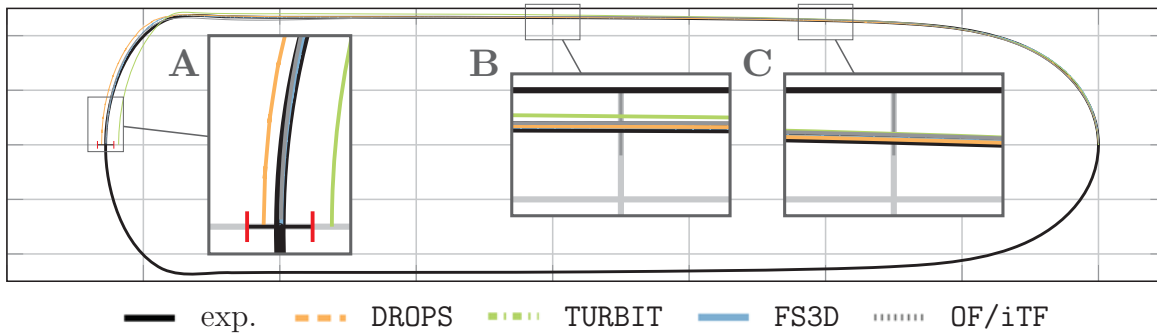
Figure 5.3.3: Simulation results for DROPS. Mesh and interface at a lateral cut. Coloring indicates the relative velocity $\mathbf{v} - (0, 0, 1)^T U_B$. In the visualization each triangle has been divided into four triangles.

	DROPS	exp.	TURBIT	FS3D	OF/iTF
press. difference $\Delta p[Pa]$	267.4	-	275.57	-	-
av. liquid vel. $U_L[m/s]$	-	-	-	0.1382	0.1261
bubble rise vel. $U_B[mm/s]$	206.92	205.57	207.8	197.46	205.77
bubble length $l_B[mm]$	7.23	7.20	7.11	7.197	7.202
vert. min. film thickn. $t_v[mm]$	0.049	0.0505	0.028	0.0477	0.059
diag. min. film thickn. $t_d[mm]$	0.4392	0.4331	0.442	0.4362	0.4392
max. mean curv. $\kappa[m^{-1}]$	4083.4	4055.0	4306.4	4045.0	4020.9

 Table 5.3.2: Simulation results of the benchmark problem in [MBL⁺14]

5.3.5 Simulation results

Several global quantities have been compared such as the obtained bubble rise velocity, the bubble length, the vertical minimal film thickness and the diagonal minimal film thickness and the maximum mean curvature. The numbers are shown in table 5.3.2. The results are in good agreement. Note that the minimal film thickness in the lateral cut $t_v \approx 5 \times 10^{-2} mm$ is smaller than the smallest edge length in DROPS which is $h = 9.895 \times 10^{-2} mm$.


 Figure 5.3.4: Comparison of interface positions between experiments and numerical simulations based on a lateral cut. Graphic taken from [MBL⁺14].

In figure 5.3.4 the interface in a lateral cut is compared between the codes and the experiment. To this end, the bubble tip of all the methods is fixed at the same location. One observes that the shape of the interface is hardly distinguishable except for small deviations at the bubble rear. Those are essentially stemming from different bubble lengths. In [MBL⁺14] further diagrams and a more detailed discussion can be found. We conclude that all codes give reliable results for the considered case. This is particularly interesting for the results obtained with DROPS as the viscosity ratio in this case is very high (≈ 1000) which leads to kinks in the velocities. These kinks are not specifically accounted for in the discretization with DROPS, yet. Even though small spurious oscillations close to the rear of the bubble have been observed, the results are in very good agreement with experiments and the other numerical codes.

Remark 5.3.1 (Comparison of velocity fields). *The presented benchmark problem compared bubble shapes that have been measured in the experiments. Currently, within the SPP 1506 (see [SPP14]) the comparison and validation of velocity profiles between the same numerical codes and experiments is the topic of a research collaboration. Experiments are carried out at the Institute of Multiphase Flows, Hamburg University of Technology. The results indicate that the setup considered within this collaboration is more demanding compared to the one discussed here. Preliminary results, however, look promising. In the publication [MHS14] which is mainly about the experimental setup for this new comparison study some preliminary results obtained with DROPS can be found.*

5.4 Two-phase flow problem with mass transport

In the last section we presented a benchmark problem which demonstrated that the numerical methods used in DROPS are suitable for the simulation of complex two-phase flow problems. The setting in that section did not consider mass transport. In this section we consider the simulation of two-phase flows taking mass transport of a soluble species into account. For the simulation of the mass transport we use the Space-Time-DG Nitsche-XFEM method. To this end we consider the *one-way* coupling of the fluid dynamics problem, problem 5.1.3 and the problem of mass transport, problem 3.1.1. This one-way coupling means that we only consider the influence of the flow field on the mass transport but no back-coupling of the concentration field on the fluid dynamics. Such a back-coupling could be of importance if effects of variable surface tension coefficient are significant, see also remark 5.1.1.

We validated the Space-Time-DG Nitsche-XFEM method in section 3.4 for artificial academic problems with deforming interfaces revealing a high accuracy of the method. In this section we consider a more realistic problem.

In the literature several simplifications are typically made for the validation of methods concerning the simulation of mass transport. One typical simplification is the restriction to a stationary interface, e.g. by considering a rising bubble in (hydrodynamically) stationary state with respect to a frame of reference following the bubble. Such a setting is less interesting for us as after a restriction to stationary interfaces the space-time finite element approach considered in this work would no longer be necessary. The feature of a deforming interface is one of the most challenging aspects of the considered problem class.

In [HT11] challenging scenarios have been considered for two-dimensional direct numerical simulation. In that paper mass transport coefficients are compared to relations known from the engineering literature. Concerning the complexity of the problem an interesting configuration has also been considered in [Bäu14] and [BB13] where rotational symmetry is exploited to apply two-dimensional simulations for a real liquid-liquid system. The numerical results in these publications have been validated with experimental data from

[Weg09]. The parameters in both cases lead to boundary layers which are extremely thin. In [HT11], [Bäu14] and [BB13] those could be resolved using two-dimensional simulations. DROPS is a pure three-dimensional code such that both cases are not feasible for a simulation with DROPS.

In the following we consider a setting as in [KBW03] and [BKW04], where the dissolution of oxygen from a rising (and deforming) $4mm$ air bubble into a water-glycerol solution is investigated.

The structure of this section is as follows. We discuss the physical setting of the problem in section 5.4.1 and the setup of the simulation with DROPS in section 5.4.2. In section 5.4.3 we conclude the section with the discussion of the simulation results.

5.4.1 Physical setting

We explain the setting of the example considered in [KBW03], [BKW04], [Koe04, Chapter 9.8.3] and [One07, Chapter 4.3.2]. A sufficiently large container filled with a water-glycerol mixture is considered and close to the bottom a $4mm$ air bubble with a spherical shape is placed. The concentration of oxygen inside the fluid is assumed to be zero at the initial state and constant (u_0) in the gas. The bubble rises up due to buoyancy forces and during this process the bubble is deforming until it reaches a quasi-stationary state with an ellipsoidal shape. During this evolution oxygen dissolves into the fluid and a wake of oxygen concentration follows the path of the bubble (see figure 5.4.1).

In [KBW03, BKW04, Koe04, One07] the physically correct material parameters concerning the fluid dynamics have been considered. However, for the diffusion in the liquid phase an artificial parameter has been used to prescribe different Schmidt numbers (Sc) in the liquid. The Schmidt number describes the ratio between the kinematic viscosity $\nu = \mu/\rho$ and the diffusion coefficient α in a fluid,

$$Sc = \frac{\mu}{\rho \alpha}.$$

The Schmidt number can also be characterized as the ratio between the Péclet number Pe and the Reynolds number Re , $Sc = Pe/Re$. Typical values for the Schmidt number are $Sc = 1000$ in liquids and $Sc = 1$ in gas.

The mass transport at the interface is essentially determined by the interplay of two processes: convection and diffusion. First, there is the diffusion through the interface which acts in normal direction to the interface. Second, species are transported away from the interface by convection. Note, that the velocity field at the interface is tangentially aligned to the interface. This leads to boundary layers the thickness of which is determined by the ratio between diffusion and convection.

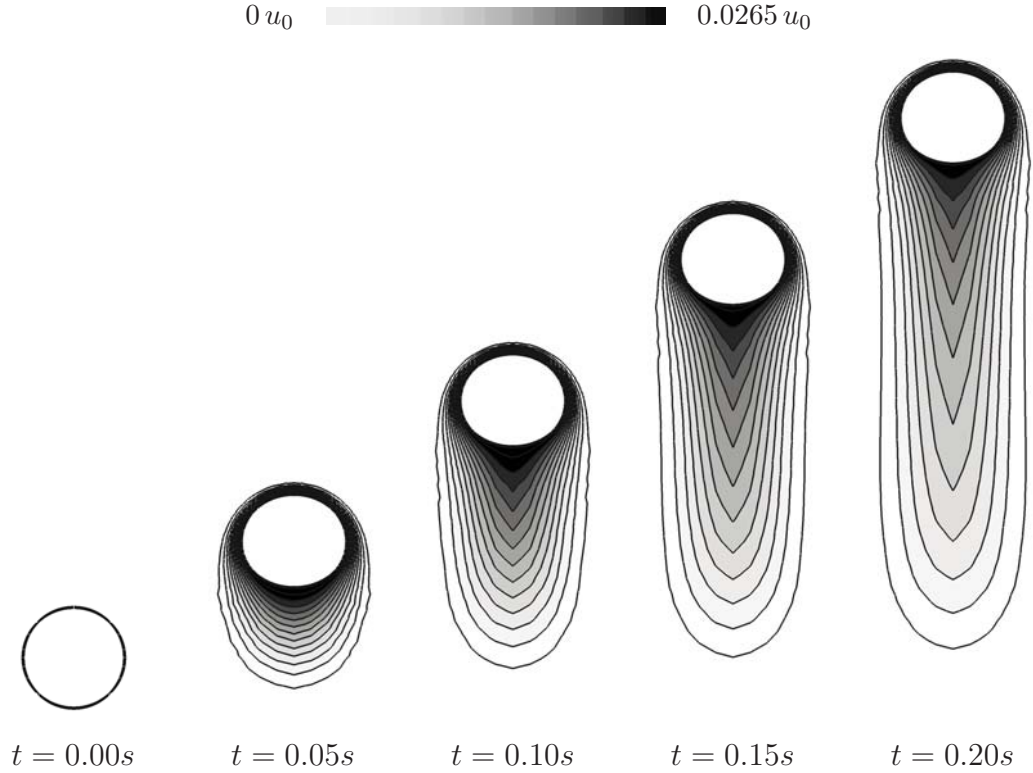


Figure 5.4.1: Concentration contours in the fluid phase at several time for the dissolution process of oxygen from a rising air bubble in a water-glycerol mixture for Schmidt number $Sc = 10$.

5.4.1.1 Fluid dynamics

The substance system under consideration consists of a water-glycerol mixture as the liquid phase and air as the gas phase. The material parameters for this substance system are listed in table 5.4.1. Here, the mixture for water and glycerol consists of 18% (volume) water and 82% (volume) glycerol. The parameters are taken from [KBW03, Koe04] and [RR00]. In [RR00] experimental results for the fluid dynamics are given. Starting from a system at rest the initially spherical bubble rises and reaches a quasi-stationary rise configuration with rise velocity U_B after short time. In this rise configuration the bubble has an approximately ellipsoidal shape. To characterize the shape of the bubble the aspect ratio ω between the shortest diameter (vertical) and the longest diameter (horizontal) is used.

The quasi-stationary rise velocity U_B and the aspect ratio ω have been determined experimentally to $U_B = 0.135m/s$ (cf. [RR00, Fig.9, filled circles]) and $\omega = 0.86$ (cf. [RR00, Fig.11, filled circles]).

The direct numerical simulation of the fluid dynamics for this system has been considered in [KBW03, KBW04, Koe04] and [One07, Chapter 4.3.2] with FS3D and TURBIT-VoF where the rise velocity has been determined to $0.12m/s$ (FS3D) and $0.1136m/s$ (TURBIT-VoF),

	liquid phase ($\Omega_2 = \Omega_L$)	disperse phase ($\Omega_1 = \Omega_B$)
density ρ [kg/m^3]	1205	1.122
dynamic viscosity μ [$Pa\cdot s$]	0.075	1.824×10^{-5}
Henry weight β [1]	1	33
diffusion coeff. α [m^2/s]	$6.224 \times 10^{-5} \cdot Sc^{-1}$	1.916×10^{-5}
surface tension τ [N/m]		0.063
init. bubble diameter d [m]		0.004
gravity g [m/s^2]		9.81

Table 5.4.1: Material parameters for the considered setting. The setting is the same as in [KBW03, BKW04, Koe04].

respectively and the aspect ratio has been determined to 0.86 (FS3D) and 0.87 (TURBIT-VoF). Further, the distance covered by the barycenter of the bubble has been determined as 20.5mm (FS3D) and 21.28mm (TURBIT-VoF).

The dimensionless numbers corresponding to the quasi-stationary rise configuration are $Re \approx 7.6$ and $Ca \approx 0.14$, that means that surface tension plays an important role and that the flow is laminar.

5.4.1.2 Mass transport

We consider the diffusion coefficient α_B of air. The value is taken from the literature. For the diffusion coefficient in the liquid we use an artificial value depending on the Schmidt number $\alpha_L = \mu_L / Sc$. In [KBW03, BKW04, Koe04] different Schmidt numbers have been considered for this setting ($Sc = 1, 2, 5, 10, 20, 50, 100$). Here, we only consider the case $Sc = 10$ where convection dominates but the thickness of boundary layers is not too large. This case is part of the studies in [KBW03, BKW04, Koe04]. Similar studies have also been considered in the recent paper [BF13].

The diffusion coefficient of air is larger than in the liquid for $Sc > 1$. For $Sc = 10$ the diffusion inside the bubble is considerably faster than in the liquid such that the smallest boundary layers are in the liquid phase.

The Henry weights have a ratio of 33, which implies that the concentration at the interface from inside the bubble is 33 times higher than outside. Note that the ratio of 33 in the Henry weight is not “small” in the sense of assumption 1.2.3 (moderate ratios of β). The ratio of diffusivities (for $Sc = 10$) is of the same order (≈ 33) whereas the ratio of scaled diffusivities α/β is of order one.

As initial conditions we consider a constant (non-zero) oxygen concentration u_0 inside the air bubble and no concentration in the liquid. Note that this is not in agreement with the Henry interface condition (1.2.1c) which for $t > 0$ leads to a parabolic boundary layer, cf. section 2.5.5, especially figure 2.5.10, where the same effect has been discussed

for a transient problem with a stationary interface and an artificial flow field.

The ratio between convection and diffusion can be characterized using the dimensionless Péclet number $Pe = d U_B / \alpha$. For the gaseous phase we have the Péclet number $Pe_B \approx 25$ and for the liquid phase we have $Pe_L \approx 7.6 \cdot Sc = 76$. Hence, the problem is convection dominated.

5.4.2 Case setup in DROPS

We explain how the problem described before is simulated in **DROPS**. As wall effects are assumed to be negligible the initial configuration of the physical setting is essentially rotational symmetric. Thus, the setting allows for a two-dimensional simulation exploiting rotational symmetry.

DROPS is a pure three-dimensional code, that means that a two-dimensional (rotational symmetric) setup is not possible with **DROPS**. Nevertheless we can exploit the symmetry to some extent. To this end, we consider the problem on the cylindrical domain with radius $8mm$ and height $40mm$. The initial radius of the air bubble is $2mm$ such that we can assume that the boundary conditions at the wall have negligible effect on the rise behavior and allow for standard no-slip boundary conditions on the whole boundary. Now we consider a 30° -wedge of the cylinder and impose symmetry boundary condition on the rectangular faces of the wedge, see also figure 5.4.2. The corresponding mesh has been generated using the **NETGEN** mesh generator [NET14, Sch97].

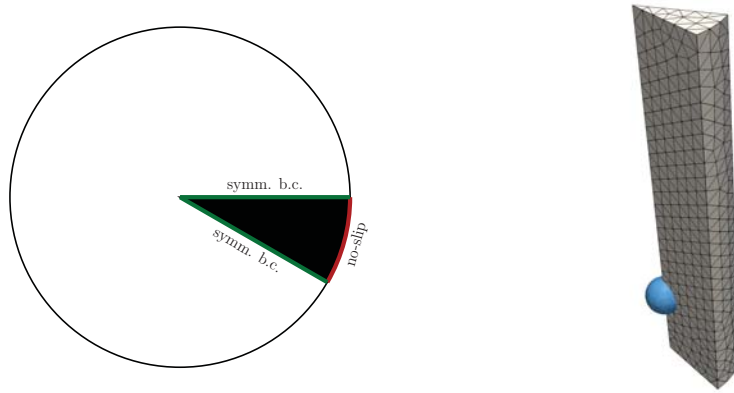


Figure 5.4.2: Exploiting rotational symmetry to reduce computational overhead. Problem on cylindrical domain is reduced to a problem on a 30° -wedge.

The symmetry boundary conditions are

$$\mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega_{\text{symm}} \text{ and } P_\tau(\sigma \cdot \mathbf{n}) = 0 \text{ on } \partial\Omega_{\text{symm}},$$

where $P_\tau = I - \mathbf{nn}^T$ is the tangential projection on $\partial\Omega_{\text{symm}}$. To impose these boundary conditions a slightly adapted version of **DROPS** is used where a Nitsche-type technique has been applied to implement these boundary conditions. Note that this Nitsche technique

is non-standard as it also has to account for surface tension forces stemming from the fact that the interface touches the symmetry boundary $\partial\Omega_{\text{symm}}$.¹

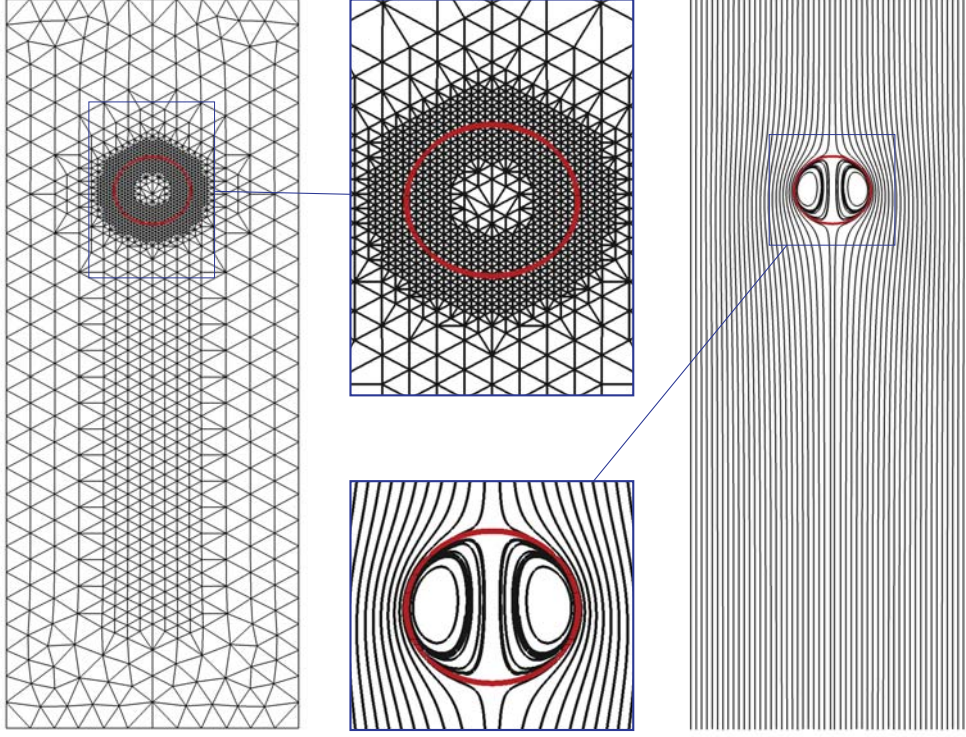


Figure 5.4.3: Final mesh and streamlines of the flow field at time $T = 0.2$.

For the discretization in space we consider the initial grid as shown in figure 5.4.2(right) with three additional levels of adaptive refinements where we refine towards the concentration wake and even more towards the bubble interface (cf. figure 5.4.3). The smallest mesh size is $h \approx 0.15\text{mm}$ which corresponds to a resolution of the bubble diameter with around 25 cells which is comparable to the resolution used in [KBW03, BKW04, Koe04] where the bubble diameter is resolved with 32 cells. The resolution in **DROPS** corresponds to approximately 100000 velocity unknowns and approximately 11000 (space-time) concentration unknowns. Note that the grid is adapting in time depending on the bubble position. Further note, that convection is dominating especially in the liquid phase. Here, the maximum *mesh Péclet number* is $P_h^T = \frac{|\mathbf{v}|_{\infty, T} h}{2\alpha_L} \approx 15.5$. However, no stabilization w.r.t. the dominating convection is applied.

For the discretization in time we consider a fixed step size of $\Delta t = 0.0025$, such that 800 time steps are needed to reach the final time $T = 0.2$. For the fluid dynamics we use a backward Euler discretization. Due to the non-linearity of the Navier-Stokes equations and the coupling with the level set equation the solution of the fluid dynamics problem is by far more expensive than the solution of the mass transport problem with

¹At this point, the author would like to thank Liang Zhang for providing his implementation of these boundary conditions in **DROPS**.

the Space-Time-DG Nitsche-XFEM method. The linear systems for the fluid dynamics are solved using direct solvers. This is due to the fact that the iterative linear solvers in DROPS are not robust for problems with high ratios in the density *and* the viscosity such that direct solvers turned out to be faster in this case.

The Nitsche stabilization in the Space-Time-DG Nitsche-XFEM method is chosen as $\lambda = 20$ and a diagonally preconditioned GMRES method has been applied to solve the arising linear systems.

5.4.3 Simulation results

We present the results obtained with DROPS. The final velocity field obtained with DROPS is displayed in terms of streamlines in figure 5.4.3. For the rise velocity and the distance covered by the barycenter of the bubble we get values of $0.112m/s$ (cf. figure 5.4.4) and $21.44mm$, respectively. Both measures are in very good agreement with the results in [One07] and in acceptable agreement with the results in [KBW03, BKW04, RR00]. The final aspect ratio ω of the bubble shape is 0.88 and thus close to the values in [One07, KBW03, BKW04, RR00]. We conclude that the flow field obtained with DROPS is sufficiently accurate to serve as reliable input for the simulation of mass transport.

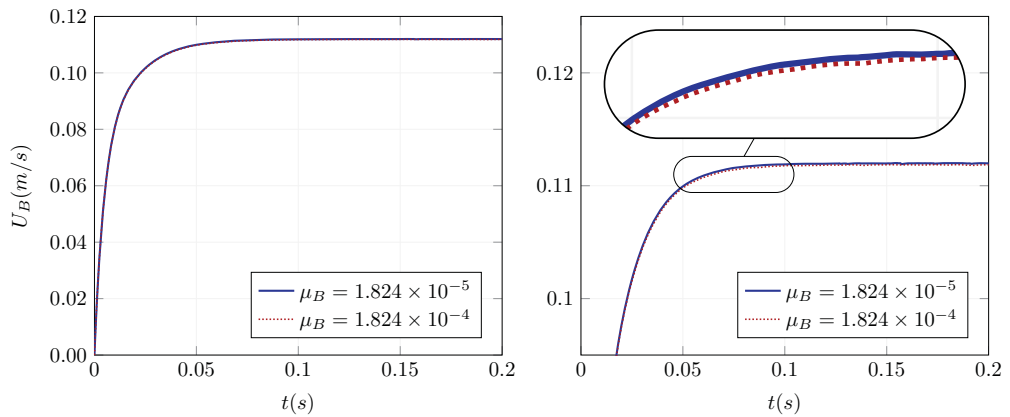


Figure 5.4.4: Bubble rise velocity for physical correct material parameters and increased viscosity for the air bubble, cf. remark 5.4.1.

In figure 5.4.1 the evolution of concentration isolines in the liquid for the simulation with the DROPS setup as describe in section 5.4.2 at different time stages is shown. The corresponding concentration fields inside the bubble at different times and the streamlines corresponding to the velocity (relative to the bubble rise velocity) at $T = 0.2$ are shown in figure 5.4.5.

The simulation captures the physics of the problem very well. We discuss the main effects. The concentration that diffuses into the liquid phase is directed towards the wake of the bubble by the convective flow field. As a consequence boundary layers form and we observe very steep gradients close to the interface. The gradients are steepest at the tip

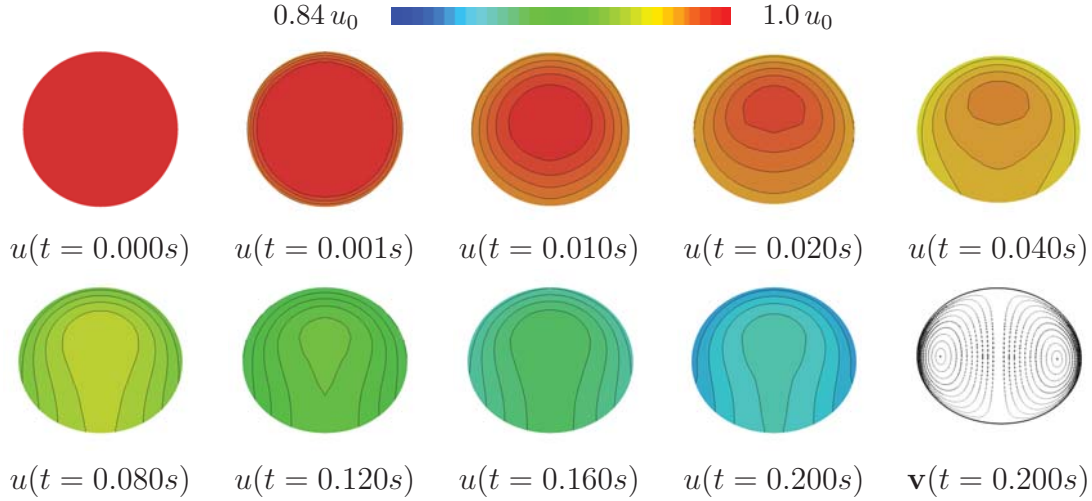


Figure 5.4.5: Concentration distribution inside the gaseous bubble.

of the bubble. This is due to the fact that the tip always sees “fresh” liquid arriving from above while at all other locations at the interface the fluid already traveled some distance along the interface such that diffusion could take place. Accordingly the concentration gradient is lowest at the rear of the bubble. This again leads to the fact that inside the bubble the concentration at the interface is highest at the rear and is pushed towards the tip of the bubble due to the vortex that has formed inside the bubble. Due to the fact that convection dominates in the liquid phase the concentration at the tip is very small. Due to the Henry’s law this also results in a small concentration at the tip inside the bubble. The combination of the fast transfer of concentration away from the tip inside the fluid and the vortex pushing the liquid with high concentration (at the rear of the bubble) towards the tip inside the bubble results in boundary layers inside the bubble. Those can be observed very well in figure 5.4.5. The boundary layers, however, are large compared to the ones outside the bubble due to a larger diffusion inside the gas.

Next, in figure 5.4.6 we consider the concentration along straight lines which are crossing the center of the bubble. We consider the line through the tip of the bubble (0°), a line through the equator (90°) and one line at a 135° angle from the tip. On those lines we plotted the concentration. Due to the Henry interface condition the concentration has jumps across the interface such that the concentration inside the bubble is 33 times larger than outside. We adapted the scaling for the concentration inside and outside the bubble. The scaling is chosen such that a continuous line in the plot corresponds to a concentration field fulfilling the Henry interface condition. We observe in the plot that this condition is fulfilled very accurately. We considered the data given in [Koe04, Figure 9.35] for a comparison. In [Koe04] the initial concentration u_0 is not given explicitly. From the data given in that work we fitted the initial concentration resulting in an assumption of an initial concentration of $u_0 = 25 \times 10^{-6} \frac{\text{mmol}}{\text{l}}$. The results are in good agreement. In the regions where the boundary layer is thin the resolution is probably not high enough for both codes to resolve the boundary layer accurately. The agreement is better for the 135° degree angle where the boundary layer is much thicker and thus

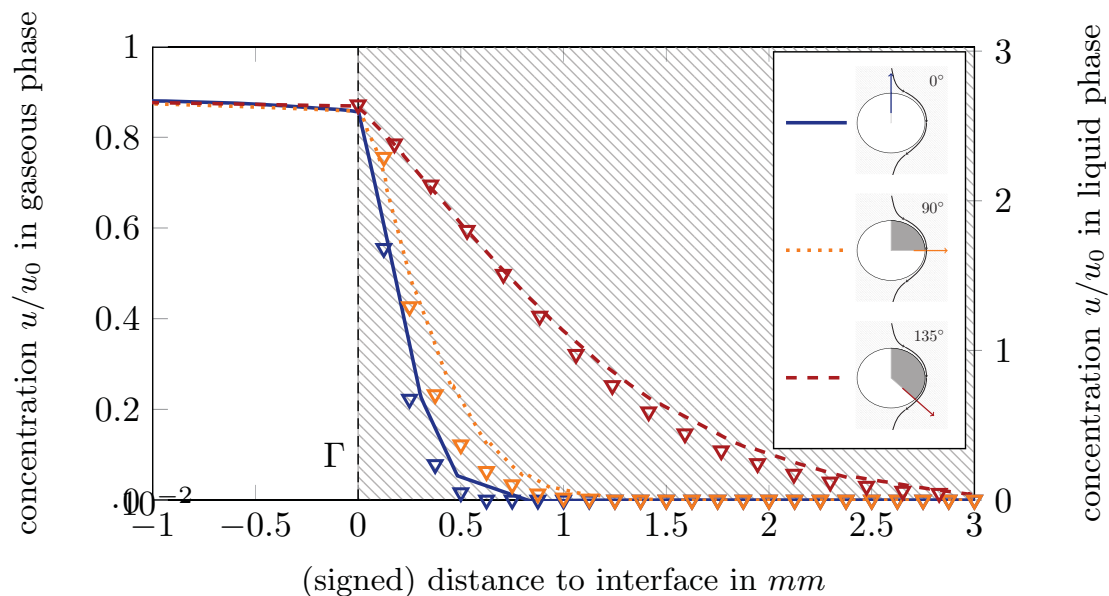


Figure 5.4.6: Concentration along lines for angles 0° , 90° and 135° computed with DROPS (lines) and comparison data from [Koe04] (triangles).

the resolution is higher.

Finally, in figure 5.4.7 the concentration wake obtained with DROPS is compared with the one in [Koe04]. The overall agreement between both simulations is very good. However, the thickness of the wake is slightly different. We assume that this is due to the fact that the resolution in DROPS inside the wake is lower than close to the interface (see figure 5.4.3) and also lower than in FS3D. Further, one observes in figure 5.4.7 that the structure of the contour lines in the liquid phase in a small region behind the rear of the bubble is different for [Koe04] and the results obtained with DROPS. It is not clear where this difference comes from. It might however result from different concentration solutions inside the bubble where we have no reference data for.

We conclude that the obtained results are very reasonable and catch the important features very well. The overall agreement in terms of the flow field and the mass transport with data from the literature is good. However, there is still the need for further investigations to validate the mass transport simulations.

Within the SPP 1506 it is planned to compare results for mass transport problem for Taylor bubbles in counter-current flow between experimental and numerical groups. Finding a suitable setting which allows for reliable measurements and computations at the same time is difficult. It is expected that the collaboration will result in a benchmark problem for mass transport problems which is demanding for direct numerical simulation but at the same time offers sufficiently many data to serve as a foundation for the validation of mass transport simulations. This is the topic of an ongoing research collaboration.

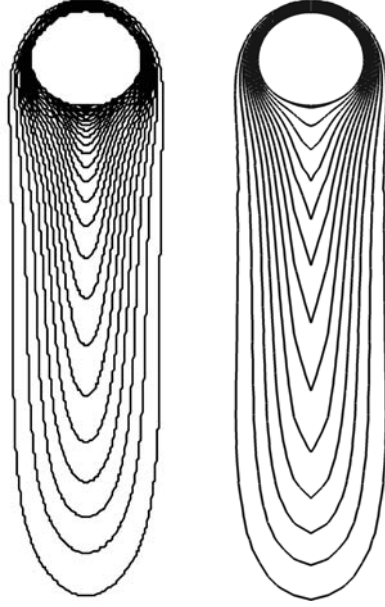


Figure 5.4.7: Comparison between concentration contour lines with FS3D taken from [Koe04] (left) and with DROPS (right).

Remark 5.4.1. *The flow in the considered example is predominantly determined by the densities in both fluids and the viscosity in the liquid. A change in the viscosity of the gas does only have a small effect on the flow behavior. For the solvers (the non-linear solver for the coupling of level set equation and the Navier-Stokes equations and the linear solvers for the Navier-Stokes problems at each linearization) and the discretization in DROPS a change in the gas viscosity can have a significant influence on the performance. An increased gas viscosity implies a smaller viscosity ratio and a smaller Reynolds number inside the bubble which is both beneficial for the performance of DROPS. Even though we used the physically correct parameters in this setting, we also ran the same simulation with a 10 times larger viscosity in the gas phase. This reduced the computation time by approximately one third. In figure 5.4.4 the rise velocity for the case of physically correct and the increased viscosity is shown. We also compared the concentration profiles obtained with both settings and observed that this simplification has only a very small impact on the system behavior.*

CHAPTER 6

Summary and Outlook

We presented a new numerical method for the solution of mass transport problems in two-phase incompressible flows. The considered model for this problem involves moving interfaces which are not fitted by the computational mesh and across which a jump condition (Henry condition) is imposed. The method is a combination of an extended finite element (XFEM) space, the Nitsche method for the imposition of interface conditions and a discrete variational formulation in space-time. We summarize the most important achievements of this thesis and discuss open problems.

6.1 Summary

Stationary interface

The methods presented in this thesis for the stationary interface case are based on similar methods from the literature. Instead of problems with a jump discontinuity, problems where the solutions are continuous but have kinks across the interface are often considered. For the case of a stationary interface we use a method introduced in [RN09, Ngu09] which is based on the Nitsche-XFEM method originally introduced in [HH02]. The results in this work extend the existing methods and their analysis with respect to the following aspects.

Parameter-free Nitsche-XFEM. The Nitsche method is sometimes criticized for the existence of the stabilization parameter λ which has to be chosen “sufficiently large”. We introduced a new parameter-free variant of the Nitsche-XFEM method which guarantees stability independent of a condition on the stabilization parameter, cf. section 2.2.3.2.

Nitsche-XFEM with convection stabilization. For the convection-dominated case a combination of the Nitsche-XFEM method with the Streamline Diffusion method seems natural. We introduced a proper combination of both methods in section 2.2.5 and discussed their interplay. One important finding is the fact that it is necessary to adapt the scaling of the Nitsche penalty term in order to get results which are robust with respect to vanishing diffusion coefficients. We derived a priori error estimates in section 2.3.2 which are robust with respect to the diffusion parameter and applied the methods on numerical test cases.

Optimal preconditioners for Nitsche-XFEM discretizations. The topic of preconditioning of linear systems arising from XFEM discretizations, especially for the Nitsche-XFEM method, is only rarely discussed in the literature. We analyzed properties of the linear systems arising from a Nitsche-XFEM discretization of elliptic interface problems and discovered that the splitting of the enriched finite element space into the standard finite element space and the space of enrichment functions is stable. This inspired the design of a new preconditioner for which we could prove that the condition number is bounded independent of the mesh size h and the position of the interface. Due to the linear costs associated to the application of the preconditioner this is an optimal preconditioner.

Moving interface

For the mass transport problem with a moving interface there are, to the best of our knowledge, no methods which provide an error analysis with second order bounds for the discretization error in space and time. The major contribution of this thesis is the presentation and analysis of the Space-Time-DG Nitsche-XFEM method and the discussion of implementational aspects.

Introduction of the Space-Time-DG Nitsche-XFEM method. In section 3.2 we introduced the Space-Time-DG Nitsche-XFEM method which combines a space-time formulation with an extended finite element (XFEM) space. The space-time domain is divided into time slabs and a Discontinuous Galerkin formulation is used to couple the time slabs. The subdivision into time slabs allows for a computational structure of a time stepping scheme. Within each time slab a space-time finite element space with a tensor-product structure is defined and enriched with the extended finite element method. To enforce the interface condition a space-time version of the Nitsche formulation is applied. This method is new. We analyzed the method and discussed implementational aspects. Both, an implementation in spatially three dimensions as well as the error analysis of such a method has not been done before.

Error analysis of the Space-Time-DG Nitsche-XFEM method. We presented an error analysis for the Space-Time-DG Nitsche-XFEM discretization in section 3.3. The core ingredients of this analysis have also been published in [LR12]. The analysis in this thesis extends the analysis in [LR12] with respect to the considered interpolation operator. In contrast to the interpolation operator considered in [LR12], we considered an L^2 projection on each time slab, cf. section 3.3.2. This interpolation operator requires less regularity of the solution and allows for anisotropic estimates in the analysis of the interpolation operator. While in [LR12] the simplification $h \simeq \Delta t$ has been considered, the analysis in this thesis does not need such a requirements on the ratio between temporal and spatial resolution. Another benefit of taking the L^2 projection as the interpolation operator is that the error analysis allows for changing grids which has not been the case for the error analysis in [LR12].

Numerical integration on four-dimensional cut geometries. An implementation of the Space-Time-DG Nitsche-XFEM method requires the computation of integrals on space-time geometries which are intersected by the interface. This is involved as the space-time interface is only implicitly defined. While solution strategies for cases where the total dimension of the (space-time) domain is less or equal to three are known in the literature, this is no longer true in the spatially three-dimensional case. We presented a robust and second order accurate strategy to construct integration rules on implicitly defined domains (and interfaces) up to spatial dimension three in chapter 4.

Application of the Space-Time-DG Nitsche-XFEM method to realistic problems. The presented Space-Time-DG Nitsche-XFEM method is new. Besides the derivation, error analysis and discussion of implementational aspects (for instance numerical integration in chapter 4 and preconditioning in section 3.5) we applied the method to test problems. In section 3.4 we considered mathematical test problems with reference solutions to investigate the accuracy of the method. In section 5.4 we applied the method to a complex two-phase flow problem with coupled mass transport. The examples demonstrate the accuracy and robustness of the method.

6.2 Open problems and outlook

We outline a few open problems related to the discretizations considered in this thesis for mass transport problems with stationary and moving interfaces that we think are interesting topics for future research.

Stationary interface

Large contrast problems. In this thesis we assumed that the Henry coefficients β and the diffusion parameters α have a small contrast. This assumption is violated in some applications, for instance in many liquid-gas systems. For large contrast problems the discretization method should be modified, especially the averaging operator in the Nitsche stabilization, see the discussion in [BZ12]. This modification in the averaging however may contradict with the stability condition for the Nitsche stabilization. A remedy to this issue is to apply an additional stabilization, for instance the “ghost penalty”, cf. section 2.2.3.5. This stabilization provides stability independent of the averaging operator. This stabilization also guarantees a spectral condition number of the system matrix, $\kappa(\mathbf{A}) \leq ch^{-2}$. However, the presented theory for the preconditioner in section 2.4 is based on a stable subspace splitting. This splitting is no longer robust with respect to h if a “ghost penalty” stabilization is added. According modifications in the stabilization or the preconditioning would need to be found.

Higher order methods. Higher order methods are desirable to get highly accurate results efficiently. In many applications the solution is very smooth in the sub-domains which is in favor of higher order methods. In this work we essentially focused on linear finite elements. The methods (and to a great extent also the analysis) in this work have a natural extension to higher order, see for instance remark 2.3.1. There are, however, open problems which need to be considered carefully before higher order Nitsche-XFEM methods are of practical use.

- The approximations of the domains and the interface used in this thesis are piecewise planar and thereby only of second order accuracy. Although the error of the numerical integration can be reduced by subdivisions, the asymptotic behavior of a method using a piecewise planar approximation is limited to second order. To achieve higher order methods robust methods which allow for higher order numerical integration on implicit domains is indispensable.
- In the analysis of the Streamline Diffusion stabilization a term involving $\operatorname{div}(\varepsilon \nabla u_h)$ arises if a higher order finite element space is used. It is not clear if this term can be controlled for the extended finite element space, cf. remark 2.3.3.

- The preconditioning strategies considered in section 2.4 rely on the fact that the finite element functions are piecewise linear. For a Nitsche-XFEM discretization with higher order finite elements the analysis can not be applied any more. Numerical experiments further indicated that diagonal preconditioning no longer leads to robustness with respect to the interface position if the polynomial degree of the underlying finite element space is greater than one. A higher order version of the “ghost penalty” stabilization, cf. section 2.2.3.5, could be a possible tool to solve this problem.

Moving interface

Error analysis of the Space-Time-DG Nitsche-XFEM method. The error analysis of the space-time method in section 3.3 seems sub-optimal. For simpler parabolic problems without discontinuous solutions (and without XFEM) a third order convergence in time has been proven in the literature, see for instance [Tho97]. The numerical results in section 3.4 indicate the same, a third order of convergence in time. It is not clear if error bounds of third order in time can be obtained.

In the analysis in section 3.3 we assumed that a regularity statement of the form $\|u\|_{2,Q_{1,2}} \leq c\|u_0\|_{1,\Omega}$ holds for a solution of the mass transport problem. This assumption is crucial to obtain the second order bounds. It is however not clear if this assumption can be rigorously justified.

Further, the second order bound derived with the duality arguments is only shown in the weak -1 -norm. To improve the error analysis with respect to the considered norms and the regularity assumptions that are used a better understanding of the regularity of solutions and the relation to the considered anisotropic Sobolev spaces is necessary.

Convection stabilization of the Space-Time-DG Nitsche-XFEM method. In section 3.2.4.3 we briefly proposed a combination of the Space-Time-DG Nitsche-XFEM method with a Streamline Diffusion stabilization in space-time. This method has not been investigated further, yet.

Preconditioning of the Space-Time-DG Nitsche-XFEM method. The efficient preconditioning of the Space-Time-DG Nitsche-XFEM method is important for the practicability of the method. In section 3.5 we observed that a simple diagonal preconditioner already results in robustness w.r.t. the interface position. Further, we motivated and tested a new preconditioner. This new preconditioner uses a decomposition into standard and XFEM unknowns and a two-grid preconditioner in time. The results of this new preconditioner look promising. However, there exists the need for further investigations, cf. remark 3.5.2.

Further investigations for realistic two-phase flow problems with mass transport.

In section 5.3 we presented a validation of the two-phase hydrodynamics in DROPS based on bubble shapes. The evaluation of a similar comparison based on velocities is topic of an ongoing collaboration with the SPP 1506. Further in section 5.4 a validation based on a comparison of mass transport solutions for different numerical codes is carried out. It would be interesting to compare simulation results to measurement data. It is planned to find a suitable benchmark configuration, similar to the one in section 5.3, which involves mass transport and allows for such a comparison.

Bibliography

- [ABCM02] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002. [21](#), [24](#)
- [ABH⁺13] S. Aland, S. Boden, A. Hahn, F. Klingbeil, M. Weismann, and S. Weller. Quantitative comparison of Taylor flow simulations based on sharp-interface and diffuse-interface models. *Int. J. Num. Meth. Fluids*, 73(4):344–361, 2013. [179](#), [180](#)
- [AEG⁺14] J. Adelsberger, P. Esser, M. Griebel, S. Groß, M. Klitz, and A. Rüttgers. 3D incompressible two-phase flow benchmark computations for rising droplets. In *Proceedings of the 11th World Congress on Computational Mechanics (WCCM XI), Barcelona, Spain*, 2014. [179](#)
- [AG08] P. Angeli and A. Gavriilidis. Hydrodynamics of Taylor flow in small channels: A review. In *Proceedings of the Institution of Mechanical Engineers Part C - Journal of Mechanical Engineering Science*, volume 222, pages 737–751. Institution of Mechanical Engineers (Great Britain), 2008. [180](#)
- [ALM⁺13] S. Aland, C. Lehrenfeld, H. Marschall, C. Meyer, and S. Weller. Accuracy of two-phase flow simulations: The Taylor flow benchmark. *Proc. Appl. Math. Mech.*, 13(1):595–598, December 2013. [179](#)
- [AMTX11] Ahmed N., Matthies G., Tobiska L., and Xie H. Discontinuous Galerkin time stepping with local projection stabilization for transient convection-diffusion-reaction problems. *Comput. Meth. Appl. Mech. Eng.*, 200(21–22):17471756, May 2011. [113](#)
- [AMW98] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual Review of Fluid Mechanics*, 30(1):139–165, 1998. [171](#)
- [Bab73a] I. Babuka. The finite element method with penalty. *Math. Comp.*, 27(122):221–228, 1973. [17](#), [26](#)

- [Bab73b] I. Babuska. The finite element method with Lagrangian multipliers. *Numer. Math.*, 20:172–192, 1973. [27](#)
- [Bäu14] K. Bäuml. *Simulation of single drops with variable interfacial tension*. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, Juli 2014. [186](#), [187](#)
- [BB13] K. Bäuml and E. Bänsch. A subspace projection method for the implementation of interface conditions in a single-drop flow problem. *J. Comput. Phys.*, 252:438–457, 2013. [186](#), [187](#)
- [BBH11] R. Becker, E. Burman, and P. Hansbo. A hierarchical NXFEM for fictitious domain simulations. *Int. J. Num. Meth. Eng.*, 86:549–559, 2011. [19](#)
- [BD81] R. E. Bank and T. Dupont. An optimal order process for solving finite element equations. *Math. Comp.*, 36(153):pp. 35–51, 1981. [121](#)
- [BdSRBH14] S. Boden, T. dos Santos Rolo, T. Baumbach, and U. Hampel. Synchrotron radiation microtomography of Taylor bubbles in capillary two-phase flow. *Exp. Fluids*, 55(7), 2014. [182](#)
- [BE86] J. W. Barrett and C. M. Elliott. Finite element approximation of the Dirichlet problem using the boundary penalty method. *Numer. Math.*, 49(4):343–366, 1986. [17](#), [26](#)
- [Beh01] M. Behr. Stabilized space-time finite element formulations for free-surface flows. *Comm. Num. Meth. Eng.*, 11:813–819, 2001. [1](#), [101](#), [171](#)
- [Beh08] M. Behr. Simplex space-time meshes in finite element simulations. *Inter. J. Numer. Meth. Fluids*, 57:1421–1434, 2008. [101](#), [166](#), [167](#)
- [BF91] F. Brezzi and M. Fortin. *Mixed and hybrid finite elements methods*. Springer Series in Computational Mathematics. Springer-Verlag, 1991. [122](#)
- [BF13] D. Bothe and S. Fleckenstein. A Volume-of-Fluid-based method for mass transfer processes at fluid particles. *Chem. Eng. Sc.*, 101:283–302, 2013. [189](#)
- [BGG⁺10] E. Bertakis, S. Groß, J. Grande, O. Fortmeier, A. Reusken, and A. Pfennig. Validated simulation of droplet sedimentation with finite-element and level-set methods. *Chem. Eng. Sc.*, 65:2037–2051, 2010. [179](#)
- [BH10] E. Burman and P. Hansbo. Fictitious domain finite element methods using cut elements: I. a stabilized Lagrange multiplier method. *Comput. Meth. Appl. Mech. Eng.*, 199(41-44):2680–2686, 2010. [23](#), [27](#), [29](#)
- [BH12] E. Burman and P. Hansbo. Fictitious domain finite element methods using cut elements: II. a stabilized Nitsche method. *Appl. Num. Math.*, 62(4):328 – 341, 2012. [23](#), [29](#), [30](#), [56](#), [58](#), [81](#)

- [BKW04] D. Bothe, M. Koebe, and H.-J. Warnecke. VOF-simulations of the rise behavior of single air bubbles with oxygen transfer to the ambient liquid. In F.-P. Schindler, editor, *Transport Phenomena with Moving Boundaries*, pages 134–146. VDI-Verlag, Düsseldorf, 2004. [187](#), [188](#), [189](#), [191](#), [192](#)
- [BKZ92] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, 100:335–354, 1992. [172](#)
- [BMM⁺99] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous finite elements for diffusion problems. *Atti Convegno in onore di F. Brioschi (Milano 1997), Istituto Lombardo, Accademia di Scienze e Lettere*, pages 197–217, 1999. [24](#)
- [BMUP01] T. Belytschko, N. Moes, S. Usui, and C. Parimi. Arbitrary discontinuities in finite elements. *Int. J. Num. Meth. Eng.*, 50:993–1013, 2001. [16](#)
- [BPS01] J. H. Bramble, J. E. Pasciak, and O. Steinbach. On the stability of the L^2 projection in $H^1(\Omega)$. *Math. Comp.*, 71(237):147 – 156, May 2001. [121](#)
- [BR97] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131(2):267–279, March 1997. [24](#)
- [BRM⁺97] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, and M. Savini. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In *Proceedings of 2nd European Conference on Turbomachinery, Fluid Dynamics and Thermodynamics*, pages 99–108. Technologisch Instituut, Antwerpen, Belgium, 1997. [24](#)
- [BS09] C. R. Buie and J. G. Santiago. Two-phase hydrodynamics in a miniature direct methanol fuel cell. *Int. J. Heat Mass Tran.*, 52(21):5158–5166, 2009. [180](#)
- [BS11] E. Burman and G. Smith. Analysis of the space semi-discretized SUPG method for transient convection-diffusion equations. *Math. Models and Meth. Appl. Sciences*, 21:2049–2068, 2011. [55](#)
- [Bur10] E. Burman. Ghost penalty. *Comptes Rendus Mathematique*, 348(21-22):1217 – 1220, 2010. [23](#), [29](#)
- [Bur12] E. Burman. A penalty-free nonsymmetric Nitsche-type method for the weak imposition of boundary conditions. *SIAM J. Numer. Anal.*, 50(4):1959–1981, 2012. [23](#)
- [Bur14] E. Burman. Projection stabilization of Lagrange multipliers for the imposition of constraints on interfaces and boundaries. *Numer. Meth. Part. Diff. Eq.*, 30(2):567–592, 2014. [28](#)
- [BY14] R. E. Bank and H. Yserentant. On the H^1 -stability of the L_2 -projection onto finite element spaces. *Numer. Math.*, 126(2):361–381, 2014. [121](#)

- [BZ12] E. Burman and P. Zunino. Numerical approximation of large contrast problems with the unfitted Nitsche method. In J. Blowey and M. Jensen, editors, *Frontiers in Numerical Analysis - Durham 2010*, volume 85 of *Lecture Notes in Computational Science and Engineering*, pages 227–282. Springer Berlin Heidelberg, 2012. [19](#), [23](#), [29](#), [55](#), [58](#), [70](#), [200](#)
- [CB03] J. Chessa and T. Belytschko. An extended finite element method for two-phase fluids. *ASME J. Appl. Mech.*, 70:10–17, 2003. [16](#)
- [CB04] J. Chessa and T. Belytschko. Arbitrary discontinuities in space-time finite elements by level sets and x-fem. *Inter. J. Numer. Meth. Engng.*, 61:2595–2614, 2004. [106](#)
- [CB06] J. Chessa and T. Belytschko. A local space-time discontinuous finite element method. *Comput. Meth. Appl. Mech. Eng.*, 195(13–16):1325–1343, 2006. [106](#)
- [CH11] A. Chernov and P. Hansbo. *An hp-Nitsches method for interface problems with nonconforming unstructured finite element meshes*, volume 76 of *Lecture Notes in Computational Science and Engineering*, pages 153–162. Springer, 2011. [19](#)
- [CHCOB09] R. Codina, G. Houzeaux, H. Coppola-Owen, and J. Baiges. The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *J. Comp. Phys.*, 228(5):1591–1611, March 2009. [101](#)
- [CHMO96] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *J. Comput. Phys.*, 124:449–464, 1996. [175](#)
- [Clé75] P. Clément. Approximation by finite element functions using local regularization. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 9(R2):77–84, 1975. [120](#), [122](#)
- [CZ98] Z. Chen and J. Zou. Finite element methods and their convergence for elliptic and parabolic interface problems. *Numer. Math.*, 79:175–202, 1998. [39](#)
- [DD76] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic Galerkin methods. In R. Glowinski and J. Lions, editors, *Comput. Meth. Appl. Sc.*, volume 58 of *Lecture Notes in Physics*, pages 207–216. Springer Berlin Heidelberg, 1976. [21](#)
- [DH03] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons, 2003. [2](#), [31](#), [100](#), [113](#)
- [DHPRF04] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran. Arbi-

- trary Lagrangian-Eulerian methods. In *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd, 2004. 1, 100, 171
- [DR82] J. Douglas, Jr. and T. Russell. Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Numer. Anal.*, 19(5):871–885, 1982. 101
- [DRO14] The DROPS package. <http://www.igpm.rwth-aachen.de/DROPS/>, accessed on October, 2014. 171
- [Duf82] M. G. Duffy. Quadrature over a pyramid or cube of integrands with a singularity at a vertex. *SIAM J. Numer. Anal.*, 19(6):1260–1262, December 1982. 167
- [EG04] A. Ern and J.-L. Guermond. *Theory and practice of finite elements*. Springer, New York, 2004. 115, 122
- [ESW05] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers*. Oxford University Press, Oxford, 2005. 31, 32, 86
- [FR14] S. Frei and T. Richter. A locally modified parametric finite element method for interface problems. *SIAM J. Numer. Anal.*, 52(5):2315–2334, 2014. 18, 80
- [FS95] J. Freund and R. Stenberg. On weakly imposed boundary conditions for second order problems. In *Proceedings of the Ninth International Conference on Finite Elements in Fluids*, M. Cecchi et al., eds., pages 327–336. Universita di Padova, 1995. 23
- [FS314] The FS3D package. <http://www.uni-stuttgart.de/itlr/forschung/tropfen/fs3d/>, accessed on October, 2014. 182
- [FZ09] T.-P. Fries and A. Zilian. On time integration in the XFEM. *Int. J. Num. Meth. Eng.*, 79(1):69–93, 2009. 99
- [GFH10] R. Gupta, D. F. Fletcher, and B. S. Haynes. Taylor flow in microchannels: A review of experimental and computational work. *J. Comp. Mult. Flows*, 2(1):1–32, 2010. 180
- [GG95] V. Girault and R. Glowinski. Error analysis of a fictitious domain method applied to a Dirichlet problem. *Japan J. Ind. Appl. Math.*, 12(3):487–514, 1995. 17, 27
- [GPP94a] R. Glowinski, T.-W. Pan, and J. Periaux. A fictitious domain method for Dirichlet problem and applications. *Comput. Meth. Appl. Mech. Eng.*, 111(3–4):283–303, 1994. 17, 27
- [GPP94b] R. Glowinski, T.-W. Pan, and J. Periaux. A fictitious domain method for

- external incompressible viscous flow modeled by Navier-Stokes equations. *Comput. Meth. Appl. Mech. Eng.*, 112(1–4):133–148, 1994. [17](#), [27](#)
- [GR07a] S. Groß and A. Reusken. An extended pressure finite element space for two-phase incompressible flows. *J. Comput. Phys.*, 224:40–58, 2007. [176](#)
- [GR07b] S. Groß and A. Reusken. Finite element discretization error analysis of a surface tension force in two-phase incompressible flows. *SIAM J. Numer. Anal.*, 45:1679–1700, 2007. [176](#), [177](#)
- [GR11] S. Groß and A. Reusken. *Numerical Methods for Two-phase Incompressible Flows*. Springer, Berlin, 2011. [11](#), [12](#), [14](#), [35](#), [44](#), [96](#), [98](#), [105](#), [106](#), [123](#), [149](#), [151](#), [171](#), [172](#), [175](#), [176](#), [177](#), [179](#)
- [Gro08] S. Gross. *Numerical methods for three-dimensional incompressible two-phase flow problems*. PhD thesis, RWTH Aachen, July 2008. [176](#)
- [GRR06] S. Groß, V. Reichelt, and A. Reusken. A finite element based level set method for two-phase incompressible flows. *Comput. Visual. Sci.*, 9:239–257, 2006. [176](#)
- [Hac03] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer-Verlag, Berlin, second edition, 2003. [68](#)
- [Han05] P. Hansbo. Nitsche’s method for interface problems in computational mechanics. *GAMM-Mitt.*, 47(2):183–206, 2005. [19](#)
- [HB79] T. Hughes and R. Brooks. A multidimensional upwind scheme with no crosswind diffusion. In T. Hughes, editor, *Finite Element methods for Convection Dominated flows*, volume 34, pages 19–35. ASME, New York, 1979. [2](#), [32](#)
- [HB82] T. Hughes and R. Brooks. Streamline upwind/Petrov-Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Meth. Appl. Mech. Eng.*, 32:199–259, 1982. [2](#), [32](#)
- [HH02] A. Hansbo and P. Hansbo. An unfitted finite element method, based on Nitsche’s method, for elliptic interface problems. *Comput. Meth. Appl. Mech. Eng.*, 191:5537–5552, 2002. [2](#), [14](#), [19](#), [21](#), [22](#), [39](#), [40](#), [44](#), [58](#), [123](#), [124](#), [197](#)
- [HLZ13] P. Hansbo, M. G. Larson, and S. Zahedi. Characteristic cut finite element methods for convection-diffusion problems on time dependent surfaces. Technical Report 2013-004, Jönköping University, March 2013. [101](#)
- [HLZ14] P. Hansbo, M. G. Larson, and S. Zahedi. A cut finite element method for a Stokes interface problem. *Appl. Num. Math.*, 85:90–114, 2014. [56](#)

- [HN81] C. Hirt and B. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39(1):201 – 225, 1981. [1](#), [171](#)
- [Hol91] S. R. Hollasch. Four-space visualization of 4D objects. Master’s thesis, Arizona State University, 1991. [168](#)
- [HP02] B. Heinrich and K. Pietsch. Nitsche type mortaring for some elliptic problem with corner singularities. *Comput.*, 68:217–238, 2002. [19](#)
- [HR09] J. Haslinger and Y. Renard. A new fictitious domain approach inspired by the extended finite element method. *SJNA*, 47(2):1474–1499, February 2009. [30](#)
- [HT11] K. Hayashi and A. Tomiyama. Interface tracking simulation of mass transfer from a dissolving bubble. *J. Comp. Mult. Flows*, 3(4):247–261, 2011. [186](#), [187](#)
- [HTK⁺09] S. Hysing, S. Turek, D. Kuzmin, N. Parlini, E. Burman, S. Ganesan, and L. Tobiska. Quantitative benchmark computations of two-dimensional bubble dynamics. *Int. J. Numer. Meth. Fluids*, 60:1259–1288, 2009. [179](#)
- [HWGW14] F. Henke, M. Winklmaier, V. Gravemeier, and W. A. Wall. A semi-Lagrangian time-integration approach for extended finite element methods. *Int. J. Num. Meth. Eng.*, 98(3):174–202, 2014. [101](#)
- [Ish75] M. Ishii. *Thermo-Fluid Dynamic Theory of Two-Phase Flow*. Eyrolles, Paris, 1975. [4](#)
- [Jac99] D. Jacqmin. Calculation of two-phase Navier-Stokes flows using phase-field modeling. *J. Comput. Phys.*, 155:96–127, 1999. [171](#)
- [KA10] M. K. Kadalbajoo and P. Arora. Space-time Galerkin least-squares method for the one-dimensional advection-diffusion equation. *Int. J. Comp. Mat.*, 87:103–118, 2010. [113](#)
- [Kar13] M. Karkulik. L^2 -orthogonal projections onto finite elements on locally refined meshes are H^1 -stable. Technical report, Institute for Analysis and Scientific Computing, Vienna University of Technology, July 2013. [121](#)
- [KBW03] M. Koebe, D. Bothe, and H.-J. Warnecke. Direct numerical simulation of air bubbles in water/glycerol mixtures: shapes and velocity fields. In *Proceedings 2003 ASME joint U.S.-European Fluids Eng. Conf.*, Honolulu, 2003. ASME. FEDSM2003-451354. [187](#), [188](#), [189](#), [191](#), [192](#)
- [KKMH05] M. T. Kreutzer, F. Kapteijn, J. A. Moulijn, and J. J. Heiszwolf. Multiphase monolith reactors: chemical reaction engineering of segmented flow in microchannels. *Chem. Eng. Sc.*, 60(22):5895–5916, 2005. [180](#)
- [Koe04] M. Koebe. *Numerische Simulation aufsteigender Blasen mit und ohne*

- Stoffaustausch mittels der Volume of Fluid (VOF) Methode*. PhD thesis, Universität Paderborn, 2004. [187](#), [188](#), [189](#), [191](#), [193](#), [194](#), [195](#)
- [KvdVvdV06] C. Klaij, J. van der Vegt, and H. van der Ven. Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations. *J. Comput. Phys.*, 217(2):589 – 611, 2006. [101](#)
- [Leh15] C. Lehrenfeld. The Nitsche XFEM-DG space-time method and its implementation in three space dimensions. *SIAM J. Sci. Comput.*, 37:245–270, 2015. [106](#), [130](#), [159](#)
- [Li95] Z. Li. A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, 35:230–254, 1995. [76](#)
- [Lio57] J.-L. Lions. Sur les problemes mixtes pour certains systemes paraboliques dans les ouverts non cylindriques. *Annales de l'institut Fourier*, 7:143–182, 1957. [114](#), [115](#)
- [LL94] R. J. Leveque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 31(4):1019–1044, 1994. [17](#)
- [LM72] J. L. Lions and E. Magenes. *Non-Homogeneous Boundary Value Problems and Applications II*. Springer-Verlag, Berlin, 1972. [96](#)
- [LMDM14] C. Lang, D. Makhija, A. Doostan, and K. Maute. A simple and efficient preconditioning scheme for heaviside enriched XFEM. *Comput. Mech.*, pages 1–18, 2014. [56](#)
- [Loc13] E. Loch. *The level set method for capturing interfaces with applications in two-phase flow problems*. PhD thesis, RWTH Aachen, September 2013. [173](#), [176](#)
- [LR12] C. Lehrenfeld and A. Reusken. Nitsche-XFEM with Streamline Diffusion stabilization for a two-phase mass transport problem. *SIAM J. Sci. Comput.*, 34:2740–2759, 2012. [35](#), [47](#), [86](#), [89](#), [199](#)
- [LR13] C. Lehrenfeld and A. Reusken. Analysis of a Nitsche XFEM-DG discretization for a class of two-phase mass transport problems. *SIAM J. Numer. Anal.*, 51:958–983, 2013. [95](#), [105](#), [106](#), [114](#), [124](#), [125](#), [126](#), [127](#), [128](#), [129](#), [130](#), [131](#)
- [LR14] C. Lehrenfeld and A. Reusken. Optimal preconditioners for Nitsche-XFEM discretizations of interface problems. Technical Report 406, Institut für Geometrie und Praktische Mathematik, RWTH Aachen, August 2014. [55](#), [78](#)
- [LSU68] O. Ladyzhenskaya, V. Solonniko, and N. Ural’ceva. Linear and quasilinear equations of parabolic type. *Translations of Mathematical Monographs*, 23, 1968. [96](#)

- [Mas12] R. Massjung. An unfitted discontinuous Galerkin method applied to elliptic interface problems. *SIAM J. Numer. Anal.*, 50(6):3134–3162, 2012. [18](#), [40](#), [75](#)
- [Mau01] B. Maury. A fat boundary method for the poisson problem in a domain with holes. *J. Sc. Comput.*, 16(3):319–339, 2001. [17](#)
- [MBL⁺14] H. Marschall, S. Boden, C. Lehrenfeld, C. J. F. Delgado, U. Hampel, M. Wörner, A. Reusken, and D. Bothe. Validation of interface capturing and tracking techniques with different surface tension treatments against a Taylor bubble benchmark problem. *Comput. & Fluids*, 102:336–352, 2014. [179](#), [180](#), [181](#), [182](#), [183](#), [184](#), [185](#)
- [MBT06] N. Moës, E. Béchet, and M. Tourbier. Imposing Dirichlet boundary conditions in the extended finite element method. *Int. J. Num. Meth. Eng.*, 2006. [18](#)
- [MCCR03] N. Moës, M. Cloirec, P. Cartraud, and J.-F. Remacle. A computational approach to handle complex microstructure geometries. *Comput. Meth. Appl. Mech. Eng.*, 192(28-30):3163–3177, 2003. [18](#)
- [MDB99] N. Moes, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *Int. J. Num. Meth. Eng.*, 46:131–150, 1999. [16](#), [18](#)
- [ME98] P. Mineev and C. R. Ethier. A characteristic/finite element algorithm for the 3-D Navier-Stokes equations using unstructured grids. *Comput. Meth. Appl. Mech. Eng.*, 178(1-2):39 – 50, 1998. [101](#)
- [MHS14] C. Meyer, M. Hoffmann, and M. Schlüter. Micro-PIV analysis of gas-liquid Taylor flow in a vertical oriented square shaped fluidic channel. *Int. J. Multiphase Flow*, 67(0):140 – 148, 2014. [186](#)
- [MKO13] B. Müller, F. Kummer, and M. Oberlack. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *Int. J. Num. Meth. Eng.*, 2013. [150](#)
- [MKOW12] B. Müller, F. Kummer, M. Oberlack, and Y. Wang. Simple multidimensional integration of discontinuous functions with applications to level set methods. *Int. J. Num. Meth. Eng.*, 2012. [149](#)
- [Mül14] B. Müller. *Methods for higher order numerical simulations of complex inviscid fluids with immersed boundaries*. PhD thesis, TU Darmstadt, March 2014. [149](#)
- [NET14] The NETGEN mesh generator. <http://sourceforge.net/projects/netgen-mesher/>, accessed on October, 2014. [190](#)
- [Neu13] M. Neumüller. *Space-Time Methods, Fast Solvers and Applications*. PhD thesis, TU Graz, 2013. [101](#)

- [Ngu09] T. H. Nguyen. *Numerical methods for mass transport equations in two-phase incompressible flows*. PhD thesis, RWTH Aachen, December 2009. 2, 73, 197
- [Nit71] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1):9–15, 1971. 19
- [NW76] W. Noh and P. Woodward. SLIC (Simple Line Interface Calculation). In A. van de Vooren and P. Zandbergen, editors, *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 - July 2, 1976 Twente University, Enschede*, volume 59 of *Lecture Notes in Physics*, pages 330–340. Springer Berlin Heidelberg, 1976. 1, 171
- [One07] A. Onea. *Numerical simulation of mass transfer with and without first order chemical reaction in two-fluid flows*. PhD thesis, Forschungszentrum Karlsruhe, September 2007. 187, 188, 192
- [Ope14] OpenFOAM, The open source CFD toolbox. [http://http://www.openfoam.com/](http://www.openfoam.com/), accessed on October, 2014. 182
- [OS88] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, November 1988. 1, 171
- [PDR07] J. Parvizia, A. Düster, and E. Rank. Finite cell method. *Comput. Mech.*, 41(1):121–133, 2007. 17
- [Pir92] O. Pironneau. Characteristic-Galerkin and Galerkin/least-squares space-time formulations for the advection-diffusion equation with time-dependent domains. *Comp. Meth. Appl. Mech.*, 100:117–141, 1992. 113
- [Pit79] J. Pitkäranta. Boundary subspaces for the finite element method with Lagrange multipliers. *Numer. Math.*, 33:273–289, 1979. 27
- [Pit80] J. Pitkäranta. Local stability conditions for the Babuska method of Lagrangian multipliers. *Numer. Comput.*, 35:1113–1129, 1980. 27
- [Pit81] J. Pitkäranta. The finite element method with Lagrange multipliers for domain with corners. *Numer. Comput.*, 37:13–30, 1981. 27
- [PM89] C. S. Peskin and D. M. McQueen. A three-dimensional computational method for blood flow in the heart I. immersed elastic fibers in a viscous incompressible fluid. *J. Comput. Phys.*, 81(2):372–405, 1989. 17
- [PP04] J. E. Pilliod, Jr. and E. G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J. Comp. Phys.*, 199(2):465–502, September 2004. 182

- [PS01] S. B. Pillapakkam and P. Singh. A level-set method for computing solutions to viscoelastic two-phase flow. *J. Comput. Phys.*, 174:552–578, 2001. [175](#)
- [Reu08] A. Reusken. Analysis of an extended pressure finite element space for two-phase incompressible flows. *Comput. Visual. Sci.*, 11:293–305, 2008. [16](#), [44](#), [56](#), [57](#), [71](#)
- [Rie04] M. Rieber. *Numerische Modellierung der Dynamik freier Grenzflächen in Zweiphasenströmungen*. Fortschritt-Berichte VDI: Reihe 7, Strömungstechnik. VDI-Verlag, 2004. [182](#)
- [RLGC11] M. Roudet, K. Loubiere, C. Gourdon, and M. Cabassud. Hydrodynamic and mass transfer in inertial gas–liquid flow regimes through straight and meandering millimetric square channels. *Chem. Eng. Sc.*, 66(13):2974–2990, 2011. [180](#)
- [RN09] A. Reusken and T. H. Nguyen. Nitsche’s method for a transport problem in two-phase incompressible flows. *J. Fourier Anal. Appl.*, 15:663–683, 2009. [2](#), [11](#), [19](#), [36](#), [47](#), [197](#)
- [RR00] F. Raymond and J.-M. Rosant. A numerical and experimental study of the terminal velocity and shape of bubbles in viscous liquids. *Chem. Eng. Sci.*, 55:943–955, 2000. [188](#), [192](#)
- [RST08] H.-G. Roos, M. Stynes, and L. Tobiska. *Numerical Methods for Singularly Perturbed Differential Equations — Convection-Diffusion and Flow Problems*, volume 24 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2008. [31](#), [32](#), [48](#), [55](#), [176](#)
- [RT02] H. Rui and M. Tabata. A second order characteristic finite element scheme for convection-diffusion problems. *Numer. Math.*, 92(1):161–177, 2002. [101](#)
- [SAB⁺99] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.*, 148:81–124, 1999. [175](#)
- [Sab00] W. Sabisch. *Dreidimensionale numerische Simulation der Dynamik von aufsteigenden Einzelblasen und Blasenschwärmen mit einer Volume-of-Fluid-Methode*. PhD thesis, University Karlsruhe, May 2000. [182](#)
- [SAC97] S. Sadhal, P. Ayyaswamy, and J. Chung. *Transport Phenomena with Droplets and Bubbles*. Springer, New York, 1997. [4](#)
- [SBvdV06] W. E. H. Solle, O. Bokhove, and J. van der Vegt. A space-time discontinuous Galerkin finite element method for two-fluid problems. *J. Comput. Phys.*, 141:46–77, 2006. [101](#)
- [Sch97] J. Schöberl. NETGEN- an advancing front 2D/3D-mesh generator based on abstract rules. *Comput. Visual. Sci.*, 1(1):41–52, 1997. [190](#)

- [Sch12] T. Schulte. Development and implementation of a numerical method for a class of osmotic problems in moving domains. Master’s thesis, RWTH Aachen, August 2012. 112
- [Scr60] L. Scriven. Dynamics of a fluid interface. equations of motion for newtonian surface fluids. *Chem. Eng. Sci.*, 12:98 – 108, 1960. 172
- [Set99] J. A. Sethian. *Level set methods and fast marching methods*. Cambridge University Press, 1999. 1, 171
- [SPP14] Homepage of the priority program (SPP) 1506 “transport processes at fluidic interfaces” of the german science foundation (DFG). <http://www.dfg-spp1506.de>, accessed on October 2014, 2014. 171, 179, 186
- [SSO94] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994. 1, 171
- [SSO07] J. Slattery, L. Sagis, and E.-S. Oh. *Interfacial Transport Phenomena*. Springer, New York, second edition, 2007. 4
- [Ste95] R. Stenberg. On some techniques for approximating boundary conditions in the finite element method. *J. Comp. Appl. Math.*, 63:139–148, 1995. 28
- [Str73] A. H. Stroud. Approximate calculation of multiple integrals. *Math. Comp.*, 27(122):437–440, 1973. 166
- [SvdVvD06] J. Sudirham, J. van der Vegt, and R. van Damme. Space-time discontinuous Galerkin method for advection-diffusion problems on time-dependent domains. *Appl. Num. Math.*, 56(12):1491 – 1518, 2006. 101
- [SZ90] L. R. Scott and S. Zhang. Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Math. Comput.*, 54(190):483 – 493, April 1990. 120
- [Tar07] L. Tartar. *An Introduction to Sobolev Spaces and Interpolation Spaces*. Springer, 2007. 118
- [TBML92] Tezduyar T.E., Behr M., Mittal S., and Liou J. A new strategy for finite element computations involving moving boundaries and interfaces – The deforming-spatial-domain/space-time procedure: II. *Comput. Meth. Appl. Mech. Eng.*, 94(3):353–371, feb 1992. 101
- [TE00] A.-K. Tornberg and B. Engquist. A finite element based level-set method for multiphase flow applications. *Comput. Visual. Sci.*, 3:93–101, 2000. 175

- [Tho97] V. Thomee. *Galerkin finite element methods for parabolic problems*. Springer, Berlin, 1997. 101, 105, 114, 124, 129, 131, 201
- [TJ12] Z. Tukovic and H. Jasak. A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow. *Comput. & Fluids*, 55(0):70 – 84, 2012. 182
- [TLB92] T. Tezduyar, J. Liou, and M. Behr. A new strategy for finite element computations involving moving boundaries and interfaces—the DSD/ST procedure: I. The concept and the preliminary numerical tests. *Comput. Meth. Appl. Mech. Eng.*, 94(3):339–351, 1992. 101
- [TUR14] The TURBIT-VoF package. <http://www.iket.kit.edu/394.php>, accessed on October, 2014. 182
- [Ver91] R. Verfürth. Finite element approximation of incompressible Navier-Stokes equations with slip boundary conditions II. *Numer. Math.*, 59(1):615–636, 1991. 28
- [VvLS08] P. Vos, R. van Loon, and S. Sherwin. A comparison of fictitious domain methods appropriate for spectral/hp element discretisations. *Comput. Meth. Appl. Mech. Eng.*, 197(25):2275–2289, 2008. 18
- [WB14] S. Weller and S. Basting. Efficient preconditioning of variational time discretization methods for parabolic partial differential equations. *ESAIM: Math. Mod. Num. Anal.*, in press, November 2014. 145
- [Weg09] M. Wegener. *Der Einfluss der konzentrationsinduzierten Marangonikonvektion auf den instationären Impuls- und Stofftransport an Einzeltropfen*. PhD thesis, TU Berlin, September 2009. 187
- [Wel13] S. Weller. Variational time discretization for free surface flows. Technical Report 374, Angewandte Mathematik, Friedrich-Alexander-Universität Erlangen-Nürnberg, November 2013. 101
- [WYW06] Z. Wu, J. Yin, and C. Wang. *Elliptic & Parabolic Equations*. World Scientific Publishing, Singapore, 2006. 114
- [Xu92] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34:581–613, 1992. 56, 60, 68
- [Yse93] H. Yserentant. Old and new convergence proofs of multigrid methods. *Acta Numerica*, pages 285–326, 1993. 56, 60, 67, 68
- [Zal79] S. T. Zalesak. Fully multi-dimensional flux corrected transport algorithm for fluid flow. *J. Comput. Phys.*, 31:335–362, 1979. 179

- [Zun13] P. Zunino. Analysis of backward Euler/extended finite element discretization of parabolic problems with moving interfaces. *Comput. Methods Appl. Mech. Engrg.*, 258:152–165, february 2013. [99](#)
- [ZWKB13] S. Zahedi, E. Wadbro, G. Kreiss, and M. Berggren. A uniformly well-conditioned, unfitted Nitsche method for interface problems: Part I. *BIT Numerical Mathematics*, 53:791–820, September 2013. [56](#), [70](#)

List of Figures

1.2.1	Sketch of two phases	3
2.2.1	Combined fictitious domain approach for two domains	15
2.2.2	XFEM shape function	17
2.4.1	Sketch of partitioning of interface elements into patches	61
2.4.2	Local representation of Γ as a graph.	66
2.5.1	Sketch of the setup of star fish example	77
2.5.2	Sketch for 2D conditioning test case	79
2.5.3	Dependency of conditioning on cut position (unstructured mesh)	79
2.5.4	Structured and unstructured grids of conditioning example	81
2.5.5	Dependency of conditioning on cut position (structured mesh)	82
2.5.6	Setup and solution of sharp layer example	87
2.5.7	Convergence of volume errors for sharp layer example	88
2.5.8	Convergence of interface errors for sharp layer example	88
2.5.9	Sketch of interface position and flow field (left) and mesh (right)	89
2.5.10	Evolution of boundary layer at the interface	90
2.5.11	Solution with and without SD and with diff. and conv. scaling	91
2.5.12	Numerical solutions in the wake of the cylinder	92
2.5.13	Refinement of numerical solution in the wake of the cylinder	92
3.1.1	Sketch of method of lines	99
3.1.2	Sketch of ALE method	99
3.1.3	Sketch of semi-lagrangian method	100
3.1.4	Sketch of space-time method	102
3.2.1	Sketch of the space-time triangulation	103
3.2.2	Sketch of space-time XFEM functions	107
3.2.3	Sketch of the space-time DoFs	110
3.4.1	Sketch of the geometrical setup of the quasi-1D space-time example	131
3.4.2	Convergence plot for quasi-1D space-time example	132
3.4.3	Sketch of XFEM unknowns on different space-time meshes	133
3.4.4	Sketch of different space-time geometry approximations	134
3.4.5	Sketch of geometrical setup of example with moving sphere	135
3.4.6	Convergence plot for space-time example with moving sphere	136

3.4.7	Interface positions for vortex example	137
3.4.8	Numerical solution and convergence table for vortex example	138
3.5.1	Precondition strategy for Space-Time-DG Nitsche-XFEM	138
3.5.2	Alternative precondition strategy for Space-Time-DG Nitsche-XFEM	148
4.1.1	Sketch of subdivision strategy of an intersected prism	152
4.1.2	Adaptive interface approximation for the star fish example	153
4.3.1	Sketch of a cut triangle	156
4.3.2	Sketch of a cut tetrahedron (case 1)	157
4.3.3	Sketch of a cut tetrahedron (case 2)	158
4.4.1	Algorithmic structure of decomposition strategy	159
4.4.2	4D reference geometries	161
4.4.3	Decomposition of a 4-prism	162
4.4.4	Decomposition of a hypertriangle	163
4.5.1	Duffy transformation	168
5.1.1	Level set function with signed distance property	174
5.2.1	Components of DROPS	178
5.3.1	X-ray projections of a Taylor bubble in a square capillary	180
5.3.2	Representation of the Taylor bubble interface for different methods	183
5.3.3	Simulation results for the Taylor bubble problem with DROPS	184
5.3.4	Comparison of interface positions between experiments and simulations	185
5.4.1	Concentration contours of oxygen dissolving from a rising bubble	188
5.4.2	Computational grid exploiting rotational symmetry	190
5.4.3	Final mesh and streamlines of the flow field at time $T = 0.2$	191
5.4.4	Bubble rise velocity plot	192
5.4.5	Concentration distribution inside the bubble	193
5.4.6	Concentration profiles at different angles for rising bubble problem	194
5.4.7	Comparison of concentration contour lines (DROPS/FS3D)	195

List of Tables

2.5.1	Approximation of (weakly) discontinuous functions	74
2.5.2	Convergence table for “circle” example	75
2.5.3	Convergence table for “circle” example ($k = 2$)	76
2.5.4	Convergence table for “circle” example ($k = 3$)	76
2.5.5	Convergence table for “starfish” example	77
2.5.6	Condition depending on cut position	80
2.5.7	Dependency of conditioning on h	83
2.5.8	Dependency of conditioning on λ	83
2.5.9	Dependency of conditioning on β	84
2.5.10	Dependency of conditioning on α	84
2.5.11	Iteration numbers with multigrid-precondition	85
2.5.12	Convergence table for convection-diffusion example with smooth solution	86
2.5.13	Convergence table for convection-diffusion example with smooth solution	87
3.4.1	Convergence table (L^2 errors) for test case in section 3.4.1.1.	132
3.4.2	Convergence table (L^2 errors) for the test case in section 3.4.1.2.	133
3.4.3	Ratio between XFEM and standard unknowns for test case in section 3.4.1.2	134
3.4.4	Convergence table (interface errors) for the test case in section 3.4.2 . .	136
3.5.1	Iteration numbers for Gauss-Seidel preconditioned GMRES method. . .	141
3.5.2	Iteration numbers for Space-Time-DG method without XFEM	142
3.5.3	Iteration numbers for block preconditioned GMRES	143
3.5.4	Iteration numbers for G.-S. preconditioned GMRES of XFEM block . .	144
3.5.5	Iteration numbers of two-grid preconditioned GMRES for standard FEM	146
3.5.6	Maximum numbers of outer and inner iterations for the new preconditioner	147
4.2.1	Overview on integral types	155
5.3.1	Parameters for the Taylor Bubble benchmark problem	181
5.3.2	Simulation results of the Taylor Bubble benchmark problem	185
5.4.1	Material parameters for two-phase flow problem with mass transport . .	189