

TECHNISCHE UNIVERSITÄT MÜNCHEN  
Lehrstuhl für Numerische Mechanik

An XFEM based fixed-grid approach  
to fluid-structure interaction

Axel Gerstenberger

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:

Univ.-Prof. Dr.-Ing. habil. Nikolaus A. Adams

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Wolfgang A. Wall
2. Prof. Peter Hansbo, Ph.D.  
University of Gothenburg / Sweden

Die Dissertation wurde am 14. 4. 2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 21. 6. 2010 angenommen.



## Zusammenfassung

Die vorliegende Arbeit behandelt ein Finite Elemente (FE) basiertes Festgitterverfahren zur Simulation von dreidimensionaler Fluid-Struktur-Interaktion (FSI) unter Berücksichtigung großer Strukturdeformationen. FSI ist ein oberflächengekoppeltes Mehrfeldproblem, bei welchem Struktur- und Fluidgebiete eine gemeinsame Oberfläche teilen. In dem vorgeschlagenen Festgitterverfahren wird die Strömung durch eine Eulersche Betrachtungsweise beschrieben, während die Struktur wie üblich in Lagrangscher Betrachtungsweise formuliert wird. Die Fluid-Struktur-Grenzfläche kann sich dabei unabhängig von dem ortsfesten Fluidnetz bewegen, so dass keine Fluidnetzverformungen auftreten und beliebige Grenzflächenbewegungen möglich sind. Durch die unveränderte Lagrangsche Strukturformulierung liegt der Schwerpunkt der Arbeit auf der methodischen Beschreibung des Festgitterverfahrens für inkompressible und viskose Strömungen.

Die Fluidformulierung ist in zwei Schritte unterteilt: Im ersten Schritt werden fiktiven Fluidgebiete, das heisst Gebiete mit überlappenden Fluid- und Strukturnetzen, von dem physikalischen Fluidbereich entkoppelt. Dafür wird mit Hilfe der eXtended Finite Element Method (XFEM) die Fluiddiskretisierung mit sprungartigen Diskontinuitäten erweitert. Die Verwendung der XFEM erlaubt eine scharfe Trennung zwischen fiktiven und physikalischen Strömungsgebieten und vermeidet unnötige Fluidfreiheitsgrade in dem fiktiven Fluidgebiet. Besondere Aufmerksamkeit erfährt auch die Zeitintegration im Hinblick auf zeitveränderlichen Diskontinuitäten, die durch die Bewegung der Fluid-Struktur-Grenzfläche entstehen. Den zweiten Schritt bildet die Kopplung zwischen dem physikalischen Strömungsgebiet auf dem ortsfesten Netz und der in Lagrangscher Betrachtungsweise formulierten Strukturoberfläche. Für eine drei-dimensionale Kopplung wird eine parameterfreie und auf Variationsprinzipien basierende Formulierung vorgestellt, welche prinzipielle Vorteile gegenüber alternativen Kopplungsverfahren bietet.

Schließlich wird die Integration des Festgitterverfahrens in eine bestehende, parallele Multiphysics-Software beschrieben. Für diesen Zweck wird ein Ansatz vorgeschlagen, welcher ein unabhängiges Oberflächennetz zwischen Fluid und Struktur einführt. Die dadurch entstehende methodische und algorithmische Modularität ermöglicht es, etablierte monolithische und partitioinierte FSI-Lösungsverfahren zu verwenden. Dies führt unter anderem dazu, dass alle in der Software bereits implementierten Strukturmodelle für FSI-Berechnungen zur Verfügung stehen. Exemplarische, dreidimensionale FSI-Simulationen demonstrieren die Genauigkeit sowie die Vielseitigkeit des vorgeschlagenen Ansatzes.

## Abstract

The present work describes a finite element based fixed-grid method for the simulation of three-dimensional fluid-structure interaction (FSI) under consideration of large structure deformation. FSI is a surface coupled problem where fluid and structure have a shared interface. In the proposed formulation, the material motion of the fluid is described using the Eulerian formulation, while the structure deformation is described by the Lagrangian description. The fluid-structure interface can move freely on the spatially fixed fluid mesh such that no fluid mesh deformation occurs and an arbitrary interface movement is possible. The Lagrangian structure formulation is not affected by this FSI approach, hence, the focus of this work is on the description of a fixed-grid approach to incompressible and viscous flows.

The fluid formulation is derived in two steps: In the first step, the fictitious domains, i.e. overlapping regions of fluid and structure meshes, are decoupled from the physical flow. For that purpose, the fluid approximation is enriched with step-like discontinuous functions using the extended finite element method (XFEM). The application of the XFEM allows a sharp separation between fictitious and physical fluid domain and avoids the introduction of unnecessary fluid degrees of freedom in the fictitious domain. For moving interfaces, also the time integration receives special attention. The second step is concerned with the coupling of the physical flow on the fixed grid with the Lagrangian structure surface. For a three-dimensional coupling, a parameter-free and weighted residual based formulation is proposed, which provides deciding advantages over alternative coupling approaches.

Finally, the integration of the fixed-grid approach into an existing parallel multiphysics software is described. A coupling approach is proposed that introduces a separate interface mesh between fluid and structure. The resulting methodical and algorithmic modularity allows to apply established monolithic and partitioned FSI approaches. Subsequently, all advanced structure models that are implemented in the multiphysics software are usable for fixed-grid FSI simulations. Exemplary three-dimensional FSI simulations demonstrate the accuracy and versatility of the presented fixed-grid method.

# Dank

Die vorliegende Arbeit wurde in den Jahren 2005 bis 2010 am Lehrstuhl für numerische Mechanik der Technischen Universität München erarbeitet und verfasst. Finanziert wurde die Arbeit durch die Deutsche Forschungsgemeinschaft im Rahmen der DFG Forschergruppe 493: Fluid-Struktur-Wechselwirkung: Modellierung, Simulation, Optimierung, wofür ich sehr dankbar bin.

Der größte Dank gebührt meinem Doktorvater Prof. Dr.-Ing. W. A. Wall. Seine Art, zu führen, Rat zu geben und gleichzeitig die Freiheit für eigene Fehler zu gewähren, ist verantwortlich für viele mir wertvollen Erfahrungen und für ein großes Stück meines akademischen Selbstbewusstseins. Sein Rückhalt und seine Loyalität waren etwas, auf das ich mich stets verlassen konnte. Ich habe es keine Minute bereut, seinem Lehrstuhl beigetreten zu sein und wünsche ihm für seine aktuellen und zukünftigen Projekte viel Erfolg.

Ebenso zum Gelingen der Arbeit hat das großartige Kollegium am Lehrstuhl beigetragen. Besonders wertvoll empfand ich die zahlreichen, erhellenen Diskussionen zu Grundlagen und Spezialgebieten der Mechanik und der Finite Elemente Methode (FEM). Dafür danke ich besonders B. Bornemann (PhD), Dr.-Ing. M. Frenzel, Dr.-Ing. M. Gee, Dipl.-Math. Th. Klöppel und Dipl.-Ing. L. Wiechert. Weiterhin danke ich Dr.-Ing. T. Rabzuk für seine helfenden Erläuterungen zu der eXtendend FEM sowie zu netzfreien Methoden. Für Einblicke in die Grundlagen der stabilisierten FEM für Strömungen danke ich besonders Dr.-Ing. Ch. Förster, Dipl.-Tech. Math. P. Gamnitzer, Dr.-Ing. V. Gravemeier sowie Dr.-Ing. S. Lenz.

Die Implementierung in unser gemeinsames, paralleles Finite Elemente Programm Baci war kein leichtes Unterfangen und sie war nur mit Unterstützung und in Zusammenarbeit mit vielen Kollegen möglich. Besonders danke ich Dr.-Ing. U. Küttler für seine Hilfe bei der Integration der neuen Fluidformulierung in die bestehenden Fluid-Struktur-Kopplungsalgorithmen. Bei der Implementierung unterstützten mich durch Diskussionen und gemeinsame Programmierstunden weiterhin besonders Dipl.-Tech. Math. G. Bauer, Dr.-Ing. M. Gee, Dipl.-Math. Th. Klöppel, Dipl.-Tech. Math. P. Gamnitzer und Dipl.-Ing. L. Wiechert. Ich danke auch Dipl.-Ing. (ETH) U. M. Mayer für ihre Arbeit an den geometrischen Algorithmen, die eine Basis für vorliegende Arbeit bilden. Dass ich von extrem zeitraubenden Programmierfehlern weitgehend verschont wurde, verdanke ich nicht zuletzt den Kollegen, die durch ihr selbstloses Beseitigen eigener und fremder Fehler unser Baci stetig verbessert haben.

Bei der Korrektur und argumentativen Verbesserung der schriftlichen Arbeit unterstützten mich in uneigennütziger Weise besonders meine Kollegen Dipl.-Ing. L. Wiechert, Dipl.-Ing. F. Henke, Dipl.-Ing. Sh. Shahmiri, Dipl.-

Ing. M. Gitterle, Dr.-Ing. M. Gee, Dr.-Ing. U. Küttler sowie Dipl.-Tech. Math. T. Wiesner. Weiterhin danke ich meinem Gutachter Prof. P. Hansbo (PhD) für sein detailliertes Studium der schriftlichen Arbeit und seine abschließenden Hinweise zu ihrer Verbesserung. Viele, hier nicht erwähnte Kollegen – auch außerhalb des Lehrstuhls – hatten ebenfalls ihren Anteil und ihnen allen sei hiermit gedankt.

Schließlich danke ich ganz herzlich meiner Familie, besonders meinen Eltern, meinen Großeltern und meiner Schwester Anne sowie all meinen Freunden. Sie haben trotz der manchmal großen räumlichen Ferne immer Anteil an mir und meiner Arbeit genommen und mir dadurch das Weitemachen erleichtert. Mein Dank gilt auch Marion Schmidt, welche mich durch inhaltliche, besonders aber durch ihre persönliche Unterstützung und ihren Glauben in meine Arbeit vorangebracht hat.

Die mir vielleicht wichtigste persönliche Erfahrung der letzten Jahre ist wohl, dass menschliche Zeit endlich sein kann.

Juli 2010

Axel Gerstenberger

# Contents

<b>Contents</b>	<b>i</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>Notation</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Principal approaches to FSI simulations . . . . .	2
1.3 Objective . . . . .	5
1.3.1 Requirements . . . . .	5
1.3.2 Proposal for an XFEM and Lagrange multiplier based fixed grid method . . . . .	7
1.4 Outline . . . . .	9
<b>2 Governing equations and finite element formulations</b>	<b>11</b>
2.1 Fluid . . . . .	11
2.1.1 Field equations . . . . .	11
2.1.2 Discretization . . . . .	13
2.1.3 Stabilization . . . . .	16
2.1.4 Linearization and solution . . . . .	18
2.2 Structure . . . . .	19
2.2.1 Field equations . . . . .	19
2.2.2 Discretization . . . . .	20
2.2.3 Linearization . . . . .	21
2.3 Coupled fluid-structure system . . . . .	22
2.3.1 Fluid-structure interface conditions . . . . .	23
2.3.2 Three-field formulation of the FSI system . . . . .	23

<b>3 Introduction to the XFEM and weak constraints</b>	<b>27</b>
3.1 Governing equations . . . . .	27
3.2 XFEM formulation . . . . .	29
3.3 Enforcement of interface constraints . . . . .	34
3.3.1 ‘Classical’ Lagrange multiplier . . . . .	35
3.3.2 Nitsche’s method . . . . .	36
3.3.3 Generalized Hellinger-Reissner principle . . . . .	37
3.3.4 Flux-based Lagrange multiplier . . . . .	39
3.4 Numerical integration . . . . .	41
3.5 Numerical examples for the heat conduction . . . . .	49
3.5.1 2D heat conduction in a rectangular domain . . . . .	50
3.5.2 3D heat conduction in a rectangular domain . . . . .	50
3.5.3 Heat conduction between concentric cylinder surfaces .	52
<b>4 Stationary implicit fluid surfaces</b>	<b>55</b>
4.1 Definition of the fluid problem . . . . .	55
4.2 XFEM formulation . . . . .	57
4.2.1 Discretization . . . . .	57
4.2.2 Enrichment strategies . . . . .	58
4.2.3 Practical considerations . . . . .	61
4.3 Enforcement of interface constraints . . . . .	62
4.3.1 Traction based Lagrange multiplier . . . . .	63
4.3.2 Cauchy stress based Lagrange multiplier . . . . .	66
4.3.3 Viscous stress and pressure based Lagrange multipliers .	73
4.4 Implementation and Algorithms . . . . .	75
4.4.1 Interface localization . . . . .	75
4.4.2 Integration cell generation . . . . .	77
4.4.3 Enrichment and unknown management . . . . .	79
4.4.4 Visualization . . . . .	80
4.5 Numerical examples . . . . .	81
4.5.1 Qualitative examples . . . . .	81
4.5.2 3D patch tests . . . . .	85
4.5.3 3D benchmark computations . . . . .	89
<b>5 Moving implicit fluid surfaces</b>	<b>97</b>
5.1 Introduction . . . . .	97
5.2 A Ghost-Fluid method for the XFEM . . . . .	101
5.2.1 Setup . . . . .	101
5.2.2 Discretization . . . . .	101
5.2.3 A one-dimensional example . . . . .	103
5.2.4 Implementation . . . . .	106
5.3 Treatment of moving walls . . . . .	107

5.3.1	Interface movement . . . . .	107
5.3.2	Coupling of fluid velocity with interface displacement .	108
5.3.3	Example . . . . .	111
<b>6</b>	<b>Coupled fluid-structure system</b>	<b>117</b>
6.1	Interface-structure coupling . . . . .	117
6.2	FSI system with traction Lagrange multipliers . . . . .	120
6.3	FSI system with stress Lagrange multipliers . . . . .	122
6.4	Iterative staggered strong coupling scheme . . . . .	125
6.5	Enhancement of flow solutions near the fluid structure interface	128
6.5.1	Adaptive fixed-grid methods . . . . .	129
6.5.2	A hybrid fixed-grid / ALE approach . . . . .	131
6.6	Numerical examples . . . . .	135
6.6.1	Rotating cylinder . . . . .	135
6.6.2	Resolution of boundary layers . . . . .	135
6.6.3	FSI including a compressible structure . . . . .	139
6.6.4	Channel flow over a bending beam . . . . .	141
6.6.5	Channel flow over a deforming thick plate . . . . .	142
6.6.6	Contact of submerged structures . . . . .	142
<b>7</b>	<b>Summary and conclusion</b>	<b>145</b>
7.1	Summary . . . . .	145
7.2	Outlook . . . . .	147
<b>A</b>	<b>Stress Lagrange multiplier for linear-elastic continua</b>	<b>149</b>
A.1	Problem definition . . . . .	149
A.2	Weakly enforced Dirichlet conditions . . . . .	151
A.2.1	Traction Lagrange multiplier . . . . .	151
A.2.2	Cauchy stress based Lagrange multiplier . . . . .	152
A.3	Relation to the HR variational principle and Nitsche's method .	155
<b>B</b>	<b>Reference solution for the Jeffery-Hamel flow</b>	<b>157</b>
<b>Bibliography</b>		<b>163</b>



# List of Figures

1.1	FSI as a surface coupled two-field problem . . . . .	2
2.1	FSI two-field setup . . . . .	23
2.2	FSI three-field setup . . . . .	24
3.1	From surfaces to internal interfaces . . . . .	28
3.2	Enrichment strategies for the 1D Poisson equation . . . . .	31
3.3	Integration cells . . . . .	43
3.4	Coordinate systems for physical space, elements and integration cells . . . . .	43
3.5	Convergence study of rectangular domain - 2D . . . . .	51
3.6	Convergence study of rectangular domain - 3D . . . . .	52
3.7	Convergence study for cylinder problem . . . . .	53
4.1	Fluid part of the FSI problem . . . . .	56
4.2	Enrichment for bulky and approaching structures . . . . .	59
4.3	Enrichment for thin structures . . . . .	60
4.4	Thick structures with traction Lagrange Multiplier . . . . .	65
4.5	Definition of element boundary for boundary integrals. . . . .	70
4.6	Enrichment for element stresses . . . . .	70
4.7	Parallel distribution of fluid mesh and FSI surface . . . . .	77
4.8	Fundamental steps for generating integration cells and enrichment . . . . .	78
4.9	Stationary flow around bulky structure . . . . .	82
4.10	Flow around thin structure. . . . .	83
4.11	Flow around thin structure - enriched solutions . . . . .	83
4.12	Flow around thin, closed structure: Example 1. . . . .	84
4.13	Flow around thin, closed structure: Example 2. . . . .	86
4.14	Shear flow with linear velocity distribution . . . . .	87
4.15	Hydrostatic pressure distribution and resulting wall forces. . . . .	88

4.16	Jeffery-Hamel flow: setup and example solutions . . . . .	90
4.17	Jeffery-Hamel flow: convergence rates. . . . .	91
4.18	Benchmark case 2D-1: velocity field. Shown is the coarsest mesh with hex8 elements. . . . .	93
4.19	3D benchmark cases 3D-1Q and 3D-1Z . . . . .	95
5.1	Deforming observed domain on a fixed fluid grid. . . . .	99
5.2	Moving domain boundary on an one-dimensional fixed fluid grid with Eulerian description of the fluid material movement. .	103
5.3	Force solution for various time-integration schemes for $\Delta t = 0.05$ s	111
5.4	Force solution for three time-integration schemes for $\Delta t = 0.05$ s	112
5.5	Force difference between computed and analytical wall force .	113
5.6	Temporal convergence for three time-integration schemes . .	114
6.1	Mortar coupling between structural surface and interface . . .	118
6.2	Coupled FSI system with traction Lagrange multipliers. . . . .	120
6.3	Interface construction using the discrete structural surface - 2D	121
6.4	Coupled FSI system with proposed stress Lagrange multipliers.	123
6.5	Refinement of quadratic elements using hanging nodes (indi- cated by diamonds). . . . .	130
6.6	Hybrid approach . . . . .	133
6.7	Flow field around a flexible, rotating cylinder. . . . .	136
6.8	2D - CFD benchmark using adaptivity . . . . .	137
6.9	2D - CFD benchmark using adaptivity near interface . . . . .	137
6.10	2D cylinder benchmark - interface fields . . . . .	138
6.11	2D CFD benchmark using the fluid-fluid coupling approach .	138
6.12	Setup and principle solution of compressible cylinder example.	139
6.13	Final meshes used for comparison. . . . .	140
6.14	Flow over bending beam . . . . .	141
6.15	Stationary channel flow interacting with a flexible structure. .	142
6.16	Setup of contact example and three snapshots of the first con- tact between a soft body and a relatively stiff block. . . . .	143
A.1	From surfaces to embedded interfaces . . . . .	150
B.1	Jeffery-Hamel flow: Profile of radial velocity . . . . .	161

# List of Tables

3.1	Exemplary shape functions for employed 3D elements. . . . .	46
4.1	Results for the stationary benchmark case 2D-1. . . . .	93
4.2	Results for the instationary benchmark case 2D-3. . . . .	94
4.3	Results for benchmark case 3D-1Q. . . . .	96
4.4	Results for benchmark case 3D-1Z. . . . .	96



# Nomenclature

## Acronyms

**ALE** arbitrary Lagrangian Eulerian.

**AMG** algebraic multigrid.

**BDF2** second order backward differencing.

**CAS** computer algebra system.

**CDT3** constrained Delaunay tetrahedralization.

**CFD** computational fluid dynamics.

**CG** continuous Galerkin.

**DBC** Dirichlet boundary condition.

**DG** discontinuous Galerkin.

**DLM/FD** distributed Lagrange multiplier / fictitious domain.

**DOF** degree of freedom.

**DOFs** degrees of freedom.

**FD** finite difference.

**FE** finite element.

**FEM** finite element method.

**FG/ALE** fixed-grid / arbitrary Lagrangian Eulerian.

**FSI** fluid-structure interaction.

**FV** finite volume.

**GFEM** generalized finite element method.

**HR** Hellinger-Reissner.

**IB** immersed boundary.

**IFEM** immersed finite element method.

**LSIC** least square incompressibility condition.

**NS** Navier-Stokes.

**ODE** ordinary differential equation.

**OST** one-step-theta.

**PFEM** particle finite element method.

**PSPG** pressure-stabilized Petrov Galerkin.

**PUFEM** partition of unity finite element method.

**SPH** smoothed particle hydrodynamics.

**STL** standard template library.

**SUPG** streamline upwind Petrov Galerkin.

**XFEM** extended finite element method.

**XFSI** XFEM based fluid-structure interaction.

## Modifiers

### Subscripts

$\cdot_{i,j}$	partial derivative $\partial \cdot_i / \partial x_j$
$\cdot_0$	at time level $t = 0$ , at reference configuration
$\cdot_D$	related to Dirichlet boundary
$\cdot_N$	related to Neumann boundary

### Superscripts

$\cdot^f$	associated with fluid field
$\cdot^s$	associated with structure field
$\cdot^i$	associated with interface
$\cdot^{\text{hist}}$	related to historical/old time step values
$\cdot^h$	discretized entity
$\cdot^p$	associated with submerged patch fluid
$\cdot^{\text{diff}}$	associated with diffusion problem

### Accents and modifiers

$\hat{\cdot}$	prescribed value, e.g. boundary conditions
$\tilde{\cdot}$	element unknowns (as opposed to nodal unknowns)
$\delta \cdot$	virtual or test function
$\llbracket \cdot \rrbracket$	step / discontinuity in field
$\mathfrak{c}$	coupling terms resulting from element-wise condensation
$\ddot{\cdot}$	second order tensor
$\ddot{\cdot}$	fourth order tensor

### Field related subscripts on discrete matrices and vectors

$\cdot_u$	fluid nodal velocity related
$\cdot_p$	fluid nodal pressure related
$\cdot_{\bar{\sigma}}$	fluid element stress related
$\cdot_{\bar{\tau}}$	fluid viscous element stress related
$\cdot_{\bar{p}}$	fluid element pressure related
$\cdot_{d^i}$	interface displacement related
$\cdot_{u^i}$	interface velocity related
$\cdot_{d^s}$	structure displacement related
$\cdot_{\lambda}$	Lagrange multiplier $\lambda$ related
$\cdot_{\mu}$	Lagrange multiplier $\mu$ related
$\cdot_{\kappa}$	Lagrange multiplier $\kappa$ related

## Coordinate systems

$x$	spatial coordinates
$X$	material or reference coordinates for Lagrangian description
$\xi^\Omega$	domain element coordinates
$\xi^\Gamma$	boundary element coordinates
$\eta^\Omega$	domain integration cell coordinates
$\eta^\Gamma$	boundary integration cell coordinates

## Domains

$\Omega^f$	fluid domain at time $t$
$\Omega^s$	structure domain at $t$
$\Gamma_N$	boundary with Neumann conditions
$\Gamma_D$	boundary with Dirichlet conditions
$\Gamma^i$	boundary treated with XFEM enrichments
$\Gamma^{\text{FSI}}$	fluid-structure interface

## Fields and associated parameters

### Fluid

$u$	velocity
$a$	acceleration
$p$	pressure
$\gamma$	strain rate
$\tau$	viscous stress
$\sigma$	Cauchy stress
$\bar{p}$	dual pressure fields in hybrid element formulations
$\bar{\tau}$	dual viscous stress fields in hybrid element formulations
$\bar{\sigma}$	dual Cauchy stress fields in hybrid element formulations
$\hat{u}_D$	prescribed Dirichlet velocity
$\hat{b}$	prescribed body forces
$\hat{h}$	prescribed Neumann boundary traction
$\rho^f$	density
$\mu$	dynamic viscosity
$\nu$	kinematic viscosity
Re	Reynolds number
$\tau_{\text{PSPG}}^e$	stabilization parameter
$\tau_{\text{SUPG}}^e$	stabilization parameter
$\tau_{\text{LSIC}}^e$	stabilization parameter

$h_e^V, h_e^u$	length scales for stabilization
<b>Structure</b>	
$d^s$	displacement
$u^s$	velocity
$a^s$	acceleration
$F$	deformation gradient
$E$	Green-Lagrange strain tensor
$S$	2 <sup>nd</sup> Piola-Kirchoff stress tensor
$\sigma^s$	Cauchy stress tensor
$\tilde{\sigma}^s$	dual Cauchy stress field in hybrid element formulation
$\hat{d}_D^s$	prescribed Dirichlet displacement
$\hat{b}^s$	prescribed body forces
$\hat{h}^s$	prescribed Neumann traction
$\rho^s$	density
$C$	4 <sup>th</sup> order elasticity tensor
$\tilde{C}_{ijkl}$	components of the elasticity tensor $\tilde{C}$
$E^s$	Young's modulus for linear elastic materials
$\nu^s$	Poisson ratio for linear elastic materials
<b>Interfaces</b>	
$d^i$	displacement
$u^i$	velocity
$\lambda$	fluid-interface traction/Lagrange multiplier
$\mu$	interface-structure traction/Lagrange multiplier
<b>Heat conduction</b>	
$c$	temperature
$q$	heat flux
$\bar{q}$	dual flux in hybrid element formulation
$\hat{c}_D$	prescribed Dirichlet temperature
$\hat{f}$	prescribed heat source in domain
$\hat{h}_N$	prescribed Neumann boundary heat source
$\kappa^{\text{diff}}$	diffusion/heat conduction coefficient
<b>Others</b>	
$\kappa$	ALE-fluid-structure traction Lagrange multiplier

## Operators

### Fluid

$r_u$	residual of momentum equation
$r_p$	residual of continuity equation
$F_M$	weak form of stabilized fluid momentum equation
$F_C$	weak form of stabilized fluid continuity equation

### Structure

$r_{ds}$	structure momentum residual
$S$	weak form of structure momentum equation

## Discretization

### Sets

$\mathcal{S}$	unconstrained fluid DOFs
$\mathcal{C}$	constrained fluid DOFs along implicit FSI surface
$\mathcal{B}$	FSI surface structure DOFs
$\mathcal{D}$	interior structure DOFs

### Time discretization

$\theta$	one-step-theta factor for fluid velocity
$\theta^i$	one-step-theta factor for interface displacement
$\Theta$	general time factor for fluid time discretization
$\beta, \gamma$	Beta-Newmark factors for structure displacement

### Summary of notation for three exemplary fields

$\mathbf{u}$	velocity field
$\delta\mathbf{u}$	virtual velocity field
$\mathbf{u}^h$	discrete velocity field
$\delta\mathbf{u}^h$	discrete virtual velocity field
$\mathbf{u}$	global vector with velocity unknowns
$\delta\mathbf{u}$	global vector with virtual velocity unknowns
$\Delta\mathbf{u}$	global velocity increment in Newton iteration
$\mathbf{u}_e$	element vector with velocity unknowns

$p$	pressure field
$\delta p$	virtual pressure field
$p^h$	discrete pressure field
$\delta p^h$	discrete virtual pressure field
$\mathbf{p}$	global vector with pressure unknowns
$\delta\mathbf{p}$	global vector with virtual pressure unknowns

---

$\Delta p$	global pressure increment in Newton iteration
$p_e$	element vector with pressure unknowns
$\tilde{\sigma}$	dual Cauchy stress field
$\delta\tilde{\sigma}$	virtual Cauchy stress field
$\tilde{\sigma}^h$	discrete Cauchy stress field
$\delta\tilde{\sigma}^h$	discrete Cauchy stress velocity field
$\bar{\sigma}$	global vector with Cauchy stress unknowns
$\delta\bar{\sigma}$	global vector with virtual Cauchy stress unknowns
$\Delta\bar{\sigma}$	global Cauchy stress increment in Newton iteration
$\bar{\sigma}_e$	element vector with Cauchy stress unknowns

## XFEM

$N(\xi)$	polynomial shape function
$\phi(\xi)$	enriched approximation function
$\psi(x(\xi))$	general enrichment function
$\bar{\psi}(x(\xi))$	void enrichment function
$\check{\psi}(x(\xi))$	jump enrichment function

## Shape functions, continuity between neighboring elements

$N^s$	structure displacement, $C^0$ -continuous
$N^i$	interface variables, $C^0$ -continuous
$N^u$	fluid velocity, $C^0$ -continuous
$N^p$	fluid pressure, $C^0$ -continuous
$N^{\bar{\sigma}}$	fluid element Cauchy stress, $C^{-1}$ -continuous
$N^{\bar{\tau}}$	fluid element viscous stress, $C^{-1}$ -continuous
$N^{\bar{p}}$	fluid element pressure, $C^{-1}$ -continuous

## Counter

$n$	time step
$i$	partitioned FSI iteration step
$k$	Newton-Raphson iteration step
$e$	element
$c$	integration cell
$q$	integration points

## Linearized system matrices

$F$	stabilized fluid
$S$	structure
$K$	domain integrals including dual element terms
$G$	boundary integrals including dual element terms
$C$	coupling matrix resulting from element-wise condensation

$M$  rectangular mortar matrix  
 $D$  square mortar matrix

Discrete residuals and force vectors

$r$  global residuum vector  
 $\hat{r}$  residual vector resulting from element-wise condensation  
 $f$  force vector

Element types with polynomial shape functions

line2 2 nodes, 1d,  $P_1$ , linear  
line3 3 nodes, 1d,  $P_1$ , quadratic

tri3 3 nodes, 2d,  $P_1$ , linear  
tri6 6 nodes, 2d,  $P_2$ , quadratic  
quad4 4 nodes, 2d,  $P_1$ , bi-linear  
quad8 8 nodes, 2d,  $Q_2^{(20)}$ , quadratic serendipity  
quad9 9 nodes, 2d,  $Q_2$ , bi-quadratic

tet4 4 nodes, 3d,  $P_1$ , linear shape functions  
tet10 10 nodes, 3d,  $P_2$ , quadratic shape functions  
hex8 8 nodes, 3d,  $Q_1$ , tri-linear shape functions  
hex20 20 nodes, 3d,  $Q_2^{(20)}$ , quadratic serendipity shape functions  
hex27 27 nodes, 3d,  $Q_2$ , tri-quadratic shape functions

# CHAPTER 1

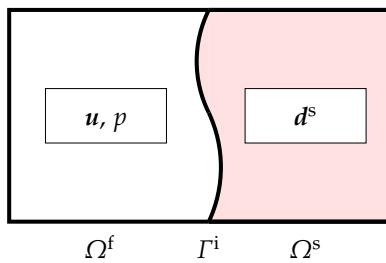
## Introduction

### 1.1 Motivation

Fluid-structure interaction (FSI) is the interaction of movable, rigid or deformable structures with internal or surrounding fluid flow. It is among the most important and, with respect to both modeling and computational issues, most challenging multiphysics problems. The variety of fluid-structure interaction (FSI) occurrences is abundant and ranges from tent-roofs to micro pumps, from parachutes via airbags to blood flow in arteries.

The numerical simulation of FSI effects is relevant in many fields of engineering as well as in the applied sciences. For engineers, numerical simulations support and verify experimental experience and they may help to reduce or avoid expensive experiments. It also guides the engineer to extrapolate known behavior towards new limits, where experiments are for example too difficult or too expensive. Examples range from wind effects on new bridge designs, gliding in mars atmosphere to stent placement in human aorta. In applied sciences, the numerical simulation is used to gain more insight into otherwise not obtainable or observable data, for instance the effect of the blood flow phenomena on aorta wall stresses or the pressure distribution inside opening airbags.

Because of such potential benefits, the development and application of FSI simulation techniques has gained great attention over the past decades. Current research areas are usually one or several of the following: the advancement from special purpose or problem specific approaches to general approaches; the ability to capture complex systems involving multiple fluid phases and multiple thin or bulky structures; the treatment of new physical effects in addition to FSI effects; the development of mathematical founda-



**Figure 1.1:** FSI as a surface coupled two-field problem: fluid field  $\Omega^f$ , structural field  $\Omega^s$  and the shared interface  $\Gamma^i$ .

tions; and robust and efficient implementations and software frameworks including mesh generation techniques.

The present work concentrates on the formulation of a general FSI approach, which shall then be extended towards more complex physical problems. The particular physical problem of interest is the interaction of incompressible flow with deformable, compressible or incompressible structures undergoing large nonlinear deformation. Examples at the Institute for Computational Mechanics at the Technische Universität München, where this thesis work has been performed, are the study of biomechanical effects in cardiovascular systems, trying to understand the properties of red blood cells during diseases or using mesoscopic FSI simulations to develop tools for medicament delivery [Mayer and Wall, 2010]. Large structure deformation may also lead to contact between approaching structures, which is an example for the extension of the general approach developed in this thesis towards incorporating more complex physics [Mayer et al., 2010].

Fluid-structure interaction is a surface coupled problem, where fluid and structure are coupled along a shared interface  $\Gamma^i$ . A general fluid-structure interaction problem statement consists of the description of fluid and solid fields, appropriate fluid-structure interface conditions at the shared interface  $\Gamma^i$  and conditions for the remaining boundaries of fluid and structure, respectively. A sketch of this setup is shown in Figure 1.1.

## 1.2 Principal approaches to FSI simulations

FSI approaches can be distinguished by various differentiating factors, for example by single field formulations, coupling schemes or discretization techniques. Among these factors, a key aspect is the formulation to describe the material motion in each participating single field. Hereby, the choice is between the Eulerian, the Lagrangian, and the arbitrary Lagrangian Eulerian (ALE) formulation.

For large deformation of structures, many finite element (FE) formulations use the Lagrangian approach to describe the material movement. If severe deformation occurs, an ALE structure formulation helps to delay remeshing and is used, for example, in simulations dealing with plastic or visco-elastic behavior. Reference [Belytschko et al., 2000, chap. 7] contains an overview on such ALE approaches for structures.

The description of the flow field in deforming fluid domains in many research and commercial codes is based on the ALE method, which is usually used in combination with the FE method. This approach goes back to early works like [Belytschko and Kennedy, 1978; Belytschko et al., 1980; Donéa et al., 1977; Hirth et al., 1974; Hughes et al., 1981; Noh, 1964]. The ALE approach is also present in recent fixed-grid methods as in [Codina et al., 2009]. The essential feature of ALE-based methods is that the fluid field is formulated and solved on a deforming grid. The fluid grid deforms such that the grid surface is aligned with the fluid-structure interface, while the interior of the fluid grid is smoothed to preserve the fluid element quality. The advantage of ALE-based methods is that the fluid flow can be described in time taking the deformation of the observed fluid domain into account. The limiting factor however is that severe fluid-structure interface movement can distort the fluid mesh in such a way that remeshing is inevitable and subsequent projections between meshes may deteriorate the accuracy of the simulation of transient effects.

Combinations of ALE and Eulerian fluid formulation have been used, where the fluid field is split into moving and non-moving grids. The deforming fluid grid is attached to the structure surface and the already mentioned ALE-fluid-structure coupling techniques are applied. Away from the structure surface, the moving grid is coupled to the fixed grid, for which various strategies can be found in literature. An example for so called overlapping fluid decomposition approaches is the Chimera method, which was originally introduced for the simulation of rigid body-fluid interaction and to simplify mesh generation [e.g. in Houzeaux and Codina, 2003; Meakin and Suhs, 1989; Steger et al., 1983; Wang and Parthasarathy, 2000]. An extension to treat flexible structures is presented in [Gamnitzer and Wall, 2006]. An example for non-overlapping approaches is the sliding mesh technique [Behr and Tezduyar, 2001, 1999] and weak fluid-fluid coupling approaches based on explicit surfaces [Bazilevs and Hughes, 2008; Gartling, 2005] or extended finite element method (XFEM) techniques [Gerstenberger and Wall, 2008a; Wall et al., 2008] (see also Section 6.5.2). The common property is that an ALE grid around the structure can freely follow the structure deformation, hence, smallest limitations to rotational and translational deformation modes apply. However, the remaining ALE grid hampers a straightforward treatment of large local structure deformation and the simulation of approaching or con-

tacting objects.

The Lagrangian approach to fluid flow is used in meshfree methods like the smoothed particle hydrodynamics (SPH) [Gingold and Monaghan, 1977; Lucy, 1977; Monaghan, 1992, 1994] and the method developed in [Idelsohn et al., 2003] as well as in mesh based methods like the particle finite element method (PFEM) [Idelsohn et al., 2004, 2006]. These approaches are appealing since complex phenomena like FSI as well as free surface and two-phase flows can be treated with algorithmic ease. In Lagrangian fluid methods, field variables are approximated using nodal unknowns and meshfree kernels or FE approximations are generated at each time instance. However, a constantly changing approximation and changing mesh with an otherwise fixed set of nodes leads to changing discrete velocity and pressure fields. Here, conservation of momentum and mass and the related surface identification requires special consideration.

Finally, so called fixed-grid methods use the Eulerian formulation to describe the fluid deformation. Prominent fixed grid methods for incompressible flow include the immersed boundary (IB) method [Peskin, 1972, 1977; Viecelli, 1969, 1971] and its many derivations as described in [Lee and LeVeque, 2003; LeVeque and Calhoun, 2001; Mittal and Iaccarino, 2005; Peskin, 2002; Wang and Liu, 2004; Zhang et al., 2004]. An FE approach with many similarities to the IB method is the distributed Lagrange multiplier / fictitious domain (DLM/FD) method [Baaijens, 2001; De Hart et al., 2000; Glowinski et al., 1994, 1999; van Loon et al., 2005; Yu, 2005]. Other fixed-grid or Cartesian-grid methods - often using finite difference (FD) or finite volume (FV) methods - are, for instance, presented in [Arienti et al., 2003; Cirak and Radovitzky, 2005; Enright et al., 2002; Fedkiw et al., 1999; Hong et al., 2007; Löhner et al., 2004; Shi and Phan-Thien, 2005; Tseng and Ferziger, 2003]. Recent reviews can be found in [Ingram et al., 2003; Löhner et al., 2008; Mittal and Iaccarino, 2005]. The key point of fixed-grid methods is that the fluid-structure interface can deform independently of any fluid element edges or grid point locations. The fluid-structure interface divides the fluid domain in a *physical* flow field and a *fictitious* field. The latter has no physical meaning to the FSI problem and requires special treatments to avoid the introduction of unphysical effects. For thin structure models like beams and shells, the interface separates independent fluid domains such that pressure and stress fields are discontinuous across the interface.

The techniques for coupling fluid and structure depend on the chosen single field formulation, but can nevertheless be sorted into two groups, namely monolithic and iterative couplings. Monolithic schemes consider the entire fluid structure system and solve for all involved degrees of freedom in one solver call. In contrast, iterative techniques allow to solve fluid and structure fields separately and forces and surface displacements are exchanged iter-

tively until an equilibrium state is reached. Both techniques belong to the strong coupling schemes. In contrast, loose coupling schemes solve the fields separately without iterating between the fields. For incompressible flow and lightweight structures, only strong coupling schemes can be applied and are exclusively used within this work. The fluid-structure coupling techniques used in this work are covered in detail in [Förster et al., 2007; Küttler and Wall, 2008, 2009; Wall, 1999].

## 1.3 Objective

For complex FSI problems, fixed grid methods have very inviting properties, namely the elimination of fluid mesh updates, fluid mesh distortion and, eventually, remeshing of the fluid domain due to the fluid-structure interface movement. More specifically, due to the independent interface description, optimally shaped elements or computational grids can be used and no mesh distortion occurs. In addition, the mesh generation is simplified, as any sufficiently fine Cartesian (i.e. axis-aligned) mesh can be used as initial mesh for an adaptive computation. However, a general and robust FE fixed-grid formulation is still missing. Hence, a three-dimensional fixed-grid FSI implementation is sought within this thesis to allow the mentioned complex FSI simulations.

### 1.3.1 Requirements

The following enumeration highlights requirements that arise when treating fluid-structure surfaces independent of the fluid grid.

**Thin and volume occupying Lagrangian structures** The structure deformation should be modeled using the Lagrangian formulation to allow reuse of available structure implementations. The method should be able to deal with both reduced structures like beams or shells and volume occupying structures. For reduced structure models pressure and strain rate discontinuities are present in the flow and need to be treated properly. In contrast to this structures, volume occupying structures introduce a fictitious domain that needs to be decoupled from the physical flow. The flow discontinuities that are present in both kinds of application may be treated using unified techniques.

**Surface coupling** Fluid-structure interaction is a surface coupled problem, where kinematic and force conditions are enforced only along the shared interface. In general, a volumetric interaction between structure and fictitious

domain limits the generality and accuracy of a fixed-grid method: First, the incompressibility constraint from the fluid carries over to the structural domain such that only incompressible structures can be computed. Although this might not be a limitation for biomechanical applications, where often water-filled and therefore incompressible tissues are considered, it is a limitation in many engineering applications, where materials and structures deform in a compressible way. Second, the double existence of fluid and structure requires the modification of the structure momentum equation to subtract the fictitious inertial and viscous forces introduced by the fictitious flow [see e.g. Wang and Liu, 2004; Yu, 2005; Zhang et al., 2004]. In other words, the FSI algorithm requires the modification of the structure momentum equation in the whole structure domain.

Volumetric interaction between fluid and structure also implies that fluid and structure degrees of freedom (DOFs) are present in the FSI system, where only structure unknowns should be present. If large portions of the computational fluid domain belong to the fictitious domain, the extra computational solution cost can not be ignored. Hence, from efficiency and computer memory point of view, the computation of fictitious fluid unknowns should be avoided. Removing fictitious fluid unknowns in turn implies surface coupling.

**Representation of kinematic/stress discontinuities** For volume occupying structures as considered in this thesis, there is a sharp interface between physical and fictitious domain. Hence, a discontinuity in the primary variables like pressure and velocity or in the derivatives like strain rate or viscous stress exists. Such a discontinuity has to be modeled correctly, otherwise momentum and mass conservation in the fluid domain deteriorates. The simulation of shell-like structures separating two fluids, where only one fluid is treated numerically, also requires a clear decoupling of the separate fluid domains.

**Advancing the interface in time** Combining the Eulerian formulation with moving boundaries requires special considerations, since the Eulerian formulation implies a fixed domain with time-independent boundaries. This discrepancy introduces numerical effects like unknown old time-step values, if the moving boundary reveals nodes that have not been within the fluid domain at the old time-step. The Ghost-fluid method as summarized in [Osher and Fedkiw, 2003] acknowledges many of the potential problems in the context of FD methods. For comparison, the ALE fluid formulation in combination with classical finite difference time discretization allows a continuous description of a moving reference independent of the material.

**Mesh size independence** Approaches like the IB method often require a certain ratio between fluid and solid mesh size at the interface to avoid artificial leakage through the impermeable structural surface. This effect has also been reported when Lagrange multiplier techniques are inappropriately used, for example without proper numerical integration. General and robust methods should allow an independent mesh size for each of the simulated fields. In addition, no limitations should apply to the thickness of the structures with respect to the fluid field grid size. This allows to model thin, but volume occupying structures instead of reduced beam or shell models.

Most methods in literature do not address all these points together such that the accuracy or the robustness of the FSI approach deteriorates. Some resulting shortcomings are traded on purpose to achieve algorithmic simplicity, which is sometimes inevitable to keep implementation and computation costs within given time or financial constraints. Consider for instance the computationally demanding simulations reported in [Peskin, 1977] more than three decades ago. With the introduction of faster computers with cheaper memory, it is possible to revisit many existing fixed-grid approaches and make them more accurate and robust.

In the following, the above considerations serve as a guideline for the development of a new fixed-grid approach.

### 1.3.2 Proposal for an XFEM and Lagrange multiplier based fixed grid method

This thesis describes a surface coupled FSI approach between a standard Lagrangian structural description and an Eulerian formulation for the fluid. The Lagrangian formulation for the structure is used as usual, such that the attention is on the fixed-grid fluid formulation and the combination with established FSI schemes.

The fluid formulation is obtained in two steps: First, the fictitious fluid domain is decoupled from the fictitious domain and, second, the remaining physical flow is coupled to the fluid structure interface. The fictitious fluid domain is completely eliminated by using features of the XFEM. The XFEM was originally introduced for the simulation of cracks and other discontinuities in structures [Belytschko and Black, 1999; Moës et al., 1999] and has been, close to the topic at hand, extended to problems of two-phase flow [Chessa and Belytschko, 2003] and Stokes flow/rigid particle interaction [Wagner et al., 2003]. The basic idea for the FSI problem is to use discontinuous velocity and pressure fields to decouple mass and momentum balance of the physical and fictitious flow regions. The XFEM approximation allows also to remove all fluid unknowns from the fictitious fluid domain.

The fluid-structure coupling takes place only along the shared interface and only between the implicit surface of the physical flow and the surface of the structure mesh. For this purpose, a new stress-based technique for imposing constraints along implicit fluid boundaries has been developed, which has been presented in [Gerstenberger et al., 2008] and [Gerstenberger and Wall, 2010]. The weak coupling is applicable to three-dimensional problems, allows independent fluid and structure mesh sizes and requires no user-defined parameters.

For time-discretization, classical finite difference schemes are employed. Although space-time methods may provide a consistent way of describing implicit interfaces in space and time as suggested in the context of XFEM in [Chessa and Belytschko, 2004; Fries and Zilian, 2009] and more specifically for XFEM-based fluid-structure interaction (FSI) methods in [Zilian and Legay, 2008], it essentially requires a partial or even a full rewriting of existing FSI software towards 4D space-time discretization [see e.g. Behr, 2008]. This conflicts with the goal of having a working fixed-grid FSI code by the time of this writing, hence traditional time-stepping schemes have solely been considered.

The fluid-structure coupling is based on the monolithic and partitioned FSI schemes presented in [Küttler and Wall, 2008; Wall, 1999]. The strict surface coupling between fluid and structure allows to reuse coupling algorithms that are originally developed and implemented for ALE based FSI schemes. In a partitioned FSI framework, the fixed-grid fluid implementation can be integrated into existing FSI codes as a replacement for ALE-based fluid implementations. The proposed FSI approach is implemented in the multi-physics code Baci that is developed at the Institute for Computational Mechanics at Technische Universität München. The parallel framework is based on the Trilinos open source library [Heroux and Willenbring, 2003; Heroux et al., 2005] that supports communications and computations on parallel distributed memory computers.

With this XFEM based fluid-structure interaction (XFSI) approach, in principle all of the mentioned critical points can be addressed. Most prominently, there is a sharp separation between physical and fictitious fluid domain, there is no influence of the fictitious fluid domain on the FSI simulation, fluid and structure are coupled only along the shared surface, and fluid and structure discretization techniques and implementation are completely independent with respect to material models, approximation and mesh sizes. For a sufficient large fictitious fluid domain, most of the unnecessary fluid unknowns can be removed and the linearized systems allow the application of iterative solvers.

## 1.4 Outline

The treatment of moving boundaries on fixed-fluid grids with consideration of the subsequent coupling to the structure surface is the main research direction of this thesis. The layout of this work is oriented towards increasing complexity of the fluid formulation, starting from the fluid finite element formulation and basic XFEM techniques and ending with moving boundaries and the FSI formulation.

In Chapter 2, the governing equations for fluid and structure and the fluid-structure coupling are reviewed. In addition, Section 2.1 and Section 2.2, the fluid and structure discretization in space and time without consideration of FSI are introduced. In Section 2.3, a fluid-structure formulation with an additional interface mesh is proposed that allows to separately treat the fluid-interface coupling and the interface-structure coupling in the remainder of the thesis.

Chapter 3 concentrates on numerical techniques for domain decomposition and weak constraints that are used for the fluid in subsequent chapters. In particular, the XFEM is reviewed using the Poisson equation as example. Beyond the review of existing coupling techniques, a new method for enforcing interface constraints is introduced. The method is developed for the specific needs of the Navier-Stokes (NS) equations, but can be applied to other physical problems as well.

The treatment of implicit interfaces for the fluid is described in two steps: the transient formulation for *non-moving* interfaces is introduced in Chapter 4. Here, two principle techniques for adding constraints along implicit fluid boundaries are derived and discussed. The chapter concludes with an in-depth numerical analysis of the XFEM and Lagrange multiplier techniques. Chapter 4 also highlights implementation details since considerable modifications to established parallel FE codes are usually required. Primary aspects are the interface treatment as additional mesh, the management of a constantly changing number of unknowns and the effect of distributed parallelism on the fixed-grid FSI approach. The extension to *moving* boundaries including the treatment of moving impermeable walls is given in Chapter 5. In both chapters, only the fluid-interface system is considered to strictly separate structure and fluid formulation.

Chapter 6 combines fluid and structure fields into the coupled FSI equations. First, the structure-interface coupling using established Lagrange multiplier/Mortar techniques is described. Subsequently, the monolithic and partitioned FSI algorithms for both the classical Lagrange multiplier and the proposed stress Lagrange multiplier are presented. Along, algorithms for the numerical solution of the partitioned, linearized system are presented. Chapter 6 also contains considerations on improving the flow solution near the

fluid-structure interface by using  $h$ -adaptivity and boundary layer meshes. The flexibility of the proposed fixed-grid approach is demonstrated on three-dimensional FSI examples including an illustrative FSI contact problem.

The conclusion in Chapter 7 reviews the key steps necessary for an accurate fixed-grid method. It also highlights requirements that are still not treated properly and suggests improvements with respect to robustness and performance, such that the fixed-grid method can be used for realistic simulations.

Appendix A demonstrates, how the new stress Lagrange multiplier approach is applied to linear elastic problems. Although no computations have been performed for linear elasticity, the elastic formulation was included to support the transition from the scalar diffusion in Chapter 3 to the incompressible Navier-Stokes equation in the Chapter 4. Appendix B demonstrates the generation of the reference solutions for the Jeffery-Hamel flow, since the required procedure requires the use of computer algebra systems and has not been described in literature so far.

# CHAPTER 2

## Governing equations and finite element formulations

Within this chapter, the modeled physics for fluid and structure field are introduced. In addition, the FE formulations for fluid and structure without consideration of FSI effects are given. The treatment of each single field is presented in much more detail in standard finite element method (FEM) literature to which references are given in each section. Subsequently, the fluid-structure coupling conditions are introduced and a new fluid-interface-structure formulation is proposed that allows a modular treatment of coupling aspects in the subsequent chapters.

### 2.1 Fluid

#### 2.1.1 Field equations

For the fluid domain  $\Omega^f$  with the position vector  $x$ , the conservation of momentum is

$$\rho^f \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \hat{\mathbf{b}}^f \quad \text{in } \Omega^f \quad (2.1)$$

Here, the material time derivative of the fluid velocity  $\mathbf{u}$  times the fluid density  $\rho^f$  is balanced by the divergence of the Cauchy stress tensor  $\boldsymbol{\sigma}$  and external, velocity independent forces  $\hat{\mathbf{b}}^f$ . Mass conservation for an incompressible fluid is expressed by

$$\nabla \cdot \mathbf{u} = 0 \quad (2.2)$$

For incompressible, viscous flows, the Cauchy stress is split into pressure  $p$  using the unit or identity tensor  $\mathbf{I}$  and the deviatoric stress tensor  $\boldsymbol{\tau}$  given as

$$\boldsymbol{\sigma}^f = -p\mathbf{I} + \boldsymbol{\tau} \quad (2.3)$$

Within this work, the Newtonian material law is solely used, which defines  $\boldsymbol{\tau}$  as

$$\boldsymbol{\tau} = 2\mu\boldsymbol{\gamma} \quad (2.4)$$

where  $\mu$  denotes the dynamic viscosity and  $\boldsymbol{\gamma}$  refers to the strain rate tensor given by

$$\boldsymbol{\gamma} = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) \quad (2.5)$$

The also used kinematic viscosity is defined as  $\nu = \mu/\rho^f$ . Wherever the context allows a clear distinction between fluid and structure variables, the superscript  $\cdot^f$  is omitted on primary and derived field variables.

The specific form of the material time derivative depends on the choice of the reference system that is employed to formulate the problem. The choice of a specific reference system is dictated by the way the moving interface is treated. There are basically three alternative reference systems: the Eulerian, the Lagrangian and the ALE formulation. For the fixed background grid as used in this work, the Eulerian formulation is used. In this case the material time derivative becomes the sum of a partial time derivative and a convective term

$$\rho^f \frac{\partial \mathbf{u}}{\partial t} \Big|_x = -\rho^f \mathbf{u} \cdot \nabla \mathbf{u} - \nabla p + \nabla \cdot \boldsymbol{\tau} + \hat{\mathbf{b}}^f \quad \text{in } \Omega^f \quad (2.6a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega^f \quad (2.6b)$$

The partial time derivative is the acceleration  $\mathbf{a}$  of the fluid material point and is defined by

$$\frac{\partial \mathbf{u}}{\partial t} \Big|_x = \mathbf{a} \quad (2.7)$$

Boundary conditions away from the fluid-structure interface are either of Dirichlet type or Neumann type. They are defined as

$$\mathbf{u} = \hat{\mathbf{u}}_D \quad \text{in } \Gamma_D^f \quad (2.8)$$

$$\boldsymbol{\sigma}^f \cdot \mathbf{n}^f = \hat{\mathbf{h}}^f \quad \text{in } \Gamma_N^f \quad (2.9)$$

The corresponding boundary sections are denoted by  $\Gamma_D^f$  and  $\Gamma_N^f$ , respectively. The superscript  $\hat{\cdot}$  above a variable indicates a known and given field.

## 2.1.2 Discretization

### Time discretization

In this thesis, two different time discretization schemes are employed: the one-step-theta (OST) method and the second order backward differencing (BDF2) method.

In the OST method, the acceleration  $\mathbf{a}$  is defined as a combination of the acceleration at the new time step level  $n + 1$  and the previous time step level  $n$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \theta \mathbf{a}^{n+1} + (1 - \theta) \mathbf{a}^n \quad (2.10)$$

No time derivatives are present for the pressure and stress fields, hence they are unknowns at the new time step level only. Values from the previous time step are not required. Sorting Eq. (2.10) for expressions at the new and the old time step gives

$$\mathbf{u}^{n+1} - \Delta t \theta \mathbf{a}^{n+1} = \mathbf{u}^n + \Delta t (1 - \theta) \mathbf{a}^n \quad (2.11)$$

The new acceleration  $\mathbf{a}^{n+1}$  is replaced by the right hand side of Eq. (2.6a) evaluated at the new time step  $n + 1$

$$\rho^f \mathbf{u}^{n+1} + \Delta t \theta \left[ \rho^f \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot \boldsymbol{\tau} - \hat{\mathbf{b}}^f \right]^{n+1} = \rho^f \mathbf{u}^n + \Delta t (1 - \theta) \rho^f \mathbf{a}^n \quad (2.12)$$

in  $\Omega^f$ . Eq. (2.12) is divided by  $\Delta t \theta$  to avoid numerous repetition of the time scaling factor  $\Delta t \theta$  in subsequent chapters

$$\frac{\rho^f}{\Delta t \theta} \mathbf{u}^{n+1} + \rho^f \mathbf{u}^{n+1} \cdot \nabla \mathbf{u}^{n+1} + \nabla p^{n+1} - \nabla \cdot \boldsymbol{\tau}^{n+1} - \hat{\mathbf{b}}^{f,n+1} = \frac{\rho^f}{\Delta t \theta} \mathbf{u}^n + \frac{1 - \theta}{\theta} \rho^f \mathbf{a}^n \quad (2.13)$$

For the complete time-discrete strong form, the incompressibility is enforced at the new time step

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (2.14)$$

Once the new velocity  $\mathbf{u}^{n+1}$  has been computed, the corresponding new acceleration  $\mathbf{a}^{n+1}$  is updated by

$$\mathbf{a}^{n+1} = \frac{1}{\Delta t \theta} (\mathbf{u}^{n+1} - \mathbf{u}^n) - \frac{1 - \theta}{\theta} \mathbf{a}^n \quad (2.15)$$

The user-defined initial velocity field  $\mathbf{u}^0$  has to fulfill the NS equations including the incompressibility constraint. More specifically, a discrete divergence-free initial velocity has to be ensured. The initial acceleration field  $\mathbf{a}^0$  is formally not required to be divergence-free. In this work, all transient computations are started with zero velocity and zero acceleration, hence, also the

initial acceleration is divergence free. The right hand side of Eq. (2.11) is referred to as

$$\boldsymbol{u}_\theta^{\text{hist}} = \boldsymbol{u}^n + \Delta t(1 - \theta)\boldsymbol{a}^n \quad (2.16)$$

Using this short notation and omitting the superscript  $n + 1$  for the new time step, the time-discrete NS-equations are written as

$$\frac{\rho^f}{\Theta_\theta} \boldsymbol{u} + \rho^f \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p - \nabla \cdot \boldsymbol{\tau} - \hat{\boldsymbol{b}}^f - \frac{\rho^f}{\Theta_\theta} \boldsymbol{u}_\theta^{\text{hist}} = \mathbf{0} \quad \text{in } \Omega^f \quad (2.17a)$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \Omega^f \quad (2.17b)$$

with  $\Theta_\theta = \Delta t \theta$ .

The BDF2 method requires two previous time step values and discretizes the velocity time derivative as

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\Delta t} = \frac{1}{3} \frac{\boldsymbol{u}^n - \boldsymbol{u}^{n-1}}{\Delta t} + \frac{2}{3} \boldsymbol{a}^{n+1} \quad (2.18)$$

The time-discrete NS equations using the BDF2 method are given as

$$\frac{\rho^f}{\Theta_{\text{BDF2}}} \boldsymbol{u} + \rho^f \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p - \nabla \cdot \boldsymbol{\tau} - \hat{\boldsymbol{b}}^f - \frac{\rho^f}{\Theta_{\text{BDF2}}} \boldsymbol{u}_{\text{BDF2}}^{\text{hist}} = \mathbf{0} \quad \text{in } \Omega^f \quad (2.19a)$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \Omega^f \quad (2.19b)$$

with  $\Theta_{\text{BDF2}} = \frac{2}{3} \Delta t$  and with the abbreviation  $\boldsymbol{u}_{\text{BDF2}}^{\text{hist}}$  for all old time step (historical) values of the velocity defined as

$$\boldsymbol{u}_{\text{BDF2}}^{\text{hist}} = \frac{4}{3} \boldsymbol{u}^n - \frac{1}{3} \boldsymbol{u}^{n-1} \quad (2.20)$$

Again,  $\boldsymbol{u}^n$  and  $\boldsymbol{u}^{n-1}$  have to be divergence free. For the computations using the BDF2 method, an initial one-step- $\theta$  step has been used.

For a discussion on startup procedures see e.g. [Gresho and Sani, 2000] or [Donéa and Huerta, 2003]. The subscript for the OST and BDF2 methods is dropped in the following as both schemes result in identical matrix structures with only the terms  $\boldsymbol{u}^{\text{hist}}$  and  $\Theta$  being different.

## Spatial discretization

For spatial discretization, the FEM is applied. The time-discrete strong form in Eq. (2.19) is a pure spatial differential equation in  $\Omega^f$ . For the FEM, the strong form is brought into its weighted residual form. For this purpose, the residuals  $\boldsymbol{r}_u(\boldsymbol{u}, p)$  and  $\boldsymbol{r}_p(\boldsymbol{u})$  are defined as

$$\boldsymbol{r}_u(\boldsymbol{u}, p) = \frac{\rho^f}{\Theta} \boldsymbol{u} + \rho^f \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p - \nabla \cdot \boldsymbol{\tau}(\boldsymbol{u}) - \hat{\boldsymbol{b}}^f - \frac{\rho^f}{\Theta} \boldsymbol{u}^{\text{hist}} \quad (2.21a)$$

$$\boldsymbol{r}_p(\boldsymbol{u}) = \nabla \cdot \boldsymbol{u} \quad (2.21b)$$

The residuals vanish, if the correct solution that fulfills the partial differential equations is inserted. Without consideration of moving interfaces, the weighted residual form of the NS equations is established by testing the residuals with arbitrary velocity and pressure test functions  $\delta\mathbf{u}$  and  $\delta p$

$$0 = (\delta\mathbf{u}, \mathbf{r}_u(\mathbf{u}, p))_{\Omega^f} - (\delta\mathbf{u}, \boldsymbol{\sigma} \cdot \mathbf{n}^f - \hat{\mathbf{h}})_{\Gamma_N^f} \quad (2.22a)$$

$$0 = (\delta p, r_p(\mathbf{u}))_{\Omega^f} \quad (2.22b)$$

which is equivalent to writing

$$0 = \int_{\Omega^f} \delta\mathbf{u} \cdot \mathbf{r}_u(\mathbf{u}, p) dx - \int_{\Gamma_N^f} \delta\mathbf{u} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}^f - \hat{\mathbf{h}}) dx \quad (2.23a)$$

$$0 = \int_{\Omega^f} \delta p r_p(\mathbf{u}) dx \quad (2.23b)$$

The operator  $(\cdot, \cdot)_{\Omega^f}$  denoting the inner product of two quantities is used throughout this work.

Within the weighted residual, the term containing the Cauchy stress is integrated by parts to lower the continuity requirement of the required approximation function

$$(\delta\mathbf{u}, \nabla \cdot \boldsymbol{\sigma}^f)_{\Omega^f} = (\delta\mathbf{u}, \boldsymbol{\sigma}^f \cdot \mathbf{n}^f)_{\Gamma^f} - (\nabla \delta\mathbf{u}, \boldsymbol{\sigma}^f)_{\Omega^f} \quad (2.24)$$

Dirichlet and Neumann conditions are applied as usual. The test function approximation is chosen such that the discrete test function is zero at Dirichlet boundaries. For Neumann conditions, Eq. (2.9) is used, where  $\hat{\mathbf{h}}$  equals the surface traction. After integrating the stress term by parts, the weak form becomes

$$B_M(\delta\mathbf{u}, \mathbf{u}, p) = 0 \quad (2.25a)$$

$$B_C(\delta p, \mathbf{u}) = 0 \quad (2.25b)$$

with

$$\begin{aligned} B_M(\delta\mathbf{u}, \mathbf{u}, p) &= (\delta\mathbf{u}, \frac{\rho^f}{\Theta} \mathbf{u})_{\Omega^f} + (\delta\mathbf{u}, \rho^f \mathbf{u} \cdot \nabla \mathbf{u})_{\Omega^f} + (\nabla \delta\mathbf{u}, -p \mathbf{I} + \boldsymbol{\tau})_{\Omega^f} \\ &\quad - (\delta\mathbf{u}, \hat{\mathbf{b}}^f)_{\Omega^f} - (\delta\mathbf{u}, \hat{\mathbf{h}}^f)_{\Gamma_N^f} - (\delta\mathbf{u}, \frac{\rho^f}{\Theta} \mathbf{u}^{\text{hist}})_{\Omega^f} \end{aligned} \quad (2.26a)$$

$$B_C(\delta p, \mathbf{u}) = (\delta p, \nabla \cdot \mathbf{u})_{\Omega^f} \quad (2.26b)$$

The time-discrete equations combined with the already defined boundary and initial conditions are discretized in space using shape functions  $N^u(\mathbf{x})$

and  $N^p(x)$ :

$$\mathbf{u}^{h,n+1}(x) = \sum_I N_I^u(x) \mathbf{u}_I^{n+1} \quad (2.27a)$$

$$\delta\mathbf{u}^h(x) = \sum_I N_I^u(x) \delta\mathbf{u}_I \quad (2.27b)$$

$$p^{h,n+1}(x) = \sum_I N_I^p(x) p_I^{n+1} \quad (2.27c)$$

$$\delta p^h(x) = \sum_I N_I^p(x) \delta p_I \quad (2.27d)$$

Hereby, the superscript ‘h’ indicates the discretized field function. Approximations for the unknown functions and the corresponding test functions are identical, which results in the Bubnov-Galerkin method. In addition, velocity and pressure approximations are also identical ( $N_I^u(x) = N_I^p(x)$ ), for which the term ‘equal order velocity-pressure interpolation’ is often used in literature.

The resulting discrete nonlinear system has a shape as

$$\frac{1}{\Theta} \mathbf{M}\mathbf{u} + \mathbf{f}_{\text{conv}}(\mathbf{u}) + \mathbf{G}\mathbf{p} + \mathbf{K}\mathbf{u} = \mathbf{f}_b + \mathbf{f}_N + \mathbf{f}^{\text{hist}} \quad (2.28a)$$

$$\mathbf{G}^T \mathbf{u} = \mathbf{0} \quad (2.28b)$$

The velocity and pressure unknown vectors are denoted as  $\mathbf{u}$  and  $\mathbf{p}$ , respectively. The source terms on the right of Eq. (2.28a) are the result of the integrals with volumetric forces, Neumann conditions and old time step values as defined in Eq. (2.26a). For the employed Newtonian fluid with constant viscosity, the only source of nonlinearity is the convection term  $\mathbf{f}_{\text{conv}}(\mathbf{u})$ .

### 2.1.3 Stabilization

In this work, linear and quadratic equal order shape functions for velocity and pressure space discretization are used. Such a formulation is known to be unstable for two reasons: (inf-sup) unstable pairs of velocity and pressure shape functions and dominating convection. As a remedy, stabilized methods are employed; for an overview on stabilized methods for fluids, see for example [Hughes et al., 2004]. In this thesis, the stabilization given in [Tezduyar and Osawa, 2000] has been used, where the momentum and continuity operators are defined as

$$\begin{aligned} F_M(\delta\mathbf{u}, \mathbf{u}, p) &= B_M(\delta\mathbf{u}, \mathbf{u}, p) - \sum_e \tau_{\text{SUPG}}^e (\mathbf{u} \cdot \nabla \delta\mathbf{u}, \mathbf{r}^u(\mathbf{u}, p))_{\Omega_e^f} \\ &\quad + \sum_e \tau_{\text{LSIC}}^e (\nabla \cdot \delta\mathbf{u}, \nabla \cdot \mathbf{u})_{\Omega_e^f} \end{aligned} \quad (2.29a)$$

$$F_C(\delta p, \mathbf{u}, p) = B_C(\delta p, \mathbf{u}) - \sum_e \tau_{\text{PSPG}}^e (\nabla \delta p, \mathbf{r}^u(\mathbf{u}, p))_{\Omega_e^f} \quad (2.29b)$$

The stabilization parameters are defined per element, hence, a summation over element integrals is used in Eq. (2.29). The abbreviations for the stabilization terms are as follows: Streamline upwind Petrov Galerkin (SUPG), Least square incompressibility condition (LSIC) and Pressure-stabilized Petrov Galerkin (PSPG).

There are several popular definitions of the stabilization parameters in fluid mechanics literature and a discussion of advantages and disadvantages is beyond the scope of this thesis. The definition used exclusively in this work follows [Barrenechea and Valentin, 2002; Codina and Blasco, 2002; Franca and Valentin, 2000; Wall, 1999], where two stabilization parameters  $\tau_M^e$  and  $\tau_C^e$  for each element  $e$  are given as

$$\tau_M^e(h_e) = \min \left[ \Delta t, \frac{h_e}{2|\mathbf{u}^h|_{L_2}}, \frac{m_e h_e^2 \rho^f}{4\mu} \right] \quad (2.30)$$

$$\tau_C^e(h_e) = \frac{|\mathbf{u}^h|_{L_2} h_e}{2} \min(\text{Re}_e(h_e), 1) \quad (2.31)$$

Here, the term  $|\mathbf{u}^h|_{L_2}$  denotes the  $L_2$  norm of the velocity field. The element Reynolds number  $\text{Re}_e(h_e)$  is defined as

$$\text{Re}_e(h_e) = \frac{m_e |\mathbf{u}^h|_{p=2} h_e \rho^f}{2\mu} \quad (2.32)$$

For linear shape functions, the constant  $m_e$  is defined as  $m_e = 1/3$ , for quadratic shape functions it is  $m_e = 1/12$ . The necessary element dimension  $h_e$  can be chosen based on the element volume  $V_e$  as

$$h_e^V = \sqrt[3]{V_e} \quad (2.33)$$

or the approximate stream-length approach, where the stream-length  $h_e^u$  is evaluated at each integration point in the element parameter space  $\xi^\Omega$  by

$$h_e^u(\xi^\Omega) = \frac{2}{\sum_I \left| \frac{\mathbf{u}}{\|\mathbf{u}\|} \cdot \nabla N_I(\xi^\Omega) \right|} \quad (2.34)$$

as given in [Tezduyar et al., 1992c] and [Wall, 1999]. here, operator  $|\cdot|$  denotes the absolute value of the individual vector components. Using  $\tau_M^e$  and  $\tau_C^e$ , the stabilization parameters  $\tau_{\text{SUPG}}^e$ ,  $\tau_{\text{PSPG}}^e$ , and  $\tau_{\text{LSIC}}^e$  are computed based on the following element sizes

$$\tau_{\text{SUPG}}^e = \tau_M^e(h_e^u) \quad (2.35)$$

$$\tau_{\text{PSPG}}^e = \tau_M^e(h_e^V) \quad (2.36)$$

$$\tau_{\text{LSIC}}^e = \tau_C^e(h_e^u) \quad (2.37)$$

The stabilization introduces second order spatial derivatives, which linear and bi-linear elements are unable to represent completely. Hence, quadratic elements have initially been favored in this thesis to avoid spoiling the numerical analysis of the XFEM and Lagrange multiplier techniques. Nevertheless, linear approximations have shown no problems with respect to stability and robustness and have also been used as indicated in the numerical examples.

### 2.1.4 Linearization and solution

The discrete nonlinear system of equations is defined as

$$\delta \mathbf{u} \mathbf{r}_u(\mathbf{u}, \mathbf{p}) = F_M(\delta \mathbf{u}^h, \mathbf{u}^h, p^h) = 0 \quad (2.38a)$$

$$\delta \mathbf{p} \mathbf{r}_p(\mathbf{u}, \mathbf{p}) = F_C(\delta p^h, \mathbf{u}^h, p^h) = 0 \quad (2.38b)$$

For the solution, a Newton-Raphson method is employed. For this purpose, the system is linearized by applying a Taylor expansion to Eq. (2.38). The residual at iteration step  $k + 1$  is given as

$$\begin{bmatrix} \mathbf{r}_u \\ \mathbf{r}_p \end{bmatrix}_{k+1} = \mathbf{0} = \begin{bmatrix} \mathbf{r}_u \\ \mathbf{r}_p \end{bmatrix}_k + \begin{bmatrix} \frac{\partial \mathbf{r}_u}{\partial \mathbf{u}} & \frac{\partial \mathbf{r}_u}{\partial \mathbf{p}} \\ \frac{\partial \mathbf{r}_p}{\partial \mathbf{u}} & \frac{\partial \mathbf{r}_p}{\partial \mathbf{p}} \end{bmatrix}_k \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{p} \end{bmatrix} + \text{h.o.t.} \quad (2.39)$$

where h.o.t. abbreviates higher order terms. The derivatives lead to the tangent matrix of the linearized system. In the subsequent chapters, the linearized fluid system of equations is written in an easier to read notation as

$$\begin{bmatrix} \mathbf{r}_u \\ \mathbf{r}_p \end{bmatrix}_{k+1} = \mathbf{0} = \begin{bmatrix} \mathbf{r}_u \\ \mathbf{r}_p \end{bmatrix}_k + \begin{bmatrix} \mathbf{F}_{uu} & \mathbf{F}_{up} \\ \mathbf{F}_{pu} & \mathbf{F}_{pp} \end{bmatrix}_k \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{p} \end{bmatrix} + \text{h.o.t.} \quad (2.40)$$

If higher order terms of the Taylor expansion are omitted, the linear system that needs to be solved to obtain the improved solution at  $k + 1$  is given as

$$\begin{bmatrix} \mathbf{F}_{uu} & \mathbf{F}_{up} \\ \mathbf{F}_{pu} & \mathbf{F}_{pp} \end{bmatrix}_k \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{p} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_u \\ \mathbf{r}_p \end{bmatrix}_k \quad (2.41)$$

After solving the linear system, velocity and pressure are updated as

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix}_{k+1} = \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix}_k + \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{p} \end{bmatrix} \quad (2.42)$$

and the process is repeated until a user-defined convergence criteria is met.

Only the convective term is linearized, for which the linearization process has been presented in many FEM books for fluid mechanics, for example in [Donéa and Huerta, 2003, chap. 6.7]. Note that although the stabilization parameters are velocity dependent, they are not linearized; instead, they are computed from the velocity of the last Newton-Raphson iteration step.

The application of Dirichlet conditions is implemented without changing the size of the global system. The known Dirichlet values are set at the beginning of Newton step  $k$  and then all element matrices and residuals are calculated and assembled into the global system. Here, the corresponding residuals to the Dirichlet DOFs are set to zero; the system matrix is modified by zeroing out off-diagonal terms in corresponding rows. Then the system can be solved including the Dirichlet DOFs.

## 2.2 Structure

### 2.2.1 Field equations

In most applications, the structure in  $\Omega^s$  is described using a Lagrangian description, where the material time derivative becomes a simple partial derivative with respect to time, such that

$$\rho^s \frac{\partial^2 \mathbf{d}^s}{\partial t^2} = \nabla \cdot \boldsymbol{\sigma}^s + \rho^s \hat{\mathbf{b}}^s \quad \text{in } \Omega^s \quad (2.43)$$

with the displacement  $\mathbf{d}^s$  defined as the difference between the current position  $\mathbf{x}$  and the initial position  $\mathbf{X}$  and  $\hat{\mathbf{b}}^s$  being an external acceleration field acting on the structural domain. The structural velocity  $\mathbf{u}^s$  and acceleration  $\mathbf{a}^s$  are defined as

$$\mathbf{a}^s = \frac{\partial \mathbf{u}^s}{\partial t} = \frac{\partial^2 \mathbf{d}^s}{\partial t^2} \quad (2.44)$$

For large structural deformations it is common to describe the constitutive equation using a stress-strain relation based on the Green-Lagrange strain tensor  $\mathbf{E}$  and the 2. Piola-Kirchhoff stress tensor  $\mathbf{S}(\mathbf{E})$  as a function of  $\mathbf{E}$ . The 2. Piola-Kirchhoff stress can be obtained from the Cauchy stress  $\boldsymbol{\sigma}$  as

$$\mathbf{S} = J \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \quad (2.45)$$

Here,  $J$  denotes the determinant of the deformation gradient tensor  $\mathbf{F}$ , which itself is defined as

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \quad (2.46)$$

The Green-Lagrange strain tensor  $\mathbf{E}$  is given as

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) \quad (2.47)$$

The equation of motion is usually expressed in the reference configuration, which can be found in text books on nonlinear FE for structures. The overall

behavior can be nonlinear as large deformation and rotation of the structure is allowed. For the numerical results presented in this article, St.-Venant-Kirchhoff and Neo-Hookean material laws are employed. The former describes a linear relation between  $\underline{\sigma}$  and  $\underline{E}$ , the latter a nonlinear relation.

The complete strong form can be given by

$$\rho^s \frac{\partial^2 \mathbf{d}^s}{\partial t^2} = \nabla \cdot \underline{\sigma}^s + \rho^s \hat{\mathbf{b}}^s \quad \text{in } \Omega^s \quad (2.48a)$$

$$\underline{\sigma}^s \cdot \mathbf{n}^s = \hat{\mathbf{h}}^s \quad \text{in } \Gamma_N^s \quad (2.48b)$$

$$\mathbf{d}^s = \hat{\mathbf{d}}^s \quad \text{in } \Gamma_D^s \quad (2.48c)$$

Initial conditions for the structural displacements and velocities are given as

$$\mathbf{d}^s(\mathbf{X}, t=0) = \mathbf{d}^{s,0} \quad (2.49)$$

$$\mathbf{u}^s(\mathbf{X}, t=0) = \mathbf{u}^{s,0} \quad (2.50)$$

The weak form of the structural momentum equation from Eq. (2.53) without FSI interface condition reads as

$$0 = (\delta \mathbf{d}^s, \rho^s \dot{\mathbf{d}}^s)_{\Omega^s} - (\delta \mathbf{d}^s, \nabla \cdot \underline{\sigma}^s)_{\Omega^s} \quad (2.51)$$

Integration by parts of the stress term gives

$$-(\delta \mathbf{d}^s, \nabla \cdot \underline{\sigma}^s)_{\Omega^s} = -(\delta \mathbf{d}^s, \underline{\sigma}^s \cdot \mathbf{n}^s)_{\Gamma^s} + (\nabla \cdot \delta \mathbf{d}^s, \underline{\sigma}^s)_{\Omega^s} \quad (2.52)$$

For Dirichlet boundary conditions, the surface integral vanishes due to the vanishing test function. Neumann conditions are applied by letting  $\underline{\sigma}^s \cdot \mathbf{n}^s = \hat{\mathbf{h}}^s$ . The weak form after integration by parts without consideration of the fluid-structure coupling is

$$S(\delta \mathbf{d}^s, \mathbf{d}^s) = \underbrace{(\delta \mathbf{d}^s, \rho^s \frac{\partial^2 \mathbf{d}^s}{\partial t^2})_{\Omega^s}}_{\delta W_{\text{inert}}} + \underbrace{(\nabla \delta \mathbf{d}^s, \underline{\sigma}^s)_{\Omega^s}}_{\delta W_{\text{int}}} - \underbrace{(\delta \mathbf{d}^s, \hat{\mathbf{h}}^s)_{\Gamma_N^s}}_{\delta W_N} = 0 \quad (2.53)$$

## 2.2.2 Discretization

For structural time integration, a  $\beta$ -Newmark scheme [Belytschko et al., 2000; Hughes and Liu, 1978] is employed with parameters set to  $\beta = 0.25$  and  $\gamma = 0.5$  making it an implicit time integration scheme

$$\mathbf{a}^{n+1} = \frac{1}{\beta \Delta t^2} (\mathbf{d}^{n+1} - \mathbf{d}^n) - \frac{1}{\beta \Delta t} \mathbf{u}^n - \frac{1-2\beta}{2\beta} \mathbf{a}^n \quad (2.54)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \gamma \Delta t \mathbf{a}^{n+1} + (1-\gamma) \Delta t \mathbf{a}^n \quad (2.55)$$

With the definition of time history contributions as

$$\mathbf{d}^{\text{hist}} = \mathbf{d}^n + \Delta t \mathbf{u}^n + \left( \frac{1}{2} - \beta \right) \Delta t^2 \mathbf{a}^n \quad (2.56)$$

the time discrete form of Eq. (2.53) can be written as

$$S(\delta \mathbf{d}^s, \mathbf{d}^s) = \underbrace{\frac{\rho^s}{\beta \Delta t^2} (\delta \mathbf{d}^s, \mathbf{d}^s - \mathbf{d}^{s,\text{hist}})}_{\delta W^{\text{inert}}} + \underbrace{(\nabla \delta \mathbf{d}^s, \boldsymbol{\sigma}^s)}_{\delta W^{\text{int}}} - \underbrace{(\delta \mathbf{d}^s, \hat{\mathbf{h}}^s)}_{\delta W_N} = 0 \quad (2.57)$$

The spatial discretization of the structure displacement is performed using shape function  $N_I^s(x)$

$$\mathbf{d}^{s,h} = \sum_I N_I^s(x) \mathbf{d}_I^s \quad \text{and} \quad \delta \mathbf{d}^{s,h} = \sum_I N_I^s(x) \delta \mathbf{d}_I^s \quad (2.58)$$

In this, tri-linear hexahedral (hex8) solid elements have been used exclusively. Hence, the structural surface is a piecewise bi-linear surface.

Note that above, only a displacement based formulation is given. In the large deformation examples given in subsequent chapters, the structure is discretized in space using various (hybrid-mixed) element techniques [e.g. Belytschko et al., 2000; Bischoff et al., 2004; Pian and Wu, 2006]. Ignoring incompressible structure material with separate pressure unknowns, most techniques result in improved stiffness matrices with only displacement unknowns in the final system. Hence, the following linearized system can exemplify be taken for a wide range of FE implementations for structures and no further reference on complex structural material is made throughout this thesis.

### 2.2.3 Linearization

For nonlinear material laws and large deformation including large strains, the equations for the internal work need to be linearized before they can be solved numerically. Additional linearization is required, if deformation dependent external forces are involved. Linearization can be performed on the continuous system or the discrete system. In the following, the discrete system is linearized.

Due to the arbitrariness of the test function, the nonlinear discrete system with the global displacement vector  $\mathbf{d}^s$  and the corresponding global residual  $\mathbf{r}_{d^s}^s$  and nodal forces  $\mathbf{f}^{s,\text{int}}$ ,  $\mathbf{f}^{s,\text{ext}}$  and  $\mathbf{f}^{s,\text{dyn}}$  can be written as

$$\mathbf{r}_{d^s}^s(\mathbf{d}^s) = \frac{1}{\beta \Delta t^2} \mathbf{M} \mathbf{d}^s + \mathbf{f}^{s,\text{int}}(\mathbf{d}^s) - \mathbf{f}^{s,\text{ext}}(\mathbf{d}^s) - \mathbf{f}^{s,\text{dyn}} \quad (2.59)$$

If the total Lagrangian formulation is used, e.g. in [Belytschko et al., 2000], a constant mass matrix is obtained that is denoted as  $\mathbf{M}^s$ . The force vector  $\mathbf{f}^{s,\text{dyn}}$  resulting from old time step values is denoted as

$$\mathbf{f}^{\text{dyn}} = \mathbf{M} \left( \frac{1}{\Delta t^2} \mathbf{d}^{s,n} + \frac{1}{\beta \Delta t} \mathbf{u}^{s,n} + \frac{1 - 2\beta}{2\beta} \mathbf{a}^{s,n} \right) \quad (2.60)$$

The global velocity  $\mathbf{u}^{s,n}$  and acceleration  $\mathbf{a}^{s,n}$  have to be stored to be used in the next time step. In the following the superscript  $\cdot^{n+1}$  is implicitly assumed and only old time steps are marked with superscript  $\cdot^n$ . Adopting the notation introduced for the fluid formulations, the above discretized weak form is also denoted as

$$\delta \mathbf{d}^s \mathbf{r}_{ds}^s(\mathbf{d}^s) = S(\delta \mathbf{d}^{s,h}, \mathbf{d}^{s,h}) \quad (2.61)$$

The nonlinear discrete system can be solved using a Newton-Raphson iteration as follows: To get the improved solution  $\mathbf{d}_{k+1}^s$  at step  $k+1$ , the discrete nonlinear system needs to be linearized at step  $k$  using a Taylor expansion

$$\mathbf{r}_{ds}^s(\mathbf{d}_{k+1}^s) = \mathbf{0} = \mathbf{r}_{ds}^s(\mathbf{d}_k^s) + \underbrace{\frac{\partial \mathbf{r}_{ds}^s(\mathbf{d}^s)}{\partial \mathbf{d}} \Big|_k}_{S_{ds,ds}} \Delta \mathbf{d}^s + \text{h.o.t.} \quad (2.62)$$

If higher order terms (h.o.t.) are omitted the linear system with  $\Delta \mathbf{d}^s$  as unknowns becomes

$$[S_{ds,ds}]_k [\Delta \mathbf{d}^s] = -[\mathbf{r}_{ds}^s]_k \quad (2.63)$$

After solving the linear system for the displacement increments  $\Delta \mathbf{d}^s$ , displacements at the new iteration step  $k+1$  are updated as

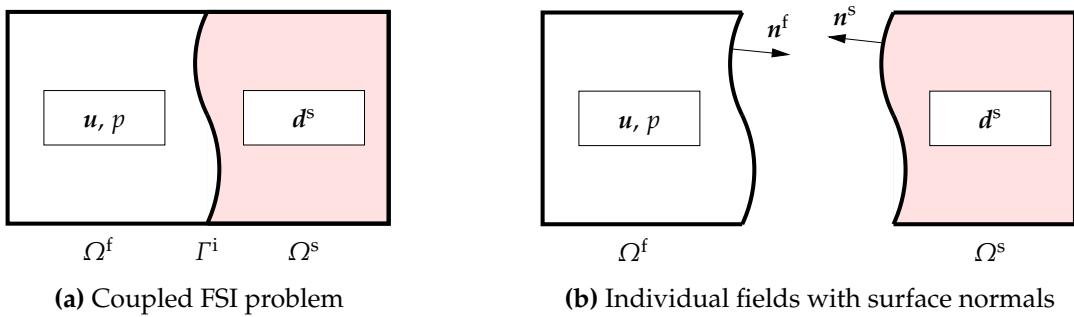
$$\mathbf{d}_{k+1}^s = \mathbf{d}_k^s + \Delta \mathbf{d}^s \quad (2.64)$$

The iteration stops when a user-defined convergence criterion is reached.

## 2.3 Coupled fluid-structure system

A general fluid-structure interaction problem consists of the description of fluid and structure fields, appropriate conditions at the shared interface and conditions for the remaining boundaries away from the fluid-structure interface. A schematic sketch of the general problem is shown in Figure 2.1a, where the interface  $\Gamma^i$  separates the structural domain  $\Omega^s$  from the fluid domain  $\Omega^f$ .

FSI is a surface coupled problem, where the interaction of the two continua takes place only along the shared interface  $\Gamma^i$ . If the simulation approach conforms to this physical fact it is usually much simpler to independently replace one structure formulation with another or to replace e.g. a compressible with an incompressible flow solver.



**Figure 2.1:** FSI two-field setup and individual fields: fluid field  $\Omega^f$ , structural field  $\Omega^s$  and the shared interface  $\Gamma^i$ .

### 2.3.1 Fluid-structure interface conditions

The main conditions at the interface are the dynamic and kinematic coupling conditions. In this work, impermeable structure surfaces are assumed such that no mass flow occurs across the interface. Consequently, the normal velocities at the interface have to match as

$$\mathbf{u} \cdot \mathbf{n}^f = -\frac{\partial \mathbf{d}^s}{\partial t} \cdot \mathbf{n}^s \quad \text{in } \Gamma^i \quad (2.65)$$

Note the opposite signs due to the different normal vectors  $\mathbf{n}^f$  and  $\mathbf{n}^s$  for fluid and structural domain, respectively (Figure 2.1b). If viscous fluids are considered, there is usually also a matching condition for the tangential velocities, which can be combined with the equation above to obtain the ‘no slip’ boundary conditions as

$$\mathbf{u} = \frac{\partial \mathbf{d}^s}{\partial t} \quad \text{in } \Gamma^i \quad (2.66)$$

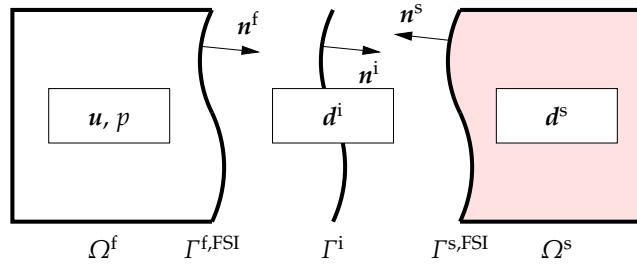
The force equilibrium requires the surface traction to be equal as

$$\underline{\sigma}^f \cdot \mathbf{n}^f = -\underline{\sigma}^s \cdot \mathbf{n}^s \quad \text{in } \Gamma^i \quad (2.67)$$

The position of  $\Gamma^i$  is varying with time and is only defined through the interaction of both fields.

### 2.3.2 Three-field formulation of the FSI system

The interface conditions have to be applied along a moving surface, which is defined by the Lagrangian description of the structure surface. For the fluid field, ALE methods were established that allow the fluid mesh to be attached to fluid-structure interface at all times. In a pure Eulerian description for the fixed fluid grid, there is no moving fluid surface mesh to which one



**Figure 2.2:** FSI three-field setup: fluid field  $\Omega^f$ , interface  $\Gamma^i$  and structural field  $\Omega^s$  along with respective domain normals and variables.

could couple the moving structural surface. Instead, the structural surface intersects the fluid element at arbitrary positions. Consequently, to re-use established surface-surface coupling approaches available for ALE methods — weak surface-surface coupling or even node-matching —, an extension of the fluid-structure system is necessary.

For maximum flexibility with respect to the choice of approximation functions, mesh size and solution techniques, a three-field setup is proposed. In the three-field setup, fluid and structural fields and an independent interface field are treated separately. Such a setup — along with the respective variables living on these fields or interfaces — is shown in Figure 2.2. In a continuous setting, the interface  $\Gamma^i$  and the FSI surfaces of fluid and structure, namely  $\Gamma^{f,FSI}$  and  $\Gamma^{s,FSI}$ , respectively, are identical. However, they may be discretized independently from each other. The most general case occurs, if neither fluid nor structural discretization match the interface mesh. Then three non-fitting meshes need to be coupled together. If, however, two surfaces share identical discretizations, the interface mesh can be removed from a monolithic FSI system and no extra interface displacement unknowns are present in the linear system.

The interface deformation is described by the Lagrangian formulation using the interface displacement  $d^i(X, t)$ . Then, for the three-field setup, the interface condition Eq. (2.66) is split into two conditions as

$$\mathbf{u} = \frac{\partial \mathbf{d}^i}{\partial t} \quad \text{in } \Gamma^i \quad (2.68)$$

$$\mathbf{d}^i = \mathbf{d}^s \quad \text{in } \Gamma^i \quad (2.69)$$

In other words, if the velocity of both fields is constrained independently to be the same as the interface velocity, the original matching condition (Eq. (2.66)) is still fulfilled. Let  $\lambda$  and  $\mu$  be two traction fields on the interface  $\Gamma^i$  such that

$$\underline{\sigma}^f \cdot \mathbf{n}^f = \lambda \quad \text{in } \Gamma^i \quad (2.70)$$

$$\underline{\sigma}^s \cdot \mathbf{n}^s = \mu \quad \text{in } \Gamma^i \quad (2.71)$$

Then the interface traction balance (Eq. (2.67)) can be written in an alternative way as

$$\lambda = -\mu \quad \text{in } \Gamma^i \quad (2.72)$$

The interface is discretized using linear or quadratic FE approximations

$$\mathbf{d}^{i,h} = \sum_I N_I^i \mathbf{d}_I^i \quad (2.73)$$

The interface mesh consists therefore of triangular or quadrilateral elements. If the interface displacement or the corresponding velocity are part of the solution, the appropriate test function  $\delta \mathbf{d}^{i,h}$  is discretized as

$$\delta \mathbf{d}^{i,h} = \sum_I N_I^i \delta \mathbf{d}_I^i \quad (2.74)$$

This setup has several features that are elucidated and exploited throughout this thesis. First of all, it is possible to treat the coupling of the single fields to the interface independently from each other. This prevents to mix structural and fluid concerns in the mathematical derivation and the implementation process, if no field gets information from beyond the interface. Finally, the proposed three-field setup allows a straightforward re-use of partitioned FSI schemes including their implementation as described in Chapter 6.

Note that similar concepts for coupling two fields using an intermediate reference surface have been presented in [Park et al., 2001] and references therein. The basic motivation of the ‘localized Lagrange multiplier method’, namely the separation of coupling concerns, is identical to the present thesis. Also, the coupling between structure and interface follows the established rules of surface-surface coupling and is reviewed in Section 6.1. However, the weak coupling of the fixed fluid grid to the interface presented in the following chapters is different and is one of the major contributions of this thesis.



# CHAPTER 3

## Introduction to the XFEM and weak constraints

The following chapter demonstrates key steps that are necessary, when using the XFEM and Lagrange multiplier techniques to treat implicit interfaces on fixed meshes. The principle ideas are demonstrated for the heat conduction problem, which is characterized by the Poisson equation. This scalar problem can also be used to become familiar with XFEM in general, however, only topics relevant for the final goal of implementing a fixed grid FSI scheme are discussed - for further information the reader is referred to the given literature. For this chapter, the interface position is assumed to be spatially fixed.

### 3.1 Governing equations

The steady state heat conduction equation in  $\Omega^+$  is given as

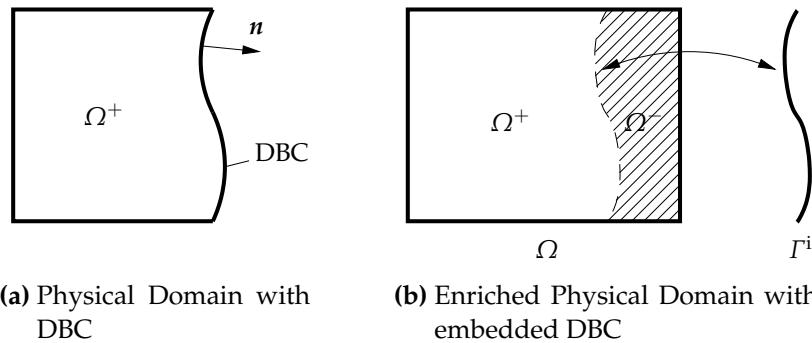
$$\nabla \cdot \mathbf{q} = \hat{f} \quad \text{in } \Omega^+ \tag{3.1}$$

The heat flux  $\mathbf{q}$  is defined as

$$\mathbf{q} = -\kappa^{\text{diff}} \nabla c \quad \text{in } \Omega^+ \tag{3.2}$$

where  $c$  denotes the temperature field and  $\kappa^{\text{diff}}$  denotes the constant diffusion or heat conduction coefficient. A temperature-independent heat source term is denoted by  $\hat{f}$ .

Assume a Dirichlet boundary condition (DBC) on parts of a boundary of a physical domain  $\Omega^+$  as depicted in Figure 3.1a. This setup is reformulated as an embedded Dirichlet problem, where an internal interface  $\Gamma^i$  divides a larger domain  $\Omega$  into a physical and a fictitious domain named  $\Omega^+$  and



**Figure 3.1:** The physical domain with DBC is shown in Fig. (a) and Fig. (b) illustrates the transition from an explicit to an embedded boundary.

$\Omega^-$ , respectively. The part of the boundary with the DBC becomes now an internal interface  $\Gamma^i$  to the domain  $\Omega$  as shown in Figure 3.1b. The location of the interface may be given, for example, by an analytic function, the zero-contour of a level-set function or by a boundary mesh. This setup is also referred to as one-sided discontinuity since there is only a physical field on one side of the discontinuity. For easier writing, two additional names for the boundary  $\Gamma^i$  are defined, namely  $\Gamma^+$  and  $\Gamma^-$ , depending on whether functions are evaluated approaching  $\Gamma^i$  from  $\Omega^+$  or  $\Omega^-$ , respectively. Thus, writing  $c$  evaluated at  $\Gamma^+$  is equivalent to writing  $c^+$  evaluated at  $\Gamma^i$ .

Using the implicit description of the domain surface, the temperature field can be seen as a discontinuous field with a jump in  $c$  between the physical values  $c^+$  and the void ( $c^-$ )

$$[c] = c^+ - \underbrace{c^-}_{=0} \quad \text{in } \Gamma^i \quad (3.3)$$

In an one-sided problem, the value of  $c^-$  is completely irrelevant for the solution in  $\Omega^+$ . Then the jump height equals the value of  $c$  at  $\Gamma^+$  and one can pose the kinematic constraints for  $c^+$  along  $\Gamma^i$  as

$$c^+ - \hat{c}^i = 0 \quad \text{in } \Gamma^i \quad (3.4)$$

where the interface temperature  $\hat{c}^i$  is given. Likewise, one can identify a jump discontinuity in the flux field  $q$

$$[q \cdot n] = q^+ \cdot n - \underbrace{q^- \cdot n}_{=0} \quad \text{in } \Gamma^i \quad (3.5)$$

The complete strong form including Neumann and Dirichlet conditions that are applied at remaining boundaries and under consideration of the ma-

terial law in Eq. (3.2) is given as

$$\nabla \cdot \mathbf{q} - \hat{f} = 0 \quad \text{in } \Omega^+ \quad (3.6)$$

$$c - \hat{c}_D = 0 \quad \text{in } \Gamma_D \quad (3.7)$$

$$\mathbf{q} \cdot \mathbf{n} - \hat{h}_N = 0 \quad \text{in } \Gamma_N \quad (3.8)$$

$$c - \hat{c}^i = 0 \quad \text{in } \Gamma^+ \quad (3.9)$$

The prescribed heat source along the Neumann boundary is denoted as  $\hat{h}_N$  and the prescribed Dirichlet value is denoted as  $\hat{c}_D$ .

The weak form, first without consideration of the interface constraint in Eq. (3.9), is constructed by multiplying the strong form with a weighting function  $\delta c$  such that

$$0 = (\delta c, \nabla \cdot \mathbf{q} - \hat{f})_{\Omega^+} + (\delta c, \mathbf{q} \cdot \mathbf{n} - \hat{h}_N)_{\Gamma^+} \quad (3.10)$$

Integration by parts of the first term gives

$$0 = -(\nabla \delta c, \mathbf{q})_{\Omega^+} - (\delta c, \hat{h}_N)_{\Gamma_N} - (\delta c, \hat{f})_{\Omega^+} \quad (3.11)$$

Eq. (3.11) with  $c$  as the unknown field is the basis for the subsequent finite element formulation. The remaining boundaries of  $\Omega^+$  coincide with the surface of the finite element mesh and are treated as in the standard FEM.

## 3.2 XFEM formulation

Describing domain boundaries independent of finite element boundaries has been proposed in methods like the XFEM [Belytschko and Black, 1999; Moës et al., 1999], the generalized finite element method (GFEM) [Strouboulis et al., 2000] and the partition of unity finite element method (PUFEM) [Babuška and Melenk, 1997]. The basic idea is to allow the user to add known solution behavior to the discrete approximation functions. If a principle solution is known, for example the singular stress field near a crack tip or a discontinuous stress field due to changing material properties, the local approximation can use the principle solution instead or in addition to standard polynomials. As a result, the coarse mesh accuracy as well as the convergence are improved, as shown in the given literature.

Applied to the temperature field, an enriched temperature approximation is defined as

$$\begin{aligned} c^h(x) = & \sum_I N_I(x) c_I^0 + \sum_J N_J(x) \psi^1(x) c_J^1 \\ & + \sum_K N_K(x) \psi^2(x) c_K^2 \\ & + \dots \end{aligned} \quad (3.12)$$

The standard nodal DOFs at node  $I$  are represented by  $c_I^0$ , while additional DOFs are denoted as  $c_I^1$  and  $c_I^2$ . The additional DOFs are multiplied by properly chosen enrichment functions  $\psi^1(x)$  and  $\psi^2(x)$ , respectively. Any number of overlapping enrichments can be applied to certain regions of the computational domain. The standard FE shape functions are denoted as  $N_I(x)$ . In the present work, shape functions are either polynomial nodal shape functions, which are  $C^0$  continuous over inter-element boundaries or polynomial element shape functions, which are discontinuous over inter-element boundaries.

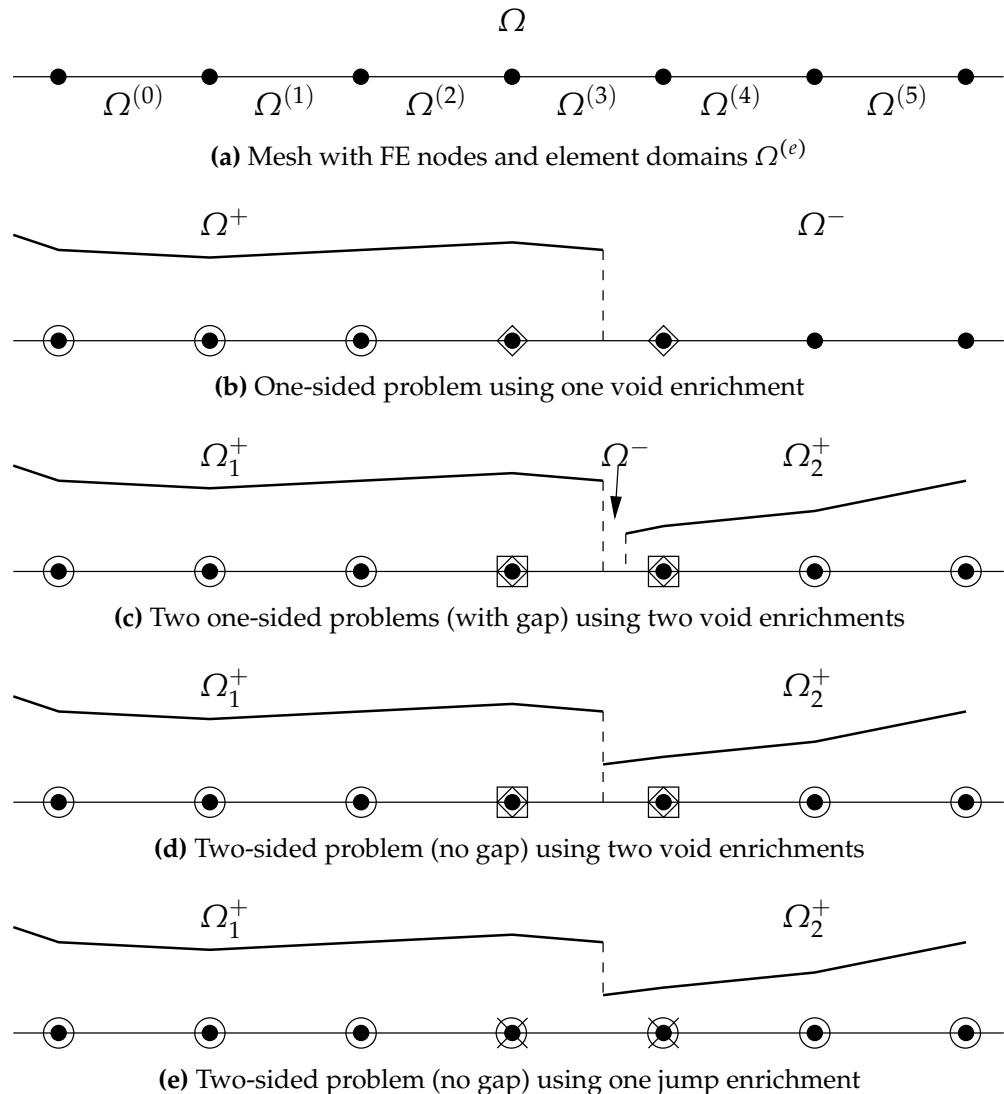
Choosing appropriate enrichments is one of the key decisions the user of XFEM has to make. Figure 3.2 shows the different enrichment scenarios used in this thesis. The different ways of enriching the solution are all applied to the FE mesh shown in Figure 3.2a.

For the given one-sided problem, the material surface of  $\Omega^+$  could be seen as a step-like change of the temperature field as introduced in Eq. (3.3). Using an appropriate step-like enrichment will allow the FE approximation to represent such steps on an arbitrarily mesh without aligning the element edges with the material surface. The appropriate enrichment is the step function  $\bar{\psi}(x)$  as discussed in [Daux et al., 2000; Sukumar et al., 2001]

$$\bar{\psi}(x) = \begin{cases} 1 & \text{in } \Omega^+ \\ 0 & \text{in } \Omega^- \end{cases} \quad (3.13)$$

The enrichment function  $\bar{\psi}$  will subsequently be called void enrichment function. For non-moving interfaces, as assumed in this chapter, the enrichment function  $\bar{\psi}$  is independent of time. Figure 3.2b shows how a single enrichment and standard DOFs are applied to model a physical domain  $\Omega^+$  that ends in the middle of element (3). Within  $\Omega^-$ ,  $c$  will remain zero at all times and no unknowns beyond element (3) are required. The resulting discrete equations are identical to the equations obtained from approaches, where integration is performed only over the physical part of an element [see e.g. Hansbo and Hansbo, 2004; Tezduyar, 2006], because omitting the integration over the fictitious part of the element is the same as integrating over  $\bar{\psi}(x)N(x)$  in  $\Omega^-$ .

Two void enrichments are necessary to model two independent physical domains  $\Omega_1^+$  and  $\Omega_2^+$  shown in Figure 3.2c. The case without a gap between  $\Omega_1^+$  and  $\Omega_2^+$  is usually called a two-sided problem. Two-sided problems with a strong discontinuity can be represented either by two void enrichments as shown in Figure 3.2d or by one jump enrichment as shown Figure 3.2e. In the latter case, two unknowns are added to the continuous approximation field. The jump enrichment function for two-sided problems without gap is given



**Figure 3.2:** Enrichment strategies for the one-dimensional Poisson equation: Standard DOFs are denoted by symbol  $\circ$ , void enrichment DOFs by  $\diamond$  and  $\square$  and jump enrichment DOFs by symbol  $\times$ .

as

$$\check{\psi}(x) = \begin{cases} +1 & \text{in } \Omega^+ \\ -1 & \text{in } \Omega^- \end{cases} \quad (3.14)$$

Figure 3.2d and Figure 3.2e highlight the equivalence between two void enrichments without gap and one jump enrichment. The equivalence is also visible in the identical number of unknowns required for these two cases. Analogue to one-side problems, two-sided problems can also be modeled with methods as described in [Hansbo and Hansbo, 2004].

In XFEM terminology, the presented void and the jump enrichments lead to strong discontinuities, where the primary variable is discontinuous. Weak discontinuities, where only the derivatives of  $c$  have a step like behavior, are not required for this thesis and will not be discussed any further.

For compact notation and also implementation, Eq. (3.12) is generalized by defining an enrichment function  $\tilde{\psi}(x)$  for all standard DOFs as

$$\tilde{\psi}(x) = 1 \quad (3.15)$$

Then, even standard degrees of freedom have an enrichment function attached, such that one can write for all degrees of freedom

$$c^h(x) = \sum_I N_I(x) \psi^0(x) c_I^0(t) + \sum_J N_J(x) \psi^1(x) c_K^1(t) + \dots \quad (3.16)$$

$$= \sum_L N_L(x) \phi_L(x) c_L(t) \quad (3.17)$$

$$= \sum_L \phi_L(x) c_L(t) \quad (3.18)$$

Here,  $\phi_L(x)$  denotes a general approximation function that may include discontinuous solution components. The corresponding unknown is denoted by  $c_L(t)$  and  $n_L$  is the number of unknowns or parameters that influence the solution at  $x$ . Similarly to meshfree methods,  $n_L$  varies spatially. If the approximation is used in the FE context, the approximation function can be written as function of the element coordinates  $\xi$  as

$$c_e^h(\xi, t) = \sum_I N_I(\xi) \psi^0(\xi, t) c_I^0(t) + \sum_J N_J(\xi) \psi^1(\xi, t) c_K^1(t) + \dots \quad (3.19)$$

$$= \sum_L N_L(\xi) \phi_L(\xi, t) c_L(t) \quad (3.20)$$

$$= \sum_L \phi_L(\xi, t) c_L(t) \quad (3.21)$$

If the enrichment function can be expressed as a function of  $\xi$  – e.g. if level-set functions are used – then  $\psi(\xi)$  can be used directly. For the FSI interface

discussed in later chapters, the enrichment function is a global function depending on  $\boldsymbol{x}$ . For elementwise integration,  $\boldsymbol{x}$  is computed from element coordinates such that  $\psi(\boldsymbol{x}(\xi))$ . Note that no superscript is introduced to formally differentiate  $\phi_L(\boldsymbol{x})$  from  $\phi_L(\xi)$  to avoid cluttering the notation.

Derivatives of  $c^h$  are calculated by using the chain rule

$$\begin{aligned}\frac{\partial c_e^h(\boldsymbol{x})}{\partial \boldsymbol{x}} &= \sum_L \frac{\partial \phi_L(\boldsymbol{x})}{\partial \boldsymbol{x}} c_L(t) \\ &= \sum_L \frac{\partial N_L(\boldsymbol{x}) \psi_L(\boldsymbol{x})}{\partial \boldsymbol{x}} c_L(t) \\ &= \sum_L \left( \frac{\partial N_L(\boldsymbol{x})}{\partial \boldsymbol{x}} \psi_L(\boldsymbol{x}) + N_L(\boldsymbol{x}) \frac{\partial \psi_L(\boldsymbol{x})}{\partial \boldsymbol{x}} \right) c_L(t)\end{aligned}\quad (3.22)$$

For the fixed-grid FSI scheme presented in the next chapters, only one or multiple void enrichments are used to model the fluid surface independently of finite elements surfaces. The derivative of the void enrichment function in  $\Omega$  is zero, since  $\bar{\psi}$  is constant everywhere in  $\Omega$ . Hence, Eq. (3.22) simplifies to

$$\frac{\partial c_e^h(\boldsymbol{x})}{\partial \boldsymbol{x}} = \sum_L \frac{\partial N_L(\boldsymbol{x})}{\partial \boldsymbol{x}} \psi_L(\boldsymbol{x}) c_L(t) \quad (3.23)$$

The shape function derivative with respect to  $\boldsymbol{x}$  is computed as usual

$$\frac{\partial N_L(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial N_L(\xi)}{\partial \xi} \underbrace{\frac{\partial \xi}{\partial \boldsymbol{x}}}_{J_{x\xi}^{-1}} \quad (3.24)$$

with  $J_{x\xi}$  being the Jacobian between the coordinate systems  $\boldsymbol{x}$  and  $\xi$ .

Along  $\Gamma^i$ , both temperature and flux field are discontinuous and enriched with  $\bar{\psi}(\boldsymbol{x})$ . The complete discretization for trial and test functions is given as

$$c^h(\boldsymbol{x}) = \sum_L \phi_L^c(\boldsymbol{x}) c_L \quad (3.25)$$

$$\delta c^h(\boldsymbol{x}) = \sum_L \phi_L^c(\boldsymbol{x}) \delta c_L \quad (3.26)$$

This approach is still a Bubnov-Galerkin method, since test and trial functions are discretized with the same approximation.

As mentioned before, Figure 3.2 illustrates how at different positions in space a varying number of discrete unknowns is used to approximate the field. For instance, in Figure 3.2c, element (1) has two unknowns, element (2) has three unknowns and element (3) four unknowns. Consequently, if elementwise numerical integration is performed, the element matrices are two-by-two, three-by-three and four-by-four matrices, respectively. For instance,

the discrete temperature in element (2) can be written (and implemented) as

$$c_e^h(\xi) = \boldsymbol{\phi} \cdot \mathbf{c} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} N_1 \tilde{\psi}^0 \\ N_2 \tilde{\psi}^1 \\ N_3 \tilde{\psi}^2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad (3.27)$$

The unknown  $c_1$  represents a standard unknown and  $c_2$  and  $c_3$  represent additional unknowns from two distinct void enrichments belonging to two distinct surfaces. The discrete derivative with respect to  $x$  is therefore

$$\frac{\partial c_e^h(\xi)}{\partial x} = \boldsymbol{\phi}_{,x} \cdot \mathbf{c} = \begin{bmatrix} \frac{\partial \phi_1}{\partial x} \\ \frac{\partial \phi_2}{\partial x} \\ \frac{\partial \phi_3}{\partial x} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial \phi_1}{\partial x} \tilde{\psi}^0 \\ \frac{\partial \phi_2}{\partial x} \tilde{\psi}^1 \\ \frac{\partial \phi_3}{\partial x} \tilde{\psi}^2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial \phi_1}{\partial x} \\ \frac{\partial \phi_2}{\partial x} \\ \frac{\partial \phi_3}{\partial x} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad (3.28)$$

After construction of the discrete approximation  $\boldsymbol{\phi}$  and its derivatives, the FE algorithms can use the new approximation instead of standard FE shape functions. Instead of looping over a number of nodes, the loops are over a number of parameters that influence a certain integration point. For instance, the entry into the element stiffness matrix resulting from  $(\delta c_{,x}, \kappa^{\text{diff}} c_{,x})_{\Omega_e}$  can be written as

$$(\delta c_{,x}^h, \kappa^{\text{diff}} c_{,x}^h)_{\Omega_e} = \delta \mathbf{c}_e \int_{\Omega_e} \boldsymbol{\phi}_{,x} \kappa^{\text{diff}} \boldsymbol{\phi}_{,x} d\mathbf{x} \mathbf{c}_e \quad (3.29)$$

In summary, the XFEM extends the approximation function locally, e.g. near an interface or surface. The strong discontinuities model a sharp surface of the physical domain. For the remainder of this chapter, one-sided problems using one void enrichment are exclusively used. The next section illustrates how boundary conditions are applied along such implicit boundaries.

### 3.3 Enforcement of interface constraints

The weak enforcement of constraints or, as a subset, of Dirichlet conditions along boundaries has been an active research topic within the last decade. Beyond the use in fixed-grid methods, weak constraints are also present when applying conditions in meshfree methods that lack the finite element interpolation property, or for general mesh tying.

In the following, four principle approaches for applying constraints to the continuous Galerkin (CG) method ( $C^0$  continuous temperature approximation) are presented and a brief discussion of their properties is given. In particular, the approaches are analyzed for the applicability to the three-dimensional NS equation. The first and the fourth approach are later implemented for the incompressible NS equations. Recent approaches based on residual

free bubbles as introduced in [Dolbow and Franca, 2008; Mourad et al., 2007] and the approach proposed in [Codina and Baiges, 2009] have not been re-reviewed and considered in detail, but should be mentioned here for completeness.

The strong form of governing equation including the interface condition at  $\Gamma^i$  reads as

$$-\nabla \cdot (\kappa^{\text{diff}} \nabla c) = 0 \quad \text{in } \Omega \quad (3.30)$$

$$c^+ - \hat{c}^i = 0 \quad \text{in } \Gamma^+ \quad (3.31)$$

The volume heat source as well as Dirichlet and Neumann conditions away from  $\Gamma^i$  are ignored in the following to concentrate on the weak Dirichlet condition treatment.

### 3.3.1 ‘Classical’ Lagrange multiplier

A standard method for weak imposition of constraints is the use of Lagrange multipliers that are described in many text books on the FE, e.g. in [Hughes, 1987, chap 4.2.1]. The corresponding weighted residual equation is given as

$$0 = (\nabla \delta c, \kappa^{\text{diff}} \nabla c)_{\Omega^+} - (\delta c, \lambda)_{\Gamma^+} \quad (3.32a)$$

$$0 = -(\delta \lambda, c - \hat{c})_{\Gamma^+} \quad (3.32b)$$

This weak form is the basis for the FE discretization. After integration by parts of the first integral in the first equation, the weighted residual equations are obtained

$$0 = (\delta c, -\nabla \cdot (\kappa^{\text{diff}} \nabla c))_{\Omega^+} - (\delta c, \lambda - \kappa^{\text{diff}} \nabla c \cdot \mathbf{n})_{\Gamma^+} \quad (3.33a)$$

$$0 = -(\delta \lambda, c - \hat{c})_{\Gamma^+} \quad (3.33b)$$

from which the Euler-Lagrange equations are obtained

$$-\nabla \cdot (\kappa^{\text{diff}} \nabla c) = 0 \quad \text{in } \Omega^+ \quad (3.34a)$$

$$c - \hat{c} = 0 \quad \text{in } \Gamma^+ \quad (3.34b)$$

$$\lambda - \kappa^{\text{diff}} \nabla c \cdot \mathbf{n} = 0 \quad \text{in } \Gamma^+ \quad (3.34c)$$

The weighted residual form is related to the following functional  $\Pi_{\lambda}^{\text{diff}}$ :

$$\Pi_{\lambda}^{\text{diff}} = (\nabla c, \frac{1}{2} \kappa^{\text{diff}} \nabla c)_{\Omega^+} - (\lambda, c - \hat{c})_{\Gamma^+} \quad (3.35)$$

The variation of  $\Pi_{\lambda}^{\text{diff}}$  with respect to the two primary unknowns  $c$  and  $\lambda$  leads to Eq. (3.32a) and Eq. (3.32b). The resulting discrete system is of the form

$$\begin{bmatrix} K_{cc} & M_{c\lambda} \\ M_{\lambda c} & \mathbf{0} \end{bmatrix} \begin{bmatrix} c \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ f_{\lambda}^{\hat{c}} \end{bmatrix} \quad (3.36)$$

where the saddle point structure of the problem is clearly visible.

When above approach is applied to implicit interfaces crossing elements, finding inf-sup stable pairs of  $c$  and  $\lambda$  is the first challenge to be approached. Based on such pairs, an appropriate interface mesh has to be generated. This is a challenging task already for two-dimensional problems and can still be considered an open problem for three-dimensional implementations, see for example the discussions in [Ji and Dolbow, 2004; Moës et al., 2006]. In contrast, arbitrary choices of Lagrange multipliers can be employed when additional stabilization is introduced, such that the inf-sup condition is circumvented. This is the approach first advocated in [Barbosa and Hughes, 1992]. In that case, proper stabilization has to be defined depending on the physical problem. Beyond the stability issue, the Lagrange multiplier requires a surface mesh in addition to the background mesh. Generating such interface meshes is a geometrically complex task for three-dimensional implementations. Hence, methods that avoid the generation of an interface mesh solely for the Lagrange multipliers promise to be much easier to implement and such methods are considered in the following.

An approach with properties that are between the classical Lagrange multiplier and the following methods is the distributed Lagrange multiplier approach proposed in [Zilian and Legay, 2008]. Here, the Lagrange multiplier unknowns are placed directly onto the fixed background grid. Applied to the present Poisson problem, the ‘distributed’ Lagrange multiplier is discretized with domain shape functions and the unknowns of  $\lambda$  reside on the nodes of the background mesh. However, the Lagrange multiplier is still only present in boundary integrals. The additional number of Lagrange multiplier unknowns without additional equations are likely the reason, why the distributed Lagrange multiplier method requires a stabilizing term and a user-defined stabilization parameter.

### 3.3.2 Nitsche’s method

Nitsche’s method [Nitsche, 1971] has recently gained great attention in the context of implicit interfaces modeled by the XFEM [Dolbow and Harari, 2009], but also as general technique for the imposition of constraints along non-matching surface grids [Bazilevs and Hughes, 2008; Becker et al., 2003; Hansbo and Hansbo, 2002; Hansbo et al., 2004; Sanders et al., 2009; Stenberg, 1995] or for meshfree techniques [Fernández-Méndez and Huerta, 2004]. In the context of the incompressible NS equations, it has been used for coupling non-matching grids [Bazilevs and Hughes, 2008; Hansbo et al., 2004]. It has been shown in [Stenberg, 1995] that Nitsche’s method [Nitsche, 1971] could be derived as a special case of the Lagrange multiplier stabilization as mentioned above.

Applying Nitsche's method to the heat conduction problem, the weak form is given as

$$0 = (\nabla \delta c, \kappa^{\text{diff}} \nabla c)_{\Omega^+} - (\delta c, \kappa^{\text{diff}} \nabla c \cdot \mathbf{n})_{\Gamma^+} - (\kappa^{\text{diff}} \nabla \delta c \cdot \mathbf{n}, c - \hat{c})_{\Gamma^+} - (\alpha \delta c, c - \hat{c})_{\Gamma^+} \quad (3.37)$$

where  $\alpha$  is some stabilization parameter to be defined. After integration by parts the weighted residual form is

$$0 = -(\delta c, \nabla \cdot (\kappa^{\text{diff}} \nabla c))_{\Omega^+} - (\kappa^{\text{diff}} \nabla \delta c \cdot \mathbf{n}, c - \hat{c})_{\Gamma^+} - (\alpha \delta c, c - \hat{c})_{\Gamma^+} \quad (3.38)$$

from which the Euler-Lagrange equations are recovered

$$-\nabla \cdot (\kappa^{\text{diff}} \nabla c) = 0 \quad \text{in } \Omega^+ \quad (3.39)$$

$$c - \hat{c} = 0 \quad \text{in } \Gamma^+ \quad (3.40)$$

The method is consistent and leads to symmetric discrete systems. The most appealing feature is that Nitsche's methods does not require the additional Lagrange multiplier field as the classical Lagrange multiplier technique. Hence, Nitsche's method requires the least implementation effort compared to all other methods reviewed in this section, since no boundary meshes and no additional variables are introduced.

On the other hand, the method inevitably leads to an unstable problem such that a least-square stabilization using  $\alpha$  or similar measures are required. However, the definition of the stabilization parameters is still a research topic, which especially for incompressible and convective flow requires further research.

### 3.3.3 Generalized Hellinger-Reissner principle

The Hellinger-Reissner (HR) principle [Hellinger, 1914; Reissner, 1950] is a hybrid or two-field formulation that introduces the additional flux  $\bar{q}$  as an independent and dual field to the primary temperature field  $c$ . The bar over the flux indicates that the flux is now an independent field as opposed to the flux field  $\mathbf{q}(c)$  that is a function of the temperature. The HR principle applied to the heat-conduction problem can be written as follows

$$0 = (\nabla \delta c, \bar{q})_{\Omega^+} - (\delta c, \bar{q} \cdot \mathbf{n})_{\Gamma^+} \quad (3.41a)$$

$$0 = -(\delta \bar{q}, \frac{1}{\kappa^{\text{diff}}} \bar{q} - \nabla c)_{\Omega^+} - (\delta \bar{q} \cdot \mathbf{n}, c - \hat{c})_{\Gamma^+} \quad (3.41b)$$

The above form is the generalized HR formulation, where Dirichlet conditions are enforced weakly using the additional flux variable. Integration by

parts of the first integral in Eq. (3.41a) gives

$$0 = -(\delta c, \nabla \cdot \bar{q})_{\Omega^+} \quad (3.42a)$$

$$0 = -(\delta \bar{q}, \frac{\bar{q}}{\kappa^{\text{diff}}} - \nabla c)_{\Omega^+} - (\delta \bar{q} \cdot \mathbf{n}, c - \hat{c})_{\Gamma^+} \quad (3.42b)$$

from which the Euler-Lagrange equations emerge as

$$-\nabla \cdot \bar{q} = 0 \quad \text{in } \Omega^+ \quad (3.43)$$

$$\bar{q} - \kappa^{\text{diff}} \nabla c = 0 \quad \text{in } \Omega^+ \quad (3.44)$$

$$c - \hat{c} = 0 \quad \text{in } \Gamma^+ \quad (3.45)$$

Flux and temperature are discretized independently such that Eq. (3.44) has to be weakly enforced as shown in Eq. (3.41b). The discrete  $\bar{q}^h$  is approximated elementwise discontinuous (see also Eq. (3.53)) and can be condensed at the element level. The linear system before condensation is of the form

$$\mathbf{A}_e \begin{bmatrix} 0 & \mathbf{K}_{c\bar{q}}^e + \mathbf{G}_{c\bar{q}}^e \\ \mathbf{K}_{\bar{q}c}^e + \mathbf{G}_{\bar{q}c}^e & \mathbf{K}_{\bar{q}\bar{q}}^e \end{bmatrix} \mathbf{A}_e \begin{bmatrix} c_e \\ \bar{q}_e \end{bmatrix} = \mathbf{A}_e \begin{bmatrix} 0 \\ f_{\bar{q}}^{*,e} \end{bmatrix} \quad (3.46)$$

where  $\mathbf{G}_{c\bar{q}}^e$  and  $\mathbf{G}_{\bar{q}c}^e$  represent the boundary integrals. Before the assembly of the global linear system, the modified element stiffness matrix  $\mathbf{K}_{cc}^{*,e}$  and force vector  $\mathbf{f}_c^{*,e}$  are computed by

$$\mathbf{K}_{cc}^{*,e} = -(\mathbf{K}_{c\bar{q}}^e + \mathbf{G}_{c\bar{q}}^e)(\mathbf{K}_{\bar{q}\bar{q}}^e)^{-1}(\mathbf{K}_{\bar{q}c}^e + \mathbf{G}_{\bar{q}c}^e) \quad (3.47)$$

$$\mathbf{f}_c^{*,e} = -(\mathbf{K}_{c\bar{q}}^e + \mathbf{G}_{c\bar{q}}^e)(\mathbf{K}_{\bar{q}\bar{q}}^e)^{-1} f_{\bar{q}}^{*,e} \quad (3.48)$$

such that the condensed discrete system is

$$\mathbf{A}_e [\mathbf{K}_{cc}^{*,e}] \mathbf{A}_e [\mathbf{c}_e] = \mathbf{A}_e [\mathbf{f}_c^{*,e}] \quad (3.49)$$

One of the primary applications of the HR method is the construction of low-order elements without locking. For an overview on such multi-variable formulation for solid mechanics see e.g. [Pian and Wu, 2006, chap 1.3]. The useful feature for the problem at hand is that after condensation, interface conditions are treated weakly without additional Lagrange multiplier unknowns. An application of the HR method to weak interface conditions in the context of XFEM has been presented recently in [Zilian and Fries, 2009]. The drawback is that the element stiffness matrix is modified for all or at least for all intersected elements. That would eventually require a completely new element formulation for the intended application on the stabilized NS equations. Hence, a modified version of the HR method is proposed in the following.

### 3.3.4 Flux-based Lagrange multiplier

If the first integral in Eq. (3.41a) is replaced with its original form in Eq. (3.11), which is solely based on the temperature and its test function, then the following weak form is obtained

$$0 = (\nabla \delta c, \kappa^{\text{diff}} \nabla c)_{\Omega^+} - (\delta c, \bar{q} \cdot \mathbf{n})_{\Gamma^+} \quad (3.50\text{a})$$

$$0 = -(\delta \bar{q}, \frac{1}{\kappa^{\text{diff}}} \bar{q} - \nabla c)_{\Omega^+} - (\delta \bar{q} \cdot \mathbf{n}, c - \hat{c})_{\Gamma^+} \quad (3.50\text{b})$$

Again, temperature  $c$  and flux  $\bar{q}$  are independent variables and the temperature gradient based on each primary variable is coupled weakly. However, the flux enters Eq. (3.50a) only via the boundary integral, where Dirichlet conditions are enforced weakly.

The usual integration by parts of the first integral in Eq. (3.50a) gives

$$0 = -(\delta c, \nabla \cdot (\kappa^{\text{diff}} \nabla c))_{\Omega^+} - (\delta c, \bar{q} \cdot \mathbf{n} - \kappa^{\text{diff}} \nabla c \cdot \mathbf{n})_{\Gamma^+} \quad (3.51\text{a})$$

$$0 = -(\delta \bar{q}, \frac{\bar{q}}{\kappa^{\text{diff}}} - \nabla c)_{\Omega^+} - (\delta \bar{q} \cdot \mathbf{n}, c - \hat{c})_{\Gamma^+} \quad (3.51\text{b})$$

from which the Euler-Lagrange equations are recovered

$$-\nabla \cdot (\kappa^{\text{diff}} \nabla c) = 0 \quad \text{in } \Omega^+ \quad (3.52\text{a})$$

$$\bar{q} - \kappa^{\text{diff}} \nabla c = 0 \quad \text{in } \Omega^+ \quad (3.52\text{b})$$

$$c - \hat{c} = 0 \quad \text{in } \Gamma^+ \quad (3.52\text{c})$$

$$\bar{q} \cdot \mathbf{n} - \kappa^{\text{diff}} \nabla c \cdot \mathbf{n} = 0 \quad \text{in } \Gamma^+ \quad (3.52\text{d})$$

The temperature shape functions  $N_I^c(x)$  are chosen as piecewise continuous polynomials, that are  $C^0$ -continuous at inter-element boundaries. The shape functions for the flux unknowns  $N_K^{\bar{q}}(x)$  shall also be polynomial functions inside each element, however, they shall be  $C^{-1}$  discontinuous at inter-element boundaries. For the XFEM approach, both temperature and flux field are discontinuous across the interface and appropriately enriched along  $\Gamma^i$ . The complete discretization for trial and test functions is

$$c^h(x) = \sum_I \phi_I^c(x) c_I \quad (3.53\text{a})$$

$$\delta c^h(x) = \sum_I \phi_I^c(x) \delta c_I \quad (3.53\text{b})$$

$$\bar{q}^h(x) = \sum_K \phi_K^{\bar{q}}(x) \bar{q}_K \quad (3.53\text{c})$$

$$\delta \bar{q}^h(x) = \sum_K \phi_K^{\bar{q}}(x) \delta \bar{q}_K \quad (3.53\text{d})$$

with  $\phi_K^{\bar{q}}(x) = N_K^{\bar{q}}(x) \bar{\psi}(x)$ .

For the examples given at the end of this chapter, temperature and flux have the same order for their shape functions. Adopting the notation from [Gresho and Sani, 2000], the approximation for temperature and flux could be termed as  $Q_1 Q_{-1}$  for trilinear,  $Q_2^{(20)} Q_{-2}^{(20)}$  for quadratic serendipity and  $Q_2 Q_{-2}$  for full triquadratic shape functions. Likewise, the approximation for linear and quadratic tetrahedrons can be defined as  $P_1 P_{-1}$  and  $P_2 P_{-2}$ , respectively. Other element shapes like pyramids and wedges should approximate the flux in an analogous manner. Note that a mathematical proof for stability is still missing. Within the present work, only numerical experience with the new approach and the original HR method gives confidence that stabilization is not necessary.

The linear system corresponding to Eq. (3.50) is of the form

$$\mathbf{A}_e \begin{bmatrix} \mathbf{K}_{cc}^e & \mathbf{G}_{c\bar{q}}^e \\ \mathbf{K}_{\bar{q}c}^e + \mathbf{G}_{\bar{q}c}^e & \mathbf{K}_{\bar{q}\bar{q}}^e \end{bmatrix} \mathbf{A}_e \begin{bmatrix} \mathbf{c}_e \\ \bar{\mathbf{q}}_e \end{bmatrix} = \mathbf{A}_e \begin{bmatrix} \mathbf{0} \\ \mathbf{f}_{\bar{q}}^{\hat{c},e} \end{bmatrix} \quad (3.54)$$

Matrix  $\mathbf{K}_{cc}^e$  is the original stiffness matrix for the unconstrained system, which is also present in Section 3.3.1 and Section 3.3.2. The upper right-corner contains only boundary terms ( $\mathbf{G}_{c\bar{q}}^e$ ). The remaining terms are the same as in the HR-formulation.

This formulation decouples the flux equation from the primary weak heat conduction equation, if no boundary crosses the element. Hence, the standard (or one-field) FE formulation is valid for all non-intersected elements; at the same time, all intersected elements have flux unknowns defined on the element. Due to the elementwise-discontinuous flux approximation, these element fluxes can be condensed on the element level and a modified element stiffness matrix is produced with only temperature unknowns as

$$\mathbf{A}_e [\mathbf{K}_{cc}^e + \mathbf{C}_{cc}^e] \mathbf{A}_e [\mathbf{c}^e] = \mathbf{A}_e [\mathbf{f}_c^e] \quad (3.55)$$

The additional matrix  $\mathbf{C}_{cc}$  and vector  $\mathbf{f}_c$  entries are computed by

$$\mathbf{C}_{cc}^e = -\mathbf{G}_{c\bar{q}}^e (\mathbf{K}_{\bar{q}\bar{q}}^e)^{-1} (\mathbf{K}_{\bar{q}c}^e + \mathbf{G}_{\bar{q}c}^e) \quad (3.56)$$

$$\mathbf{f}_c^e = -\mathbf{G}_{c\bar{q}}^e (\mathbf{K}_{\bar{q}\bar{q}}^e)^{-1} \mathbf{f}_{\bar{q}}^{\hat{c},e} \quad (3.57)$$

Splitting the assembled system Eq. (3.55) into nodes affected by boundary integrals with constraints (set  $\mathcal{C}$ ) and remaining nodes (set  $\mathcal{S}$ ), it becomes clear that the additional terms  $\mathbf{C}_{cc}^e$  and  $\mathbf{f}_c^e$  contribute only to nodes belonging to intersected elements.

$$\begin{bmatrix} \mathbf{K}_{cc}^{\mathcal{SS}} & \mathbf{K}_{cc}^{\mathcal{SC}} \\ \mathbf{K}_{cc}^{\mathcal{CS}} & \mathbf{K}_{cc}^{\mathcal{CC}} + \mathbf{C}_{cc}^{\mathcal{CC}} \end{bmatrix} \begin{bmatrix} \mathbf{c}^{\mathcal{S}} \\ \mathbf{c}^{\mathcal{C}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f}_c^{\mathcal{C}} \end{bmatrix} \quad (3.58)$$

The matrix  $C_{cc}$  is generally asymmetric. The asymmetric nature of a formerly symmetric problem may be seen as drawback, however, for the intended application – the NS equation with its asymmetric convection term – this property does not complicate the solution of the discrete system.

Approaches for constraints in discontinuous Galerkin (DG) methods [see e.g. Hemker et al., 2003; Lew and Buscaglia, 2008; Mergheim et al., 2004] have not been considered in detail, because the topic of this thesis is a CG method for the NS equation. Nevertheless, the research performed on discontinuous Galerkin (DG) methods may improve the understanding of CG methods. For example, reference [Lew and Buscaglia, 2008] remarks that either the primary variable  $c$  or the dual variable  $\lambda$  should be discontinuous. While the DG method lowers the continuity requirement for the primary variable, the proposed approach lowers the continuity of the Lagrange multiplier variable  $\bar{q}$ . Hence, the proposed scheme supports the finding in [Lew and Buscaglia, 2008].

In summary, the newly proposed method in Section 3.3.4 combines the following desirable features: (i) it requires no interface mesh for Lagrange multipliers; (ii) it is based on weak form techniques; (iii) it can be added to an existing weak forms and element formulations; (iv) its Lagrange multiplier can be condensed locally; (v) it avoids the saddle point structure of classical Lagrange multiplier methods; and (vi) it is stable without user-defined stabilization parameters. The application of this method to the three-dimensional incompressible NS equations largely benefits from the sum of these features.

## 3.4 Numerical integration

During the numerical integration of the weak form, the integration of elements containing discontinuous approximations requires a special treatment. Using standard integration rules to integrate across the discontinuity leads to reduced accuracy and often to rank deficient matrices. An accurate, but geometrically complex approach is the subdivision of the intersected element into integration cells to represent integrals on each side of the interface. Such subdivision is suggested in most publications that allow the inclusion of weak and strong discontinuities, see e.g. [Dolbow et al., 2000; Moës et al., 1999; Strouboulis et al., 2000] and is the standard technique in the context of XFEM. A similar treatment is required to perform boundary integrals due to interface constraints. The applied numerical integration schemes are discussed in the following.

In the classical FEM, domain integrals over a domain  $\Omega$  of function  $f$  are usually split into non-overlapping element integrals  $I_e$  over element domains

$\Omega_e$  such that

$$\begin{aligned} I &= \int_{\Omega} f d\mathbf{x} \\ &= \sum_e \int_{\Omega_e} f d\mathbf{x} \\ &= \sum_e I_e \end{aligned} \quad (3.59)$$

Each element integral  $I_e$  is integrated numerically in the element parameter space  $\xi$  using, e.g. efficient Gauss-Legendre integration, where the integral is replaced with a sum evaluating  $c$  at integration point  $\xi_q$  and multiplying with an integration weight  $w_q$  as

$$\begin{aligned} I_e &= \int_{\Omega_e} f(\mathbf{x}) d\mathbf{x} \\ &= \int_{\Omega_e} f(\xi) |J_{x\xi}| d\xi \\ &\approx \sum_q f(\xi_q) w_q \end{aligned} \quad (3.60)$$

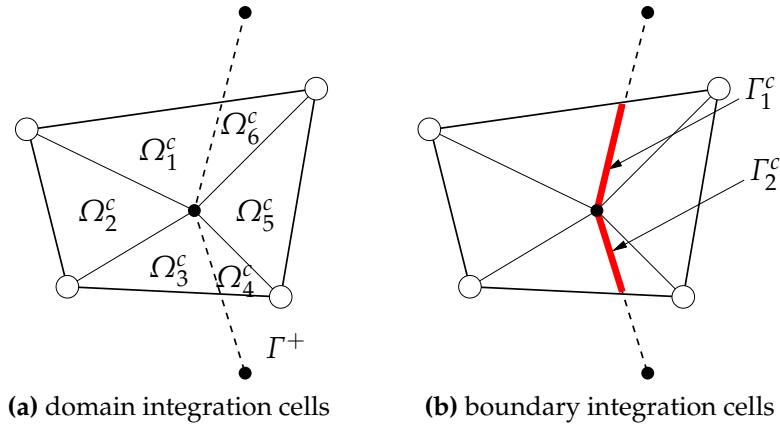
where  $q$  is the integration point counter. The Jacobian  $J_{x\xi}$  is defined as

$$J_{x\xi} = \frac{\partial \mathbf{x}}{\partial \xi} \quad (3.61)$$

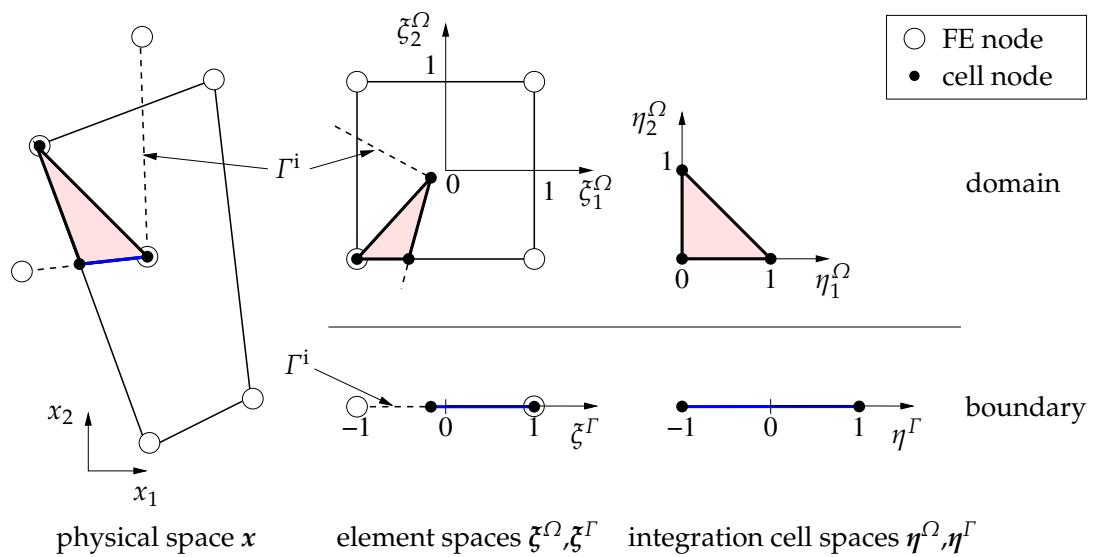
and  $|J_{x\xi}|$  denotes the determinant of the Jacobian matrix.

In the XFEM with strong discontinuities, all non-intersected elements – even if partially enriched – are integrated using Eq. (3.60). Intersected elements are subdivided into subdomains, such that no subdomain is crossed by a discontinuity. These subdomains are called domain integration cells or simply domain cells in the following. The interface  $\Gamma^i$  is subdivided into boundary integration cells to perform the numerical integration of surface integrals. Boundary cells are necessary, since kinks in the surface mesh as well as weak and strong discontinuities at edges and faces of each background element are present. In Figure 3.3, an element intersected by a piecewise linear interface  $\Gamma^+$  is depicted. The six resulting domain cells are shown in Figure 3.3a, while the two required boundary cells are depicted in Figure 3.3b. It can be seen, how boundary cells respect both the interface kink and the background element edges.

Figure 3.4 shows the associated coordinate systems for a two-dimensional problem. Integrals in the physical coordinates  $\mathbf{x}$  are transferred into domain and boundary element coordinate systems  $\xi^\Omega$  and  $\xi^\Gamma$ , respectively. Boundary elements are present, if the interface is an additional mesh or as part of the structure surface. Each integration cell has its own coordinate system:  $\eta^\Omega$  for domain cells and  $\eta^\Gamma$  for boundary cells, respectively.



**Figure 3.3:** Integration cells for an element intersected by a piecewise linear interface.



**Figure 3.4:** Coordinate systems for physical space, elements and integration cells

For the FE formulation, the shape function is typically a function of the element coordinates  $\xi$ . Hence,  $f$  and derivatives with respect to  $x_i$  can be expressed as a function of  $\xi$  as

$$f^h = \sum_{I=1}^{N^h} N_I(\xi) f_I \quad \text{and} \quad \frac{\partial f^h}{\partial x_i} = \sum_{I=1}^{N^h} \frac{\partial N_I(\xi)}{\partial \xi_j} \frac{\partial \xi_j}{\partial x_i} f_I \quad (3.62)$$

The domain integral in an element using integration cells can be expressed as

$$I_e = \int_{\Omega_e} f^h(\xi) \left| \frac{\partial x}{\partial \xi} \right| d\xi \quad (3.63)$$

$$= \sum_{c=1} \int_{\Omega_c} f^h(\xi(\eta)) \left| \frac{\partial x}{\partial \xi} \right| \left| \frac{\partial \xi}{\partial \eta} \right| d\eta \quad (3.64)$$

Numerical integration with quadrature points  $\eta_q$  and weights  $w_q$  defined for each integration cell is then performed as

$$I_e = \sum_{c=1} \sum_{q=1} f^h(\xi(\eta_q)) \left| \frac{\partial x}{\partial \xi} \right| \left| \frac{\partial \xi}{\partial \eta} \right| w_q \quad (3.65)$$

Integration of derivatives in element and cell coordinates is performed similarly

$$I_e = \int_{\Omega_e} \frac{\partial f^h(\xi)}{\partial \xi_j} \frac{\partial \xi_j}{\partial x_i} \left| \frac{\partial x}{\partial \xi} \right| d\xi \quad (3.66)$$

$$= \sum_{c=1} \int_{\Omega_c} \frac{\partial f^h(\xi(\eta))}{\partial \xi_j} \frac{\partial \xi_j}{\partial x_i} \left| \frac{\partial x}{\partial \xi} \right| \left| \frac{\partial \xi}{\partial \eta} \right| d\eta \quad (3.67)$$

Above equation illustrates that derivatives of approximation functions are with respect to  $\xi$  as in standard FE implementations. The only additional tasks are the computation of the coordinate  $\xi(\eta)$  and the Jacobi matrix  $|\partial \xi / \partial \eta|$ . Algorithm 3.1 summarizes the key steps when performing domain integrals. The necessary translations between physical, element and cell coordinate systems are included, too.

Boundary integrals are performed in a similar way. Along the boundary, no derivatives of variables with respect to surface coordinates are required, which removes the need for spatial derivatives. However, more coordinate transformations are required, if a discretized interface meshed for the Lagrange multiplier  $\lambda$  is used. Then boundary element shape function  $N^i$  including the corresponding coordinate transformations are required. If the interface mesh contains the prescribed interface value  $c^{i,h}$ , the value is interpolated to the integration point using shape function  $N^i$ . The required steps for performing boundary integrals are summarized in Algorithm 3.2.

---

**Algorithm 3.1** Integration of element domain integrals

---

```

for domain integration cell  $c^\Omega$  do
  for integration point  $q$  at position  $\eta_q^\Omega$  do
    position in fluid element coordinates  $\xi^\Omega(\eta_q^\Omega)$ 
    nodal shape function  $N_I(\xi^\Omega)$  and derivatives w.r.t.  $\xi^\Omega$ 
    shape function derivatives w.r.t.  $x$ 
    enriched approx. function  $\phi_I(\xi^\Omega)$  and derivatives w.r.t.  $x$ 
    integration factor  $w_q |J_{x\xi^\Omega}|, |J_{\xi^\Omega \eta^\Omega}|$ 
    integration point values  $c_q^h, \partial c_q^h / \partial x$  using  $\phi_L^c(\xi^\Omega)$ 
    integration point values  $\bar{q}_q^h$  using  $\phi_I^{\bar{q}}(\xi^\Omega)$ 
    weak form integrals
  end for
end for

```

---



---

**Algorithm 3.2** Integration of element boundary integrals

---

```

for boundary integration cell  $c^\Gamma$  do
  for integration point  $q$  at position  $\eta_q^\Gamma$  do
    position in boundary element coordinates  $\xi^\Gamma(\eta_q^\Gamma)$ 
    position in background element coordinates  $\xi^\Omega(\eta_q^\Gamma)$ 
    nodal shape function  $N_I^i(\xi^\Gamma)$  for boundary element
    shape function derivatives w.r.t.  $x$  for boundary element
    enriched approx. function  $\phi_I(\xi^\Omega)$  in background element
    integration factor  $w_q |J_{x\xi^\Gamma}| |J_{\xi^\Gamma \eta^\Gamma}|$ 
    integration point values  $c_q^h$  using  $\phi_I^c(\xi^\Omega)$ 
    integration point values  $\bar{q}_q^h$  using  $\phi_I^{\bar{q}}(\xi^\Omega)$ 
    integration point values  $\lambda_q^h$  using  $N_I^i(\xi^\Gamma)$ 
    integration point values  $c_q^{i,h}$  using  $N_I^i(\xi^\Gamma)$ 
    integrate weak form integrals
  end for
end for

```

---

**Table 3.1:** Exemplary shape functions for employed 3D elements.

Element type	Abbr. <sup>§</sup>	Shape function $N_1(\xi)$ of corner node 1
tet.	linear	$P_1$ $1 - \xi_1 - \xi_2 - \xi_3$
	quadr.	$P_2$ $(1 - \xi_1 - \xi_2 - \xi_3)(2(1 - \xi_1 - \xi_2 - \xi_3) - 1)$
hex.	linear	$Q_1$ $\frac{1}{8}(\xi_1 - 1)(\xi_2 - 1)(\xi_3 - 1)$
	quadr. Lagrange	$Q_2$ $\frac{1}{8}(\xi_1 - 1)(\xi_2 - 1)(\xi_3 - 1)\xi_1\xi_2\xi_3$
	quadr. serendipity	$Q_2^{(20)}$ $\frac{1}{8}(\xi_1 - 1)(\xi_2 - 1)(\xi_3 - 1)(\xi_1 + \xi_2 + \xi_3 + 2)$

<sup>§</sup> Abbr. = Abbreviation as used in [Gresho and Sani, 2000]

In practice, the construction of these integration cells is a geometrically complex task. The construction depends on the way the interface is given. The approach used in this work allows for higher order interfaces intersecting higher-order volume elements and has been published recently in [Mayer et al., 2009]. For three-dimensional problems, a triangulation of the intersecting surface into boundary integration cells is first generated. Subsequently, an constrained Delaunay tetrahedralization is performed in each intersected background element such that no domain cell is intersected by any interface  $\Gamma^i$ . Both cell types are implemented like elements using linear or quadratic shape functions and cell node coordinates. However, cells do not carry any nodal or element unknowns. For two-dimensional problems, the boundary cells are linear or higher order lines, while the domain cells are linear or higher order triangles.

How many integration points are required to compute the exact integral? Two answers to this question are given: the number of integration points for exact evaluation of integrals and the number of integration used in practice. For illustration of exact number of integration points, Table 3.1 gives examples for the five approximation functions used in this work, where an arbitrary nodal shape function is chosen. For transient problems in later chapters, the mass matrix contains the highest polynomial order, since here the two shape functions from the test and the trial functions are used without derivatives. Hence, the mass matrix represents the worst case requiring the highest number of integration points.

The domain integration cell can be arbitrarily oriented in the background element. For linear tetrahedral elements, the FE shape function along the cell boundary is linear for any cell orientation. Likewise, for quadratic tetrahedrons with piecewise plane interface elements, the shape functions remains quadratic. However, for trilinear hexahedral elements the trilinear shape function leads to cubic terms along some edges or faces of the tetrahedral integration cell. For the aforementioned mass matrix, a linear, Cartesian hex-

ahedral element with constant Jacobian  $|\partial\mathbf{x}/\partial\xi|$  requires to integrate a sixth order polynomial. A triquadratic hex27 element would require a twelfth order integration rule. A positive exception is the hex20 element with quadratic serendipity functions, where no polynomial with higher order than fourth order occurs. Here, the integral for the mass matrix needs integrations rules for eighth order polynomials. This leads to 24, 45 and 343 integration points per cell for hex8, hex20 and hex27 elements.

For testing and verification, high order Gauss rules for integrating polynomials of degree 16 (729 integration points) in tetrahedrons and of degree 15 (64 integration points) for triangles following [Peano, 1982] were implemented and applied. More efficient integration rules, although only available for certain polynomial orders, can be found in [Engels, 1980]. For an exhaustive overview including example code and integration points tables, see [Burkardt, 2010]. The testing supports the exact limits given above. Using lower order rules, exact integrals can not be computed.

This exact integration is only valid if undistorted background elements are considered, where the Jacobian between element and physical coordinates is piecewise constant in space for each element. In addition, only piecewise planar boundaries are allowed. For all other cases, Jacobians between physical and element coordinates as well as between element and integration cell coordinates become linear or higher order polynomials or even rational functions for the relation between physical space and boundary elements and cells.

The second answer is related to the practically needed number of integration points per cell. For coarse meshes, the discretization error may be higher than the integration error and a reduced integration is possible without noticeable deterioration of accuracy. Here adaptive integration may lead the optimal number of integration points, which has been used for enriched approximations for example in [Strouboulis et al., 2000]. If the number of integration cells is high enough, it is possible to reduce the number of integration points per cell down to an one-point integration rule. Hereby, the reduced integration results in higher integration errors on a given mesh, however, the convergence properties remain unchanged.

A critical lower limit is hit, if a further reduction the overall number of integration points per element leads to rank deficient element matrices. For instance, if one single domain integration cell determines the value of an integral, reduced integration leads to rank deficiencies in the element matrix. Such cases with only one integration cell occur quite frequently when void enrichments are combined with moving boundaries. For one integration cell, the exact integration rules plus additional order for linear or higher order Jacobian determinants have been used as the lower limit to obtain element matrices of full rank. In practice, the number of integration cells can be checked such that full integration rules are used if the number of cells is below a given

limit. Preliminary testing [Örley, 2009] revealed an optimal number of integration points depends on specific weak form integral terms. Sometimes a high-order rule with as few as possible integration cells or low order rules with high number of integration cells lead to a minimal number of overall integration points.

For curved interfaces and the resulting quadratic integration cells as introduced in [Mayer et al., 2009], exact integrals can not be computed due to the involved rational Jacobian determinants. Nevertheless, preliminary studies showed that it is generally beneficial to use higher order integration cells with curved boundaries. This conclusion is supported by recent studies in [Cheng and Fries, 2010]. For a given accuracy, domain cells with quadratic shapes required less overall integration points than using a higher number of linear tetrahedral cells with less points per cell. However, generating such higher order integration cells in three-dimensions is a complex geometrical task as it has been shown in [Mayer et al., 2009]; hence, linear integration cells may be preferred for increased robustness of the cell generation process. In this context, adaptive integration schemes as used in [Zlotnik and Díez, 2009] may guide the triangulation for curved interfaces.

As there are many components influencing the accuracy of the reduced integration, the required exact polynomial order is used for all results in this thesis. Thus, no rank deficiency occurs and the integration accuracy is comparable to standard FE integration in non-intersected elements.

Taking the overall number of integration points per element as a criterion, the integration cost for subdivided elements including exact integration is many times higher than for non-intersected elements with standard FE integration. For instance, a plane cut through the trilinear element in an one-sided problem generates at least five or six tetrahedral cells. Multiplying 6 cells with 24 integration points per cell gives 144. Compared to the eight integration points necessary for standard FE integration, 18 times more integration points are required for such intersected element. For complicated intersection patterns and quadratic background element shape functions, the factor can increase to 100 and more. In contrast, intersecting linear tetrahedral elements does not increase the polynomial order due to the missing  $\xi_1^\Omega \xi_2^\Omega \xi_3^\Omega$ -terms in the shape functions.

The runtime for exact integration could be improved by using alternative integration schemes as suggested in [Mousavi et al., 2010; Natarajan et al., 2009]. Here, an integration scheme for the polyhedron is constructed by minimizing the difference between the integral obtained by subdivision into integration cells and an integral obtained by direct integration of the polyhedron. Hereby, one polyhedron represents each side of the intersected element volume. These alternative schemes still represent exact integrals, but the final number of integration points can be controlled and reduced to a chosen error

limit. Hence, precomputing a special integration rule for each intersected element may reduce the time required for the repeated integration of the weak form.

In practice, the relative cost may not increase the overall runtime as dramatically as above numbers suggest. First, intersected elements are present only along the interface and, therefore, the relative cost between intersected and non-intersected elements reduces for increasing number of elements in the simulation. If  $h$ -adaptivity is applied and element subdivision is concentrated along the interface to resolve boundary features in the solution, then the relative cost decreases at a smaller rate than for uniform subdivision. Finally, for large systems, the solution of the linearized system often dominates the overall run time such that the increased cost for intersected elements becomes negligible. Hence, the effect of exact integration on the overall run time is considerable high for small simulations like the patch tests presented in later chapters, but is less dominant for large, realistic problems.

For parallel computations, an exception to this relative cheapness occurs, if most intersected elements are located on only few processors of a parallel simulation. Since the parallel layout is optimized based on node and element connectivity, it may happen, that the intersection and the element integration occurs on only few processors. For such situations, the parallel balancing may be extended to include information on the intersection situation. Even different parallel layouts for integrating the elements and solving the global linear system may be considered to minimize the overall run time of the simulation. It should also be noted that such layouts may change over time, if the interface moves as in the transient FSI simulations presented in subsequent chapters. For this thesis, no attempts have been made to implement rebalancing.

## 3.5 Numerical examples for the heat conduction

The following three examples demonstrate that the proposed flux based Lagrange multiplier works for plane and curved boundaries in three-dimensional problems. In each example, the interface is given by a number of planar quad4 surface elements. Hence, for plane surfaces, no geometric error disturbs the optimal convergence rate. For the curved surface in the third example, the surface is refined along with the domain refinement. However, suboptimal convergence rates are expected for the quadratic domain approximations due to the linear interface description.

For all three cases, an exact solution is constructed either by computing a source term from an assumed solution (examples in Section 3.5.1 and Section 3.5.2) or by analytically solving the heat equation (Section 3.5.3). For all

convergence graphs, the  $L_2$  norm defined as

$$\|c^h - c^{\text{exact}}\|_{L_2} = \varepsilon_{L_2}^c = \sqrt{\int_{\Omega^+} |c^h - c^{\text{exact}}|^2 dx} \quad (3.68)$$

is used. Note that in the graphs, sometimes the symbol  $T$  is used to denote temperature boundary conditions.

### 3.5.1 2D heat conduction in a rectangular domain

Solutions for the stationary heat conduction on rectangular domains as shown in Figure 3.5a can be constructed by superposition of one-dimensional solutions. For the two dimensional example, the chosen solution is

$$c_{2D}^{\text{exact}}(x, y, z) = \sin\left(\frac{\pi x}{a}\right) \sin\left(\frac{\pi y}{b}\right) \quad (3.69)$$

which results in the following source term

$$\hat{f}_{2D} = \left(\frac{1}{a^2} + \frac{1}{b^2}\right)\pi^2 \sin\left(\frac{\pi x}{a}\right) \sin\left(\frac{\pi y}{b}\right) \quad (3.70)$$

Using above heat source and homogeneous Dirichlet conditions along the boundary,  $c^{\text{exact}}$  shall be computed. Since three-dimensional elements are used for computation, zero Neumann conditions are applied on planes normal to the  $z$ -axis.

The background grid is rotated around the  $z$ -axis using various angles including the case of interfaces being parallel to the background mesh. The solution for a  $20^\circ$  rotation of the background mesh and the corresponding convergence rates are shown in Figure 3.5b and Figure 3.5c, respectively. For any angle, quadratic convergence for the hex8 and cubic convergence for hex20 and hex27 elements is obtained.

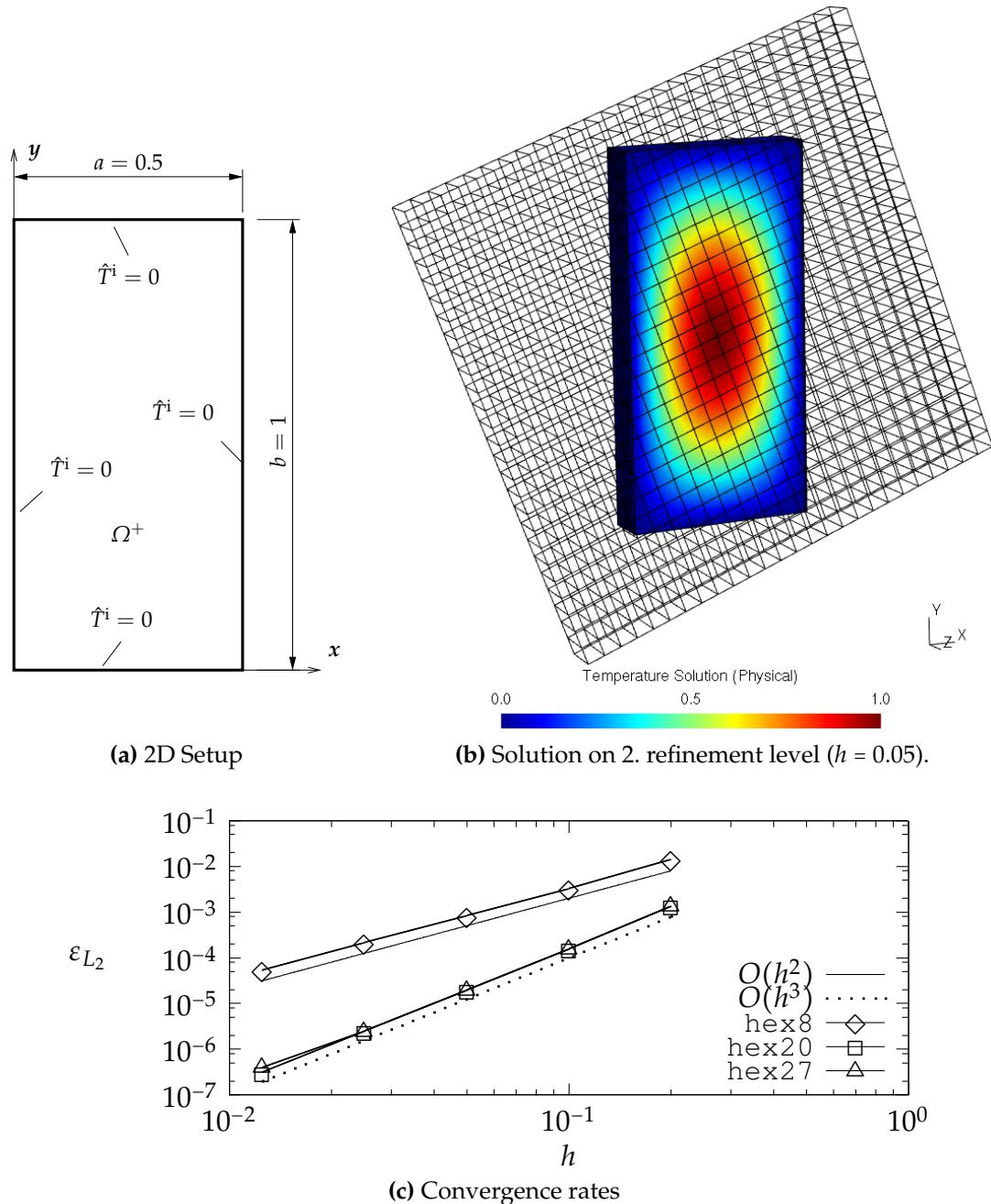
### 3.5.2 3D heat conduction in a rectangular domain

The second test case is a real three-dimensional problem, where the following exact solution is sought for

$$c_{3D}^{\text{exact}}(x, y, z) = \sin\left(\frac{x}{a}\pi\right) \sin\left(\frac{y}{b}\pi\right) \sin\left(\frac{z}{c}\pi\right) \quad (3.71)$$

The corresponding domain is depicted in Figure 3.6a and the corresponding source term is given as

$$\hat{f}_{3D} = \left(\frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2}\right)\pi^2 \sin\left(\frac{\pi x}{a}\right) \sin\left(\frac{\pi y}{b}\right) \sin\left(\frac{\pi z}{c}\right) \quad (3.72)$$



**Figure 3.5:** Convergence study of rectangular domain. The background grid is rotated about  $20^\circ$  around the  $z$ -axis.

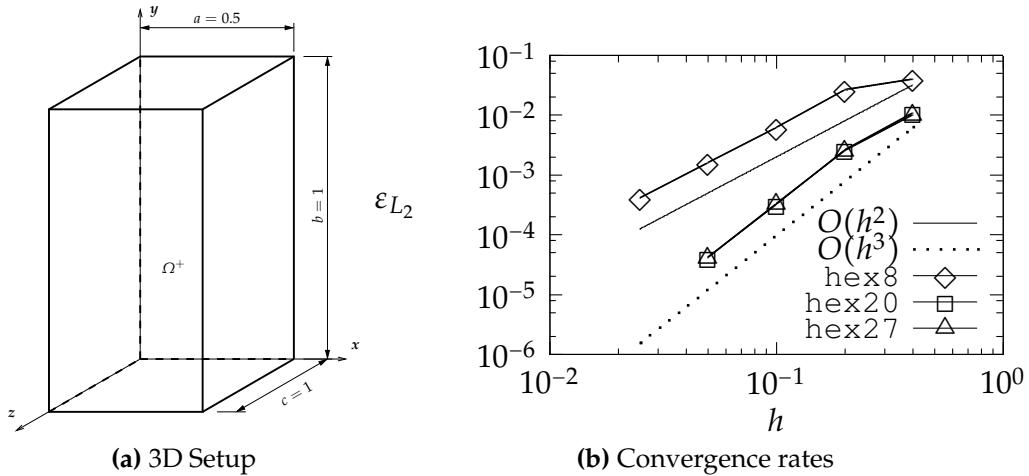


Figure 3.6: Convergence study of rectangular domain.

All boundaries are Dirichlet boundaries with zero temperature applied using the proposed embedded DBC approach.

For the convergence test whose results are plotted in Figure 3.6b the background mesh is first rotated by  $20^\circ$  around the  $x$ -axis and the resulting mesh is rotated  $10^\circ$  around the  $z$ -axis. Again quadratic convergence for the hex8 and cubic convergence for hex20 and hex27 elements is obtained.

### 3.5.3 Heat conduction between concentric cylinder surfaces

The last example for the diffusion equation is the heat conduction between two concentric cylinder surfaces. The setup and the applied temperature conditions are shown in Figure 3.7a. The solution for this problem reduces to an one-dimensional solution with respect to the radius  $r$  and can be computed analytically. The exact solution is given as

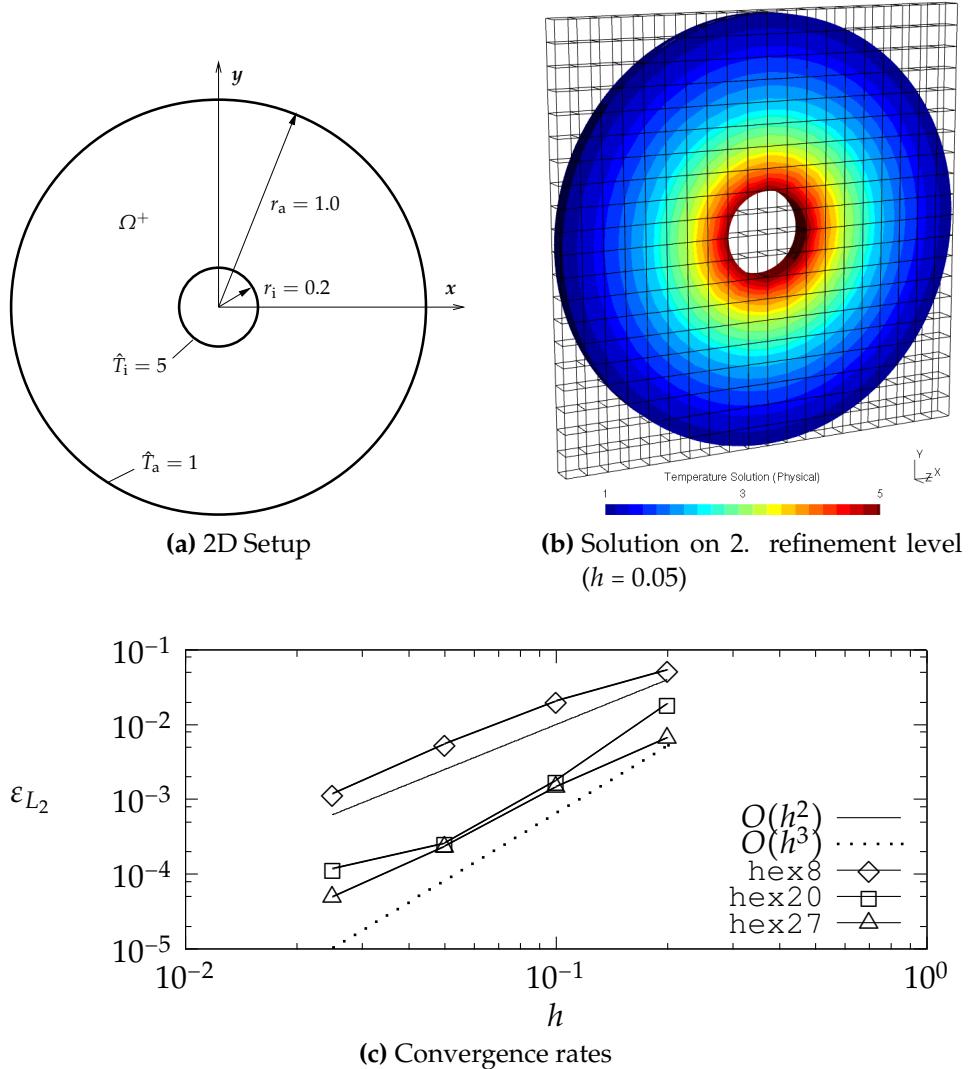
$$c_{\text{cyl}}^{\text{exact}}(r(x)) = c_i - (\ln(r) - \ln(r_i)) \frac{c_i - c_a}{\ln(r_a) - \ln(r_i)} \quad (3.73)$$

The two cylinder surfaces are aligned with the  $z$ -axis, hence the cylinder radius is defined as

$$r(x) = \sqrt{x^2 + y^2}$$

Again the grid is refined in  $x$ - and  $y$ -direction to estimate the spatial convergence rates given in Figure 3.7c. Reducing the element size of the interface with the same rate as the domain elements yields approximately quadratic convergence for the hex8 elements. However, the geometric error dominates the convergence of the serendipity and triquadratic elements, such that only reduced convergence rates between quadratic and cubic are obtained.

The source of the reduced convergence rate is most likely the bi-linear interface discretization along which the temperature is prescribed. Using smooth functions like level-set functions for interface representation, optimal (cubic) convergence rates are expected also for the hex20 and hex27 elements.



**Figure 3.7:** Setup and example solution for heat conduction between concentric cylinder surfaces.



# CHAPTER 4

## Stationary implicit fluid surfaces

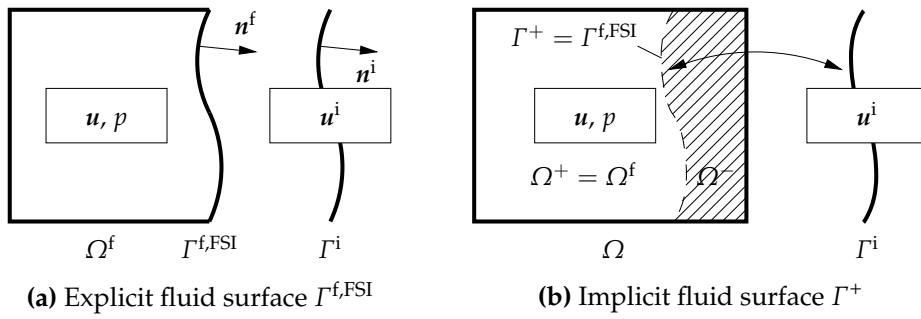
The key element of the proposed FSI approach is the proper definition of implicit interfaces on a fixed Eulerian fluid grid. This chapter illustrates, how the XFEM can be used to model implicit surfaces and how boundary conditions can be applied along such implicit boundaries. Moving interfaces and the FSI formulation are treated in the subsequent chapters.

### 4.1 Definition of the fluid problem

In standard FE simulations, the observed domain is filled with elements, such that element edges and element surfaces are aligned with material or domain boundaries. The XFEM allows to model such domain boundaries as discontinuities by enriching the finite element approximation, such that domain boundaries can arbitrarily cross element surfaces. It is therefore possible to create a computational mesh without considering where the surface is located. This property of the XFEM is exploited by first defining a computational domain and then describing the fluid surface implicitly, i.e. independent of any fluid element boundary.

The introduction of the three-field setup in Section 2.3.2 allows to concentrate on the fluid flow and its interaction with an explicitly given interface  $\Gamma^i$ . No knowledge is assumed on what is beyond the interface, such that dependencies between fluid and structure fields are avoided. The interface is given as a Lagrangian mesh and no other means of locating the interface, e.g. level-sets, are required.

The first step is the replacement of the explicit fluid surface description (Figure 4.1a) with an implicit description (Figure 4.1b). For this purpose, domain  $\Omega$  is defined that contains the fluid domain  $\Omega^f$  completely and extends



**Figure 4.1:** Fluid part of the FSI problem:  $\Gamma^i$  divides the computational fluid domain  $\Omega$  into a physical fluid domain  $\Omega^+ = \Omega^f$  and a ‘fictitious’ fluid domain  $\Omega^-$ .

into the structural domain  $\Omega^s$ . The interface  $\Gamma^i$  between fluid and structure is an internal interface that separates  $\Omega$  into two subdomains  $\Omega^+$  and  $\Omega^-$ , where  $\Omega^+$  corresponds to the *physical* fluid domain  $\Omega^f$  and  $\Omega^-$  is the remaining domain filling  $\Omega$ . The flow field in  $\Omega^-$  is entirely *fictitious* with no physical meaning to the FSI problem. In the XFEM literature, this setup is sometimes referred to as one-sided problem, where the field of interest – here the physical fluid in  $\Omega^f$  – is on one side of the interface only. The other side is denoted as void.

The jump in the velocities  $[\![\mathbf{u}]\!]$  between the physical values  $\mathbf{u}^+$  and the void ( $\mathbf{u}^-$ ) can be expressed as

$$[\![\mathbf{u}]\!] = \mathbf{u}^+ - \underbrace{\mathbf{u}^-}_{=0} \quad \text{in } \Gamma^i \quad (4.1)$$

In other words, the jump height  $[\![\mathbf{u}]\!]$  equals the value of  $\mathbf{u}^+$  along the interface, since  $\mathbf{u}$  is zero within the void. Consequently, the kinematic fluid-interface coupling from Eq. (2.68) along  $\Gamma^i$  becomes

$$[\![\mathbf{u}]\!] = \mathbf{u}^+ = \mathbf{u}^f = \mathbf{u}^i \quad \text{in } \Gamma^i \quad (4.2)$$

Likewise, one can identify a jump discontinuity in the stress field

$$[\![\boldsymbol{\sigma} \cdot \mathbf{n}^f]\!] = \boldsymbol{\sigma}^+ \cdot \mathbf{n}^f - \underbrace{\boldsymbol{\sigma}^- \cdot \mathbf{n}^f}_{=0} \quad \text{in } \Gamma^i \quad (4.3)$$

The kinematic condition between fluid and interface from Eq. (2.70) then becomes

$$[\![\boldsymbol{\sigma} \cdot \mathbf{n}^f]\!] = \boldsymbol{\sigma}^+ \cdot \mathbf{n}^f = \boldsymbol{\sigma}^f \cdot \mathbf{n}^f = \lambda \quad \text{in } \Gamma^i \quad (4.4)$$

The term  $\boldsymbol{\sigma}^+ \cdot \mathbf{n}^f = \lambda$  denotes the surface traction, which is the ‘reaction force’, if velocity constraints are enforced along the interface. The initial conditions

for the fluid domain  $\Omega^f$  are given as

$$\mathbf{u}(\mathbf{x}, t = 0) = \mathbf{u}^0 \quad \text{in } \Omega^+ \quad (4.5)$$

$$p(\mathbf{x}, t = 0) = p^0 \quad \text{in } \Omega^+ \quad (4.6)$$

As shown in the following section, the XFEM allows to sharply separate physical and fictitious domain and no flow is computed in  $\Omega^-$ . Consequently, no initial or boundary conditions are required for  $\Omega^-$  and its boundaries. Fluid body forces, if present, are only applied to  $\Omega^+$ . Hence, for the one-sided discontinuity, the values on the + side are equal to the physical values such that no extra notation needs to be kept to distinguish + side and - side of the discontinuity. The enforcement of conditions along the discontinuity is described after the XFEM formulation is presented in the subsequent section.

## 4.2 XFEM formulation

### 4.2.1 Discretization

For non-moving fluid surfaces as assumed in this chapter, the Eulerian fluid description is used. Time-discretization and stabilization due to equal order shape functions and convective effects is identical to the techniques described for the FE fluid formulation in Section 2.1. Hence, the semi-discrete, stabilized Eulerian weak form given in Eq. (2.29) is used as the starting point for the XFEM discretization.

The spatial approximation is changed from an explicit to an implicit surface using enriched approximations. For example, adopting the notation from Section 3.2, the discrete velocity in each element  $e$  can be written as

$$\mathbf{u}_e^h(\xi, t) = \sum_L^{n_L} \underbrace{N_L(\xi) \psi_L(\xi)}_{\phi_L(\xi)} \mathbf{u}_L(t) \quad (4.7)$$

The enrichment function  $\psi_L(\xi)$  for unknown parameter  $L$  can be a constant value or a special function like the Heaviside function. In the former case, the approximation function  $\phi_L$  equals the standard FE shape functions  $N_L$ , whereas in the latter case the approximation function has a jump-like behavior. The number of unknowns  $n_L$  for a scalar or vector component is independent from the number of nodes  $n^N$ . For non-moving interfaces, all  $\psi_L(\xi)$  are independent of time.

The stabilized, equal order XFEM formulation employs the same principles as the stabilized FE formulation given in Section 2.1. The Bubnov-Galerkin method requires that trial and test approximations for each field

are chosen to be equal. In addition, the shape functions for velocity and pressure is equal, i.e.  $\phi^u(\xi^\Omega) = \phi^p(\xi^\Omega)$ . Velocity and pressure are enriched in the same way by using standard and void enrichment functions following the rules given in the next section. Consequently, velocity, pressure and their test functions all use the same approximation functions as

$$\mathbf{u}^{h,n+1}(x) = \sum_L \phi_L(\xi) \mathbf{u}_L^{n+1} \quad (4.8a)$$

$$\delta \mathbf{u}^h(x) = \sum_L \phi_L(\xi) \delta \mathbf{u}_L \quad (4.8b)$$

$$p^{h,n+1}(x) = \sum_L \phi_L(\xi) p_L^{n+1} \quad (4.8c)$$

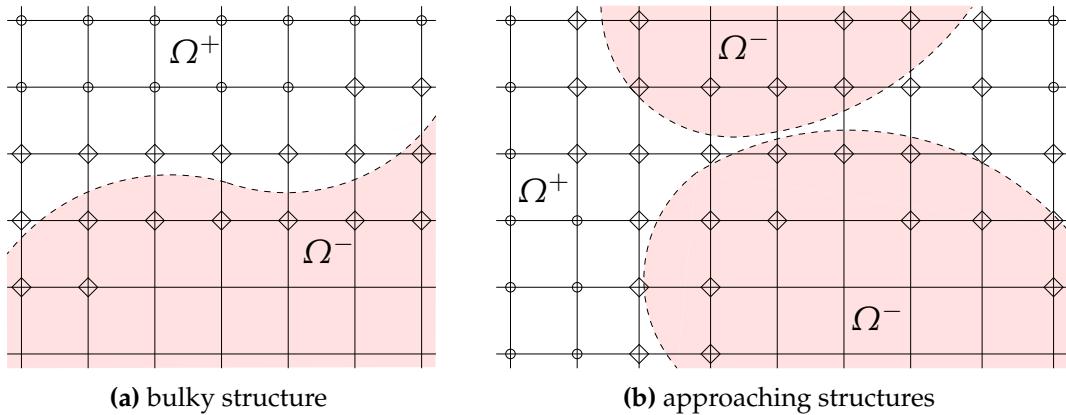
$$\delta p^h(x) = \sum_L \phi_L(\xi) \delta p_L \quad (4.8d)$$

Higher order shape functions are incorporated without additional effort making this approach capable for consistent low and high order finite element approximations. Different shape function for pressure and velocity as used, e.g. for the inf-sup stable Taylor-Hood element [Gresho and Sani, 2000], have not been tested within this work. For additional information on applying the XFEM to incompressible materials, see also [Legrain et al., 2008].

### 4.2.2 Enrichment strategies

The FSI problem requires one or several void enrichments in combination with standard FE shape functions. So far, the void enrichment function  $\bar{\psi}$  has been defined with respect to a general interface. For fluid-structure interaction, complex scenarios like contact and thin structures require to distinguish multiple interfaces that intersect an element or the approximation function support of nodal unknowns. A relatively simple rule governs all enrichment scenarios that are discussed in the following: one void enrichment is needed for each independent fluid domain within the shape function support of a nodal or elemental degree of freedom (DOF).

For bulky structures, only one interface intersects a fluid element as depicted in Figure 4.2a. Here, all nodes belonging to intersected element are enriched using the void-enrichment. All remaining nodes in  $\Omega^+$  use the standard DOFs, while the remaining nodes in  $\Omega^-$  carry no DOFs at all. This scenario corresponds to the one-dimensional example in Figure 3.2b. Only full elements and integration cells located in  $\Omega^+$  have to be integrated. This setup resembles approaches, where integration is performed only on parts of an element to account for discontinuities [Hansbo and Hansbo, 2004; Tezduyar, 2006]. Although no enrichment concepts are introduced in these schemes, the resulting linear system is identical. A principal example of such an enrichment is given in Section 4.5.1.

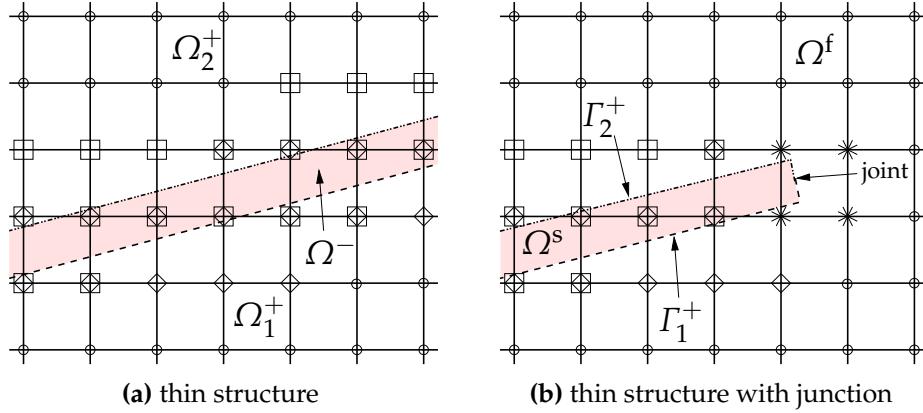


**Figure 4.2:** Enrichment for bulky and approaching structures: The void enrichments ( $\diamond$ ) are applied to nodes of intersected elements. Standard DOFs ( $\circ$ ) are used in the remaining fluid domain.

Approaching submersed structures are algorithmically similar to the previous case. All structural domains  $\Omega_n^s$  together can be treated as one domain  $\Omega^-$  such that  $\Omega^- = \sum_n \Omega_n^s$ . Figure 4.2b shows the resulting enrichment pattern. There are only four unknowns per node, since only one void enrichment is required to model the fluid between the structure surfaces. An illustrative simulation is presented in Section 6.6.6. From the view point of the XFEM based fixed-grid method it is algorithmically easy to model approaching bodies. However, the physics of fluid flow between contacting bodies is beyond the scope this thesis. If the gap between the two bodies closes, then separated fluid regions may appear and multiple local fluids have to be considered as for thin structures treated next.

Thin structures require to locally distinguish separate interfaces. It can be seen in Figure 4.3a, how for a certain node two fluids are present in its approximation function support. Physically, the structure divides the fluid domain into two independent fluid domains  $\Omega_1^+$  and  $\Omega_2^+$ . From the perspective of  $\Omega_1^+$ , everything that is behind  $\Gamma_1^i$  is a void, which also includes  $\Omega_2^+$ . Analog, for  $\Omega_2^+$ , everything behind  $\Gamma_2^i$  is void. To distinguish the two void enrichments, an independent surface identifier or label for each discontinuity is introduced. Each local void enrichment corresponds to its particular interface  $i$  and since its value is determined by the local feature, it is termed  $\bar{\psi}_i^{\text{loc}}(x)$ . An example computation showing the local enrichment is given in Section 4.5.1.

The most complex case occurs, if the FSI interface around a thin structure is a closed surface. Then the strategy for thin structures leads to a point (2d) or line (3d), where two interface labels join together. This problem exists only in dimensions higher than one. For the treatment of this joint, two types of void enrichment are distinguished: The first is the already introduced local



**Figure 4.3:** Enrichment for thin structures: For thin structures, overlapping void enrichments ( $\diamond$  and  $\square$ ) represent the distinct discontinuities  $\bar{\psi}_1^{\text{loc}}$  and  $\bar{\psi}_2^{\text{loc}}$ . Standard DOFs ( $\circ$ ) are used for the remaining fluid domain. For connecting distinct enrichments, an FSI void enrichment  $\bar{\psi}^{\text{glob}}$  (\*) is applied at junctions.

enrichment  $\bar{\psi}_i^{\text{loc}}(x)$ . In contrast, the FSI-void enrichment function  $\bar{\psi}^{\text{glob}}$  simply distinguishes between being inside or outside of any structure domain  $\Omega_i^s$  in the simulation. If a point is outside any structural domain, the fluid domain is assumed and the value of  $\bar{\psi}^{\text{glob}}$  equals one. If an interface with label  $i = 1$  forms a closed boundary with large enough interior, the use of  $\bar{\psi}^{\text{glob}}$  and  $\bar{\psi}_i^{\text{loc}}$  becomes equivalent.

An exemplary enrichment pattern is given in Figure 4.3b. Two surfaces with label 1 and 2 join at the right end of the structure. First the FSI void enrichment (\*) is applied, where two connected interface labels cross the support. Subsequently, the remaining nodes are enriched as explained in the thin structure case. On the lower left side, again two fluids can be distinguished locally. Two example simulations for such thin structures are given in Section 4.5.1.

In practice, the FSI void enrichments are applied first and subsequently, each interface is enriched independently. Finally, the standard DOFs are applied in the remaining fluid domain. The specific order leads to extra effort, when multiple processors are involved. This case is studied in a little more detail in Section 4.4.3.

For all presented enrichment situations, if by mistake too many enrichments are applied, linear dependent equations appear, where DOFs have exactly the same approximation. Since the singular system is insolvable, such ‘over’-enrichment can be found easily. If however, insufficient enrichments are applied, the approximation deteriorates without worsening the correctness of the linear system. Such mistakes can only be found by carefully exam-

ining the enrichment process and the solution for clearly defined examples. Such examples are presented in Section 4.5.1 and Section 4.5.2.

### 4.2.3 Practical considerations

A well known property of the XFEM is that particular enrichment situations lead to high condition numbers of the linearized system. For the particular case of using void enrichment functions, small element fractions with non-zero approximation functions behave similar to small elements that occur for example when  $h$ -adaptivity is applied. In contrast to adaptivity, where the minimal element size is bounded by constraining the maximal number of subdivisions, the moving interface as introduced in the next chapter generates arbitrarily small non-zero supports for an unknown. The resulting high condition number hampers the intended use of parallel and iterative algebraic multigrid (AMG) solution techniques.

As a first remedy, degrees of freedom with almost zero matrix entries are omitted from critical nodes or elements. In this work, the criterion to stop enrichment of DOFs is met, when the integral over the approximation function is less than a given threshold. The threshold is usually problem dependent and is found by trial and error. For more robust and user-friendly simulations, this rather heuristic criterion needs to be improved. In addition, parallel direct solvers are applied that are less sensitive to high condition numbers than iterative solution techniques. In particular the solver package SuperLU has been used through the Trilinos based research code. The SuperLU solver allows direct solutions on distributed memory machines, but is effective only up to 16 or 32 processors. If more processors and associated computer memory are required for complex three-dimensional transient fluid simulations, advanced preconditioning techniques for the linearized system are required before the present approach can be used for really large scale flow simulations. Examples for such tailored preconditioning in the context of crack tip enrichments can be found in [Béchet et al., 2005]. Applying the proposed techniques to the NS equations should be a logical next step to improve the global conditioning of the linear system.

Beyond the influence on the condition number, the effect of the element subdivision on the fluid stabilization parameters may require further attention. If intersected elements with small physical (and therefore large fictitious) fractions are interpreted as small elements, the element length scales  $h_e^V$  and  $h_e^u$  used in the definition of the fluid stabilization parameters may require adaptation. For the presented results, the stabilization parameters are computed at each integration point within each integration cell. For the definition of  $h_e^V$  (Eq. (2.33)), the volume of the entire element has been used. For the stream-length based length scale  $h_e^u$  (Eq. (2.34)), the shape function  $N$  is

replaced by the enriched approximation function  $\phi$  and the physical velocity at the integration point is computed using the enriched approximation in Eq. (4.8a). For the results in this thesis, optimal convergence rates are obtained for stationary flow problems and stationary boundaries with above definitions. In addition, no particular problems have been observed for in-stationary problems. Additional theoretical work should clarify, whether is numerical observation is valid.

### 4.3 Enforcement of interface constraints

The strong form of the NS equation in Eulerian formulation including the interface condition at  $\Gamma^i$  reads

$$\mathbf{r}_u(\mathbf{u}, p) = \mathbf{0} \quad \text{in } \Omega^f \quad (4.9a)$$

$$\mathbf{r}_p(\mathbf{u}) = 0 \quad \text{in } \Omega^f \quad (4.9b)$$

$$\mathbf{u} - \mathbf{u}^i = \mathbf{0} \quad \text{in } \Gamma^i \quad (4.9c)$$

$$\mathbf{u} - \hat{\mathbf{u}}_D = \mathbf{0} \quad \text{in } \Gamma_D^f \quad (4.9d)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n}^f - \hat{h} = 0 \quad \text{in } \Gamma_N^f \quad (4.9e)$$

Since  $\Gamma^i$  generally does not align with element boundaries, Dirichlet and coupling conditions require special treatment. This topic is also referred to as embedded Dirichlet problem [Dolbow and Franca, 2008; Ji and Dolbow, 2004; Moës et al., 2006].

The task for this section is to formulate the coupling between the fluid velocity of the background grid and the velocity on the interface mesh. The interface mesh consists — in the three-dimensional case — of triangular or quadrilateral elements, whereas the implicit surface of the physical fluid on the background mesh does not have such surface elements. On the background mesh, the velocity and the stress along the surface are given only by their interpolation from the tetrahedral or hexahedral domain elements. For highlighting the surface coupling character, two sets are defined. Set  $\mathcal{C}$  contains all DOFs from nodes of intersected elements, while all remaining nodal velocity and pressure DOFs are in set  $\mathcal{S}$ . With the help of these two sets, the stabilized fluid system can be split into the following form

$$\begin{bmatrix} \mathbf{F}_{uu}^{SS} & \mathbf{F}_{up}^{SS} & \mathbf{F}_{uu}^{SC} & \mathbf{F}_{up}^{SC} \\ \mathbf{F}_{pu}^{SS} & \mathbf{F}_{pp}^{SS} & \mathbf{F}_{pu}^{SC} & \mathbf{F}_{pp}^{SC} \\ \mathbf{F}_{uu}^{CS} & \mathbf{F}_{up}^{CS} & \mathbf{F}_{uu}^{CC} & \mathbf{F}_{up}^{CC} \\ \mathbf{F}_{pu}^{CS} & \mathbf{F}_{pp}^{CS} & \mathbf{F}_{pu}^{CC} & \mathbf{F}_{pp}^{CC} \end{bmatrix}_k \begin{bmatrix} \Delta \mathbf{u}^{\mathcal{S}} \\ \Delta p^{\mathcal{S}} \\ \Delta \mathbf{u}^{\mathcal{C}} \\ \Delta p^{\mathcal{C}} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_u^{\mathcal{S}} \\ \mathbf{r}_p^{\mathcal{S}} \\ \mathbf{r}_u^{\mathcal{C}} \\ \mathbf{r}_p^{\mathcal{C}} \end{bmatrix}_k \quad (4.10)$$

In the following, three methods to enforce interface constraints are presented and their applicability to the two-dimensional and three-dimension-

al fluid problem is discussed. Stating the conclusion in advance, the Lagrange multiplier method described first is difficult to formulate and implement in three dimensions, hence only a two-dimensional implementation is presented. The following two methods are applicable in one, two, and three dimensions and are therefore implemented for three dimensional problems as the most general case.

### 4.3.1 Traction based Lagrange multiplier

In the ‘classical’ Lagrange multiplier approach to constraints along boundaries, a traction field  $\lambda$  is introduced as independent unknown. The corresponding weighting function is denoted as  $\delta\lambda$ . The fluid-interface condition (Eq. (4.2)) is enforced weakly at  $t^{n+1}$  at the interface  $\Gamma^i$  by testing the condition with a test function  $\delta\lambda(\mathbf{x})$

$$0 = (\delta\mathbf{u}, \mathbf{r}_u(\mathbf{u}, p))_{\Omega^f} + (\delta\mathbf{u}, \boldsymbol{\sigma} \cdot \mathbf{n}^f - \hat{\mathbf{h}})_{\Gamma_N^f} + (\delta\mathbf{u}, \boldsymbol{\sigma} \cdot \mathbf{n}^f - \lambda)_{\Gamma^i} \quad (4.11a)$$

$$0 = (\delta p, \mathbf{r}_p(\mathbf{u}))_{\Omega^f} \quad (4.11b)$$

$$0 = -(\delta\lambda, \mathbf{u} - \mathbf{u}^i)_{\Gamma^i} \quad (4.11c)$$

$$0 = (\delta\mathbf{u}^i, \lambda)_{\Gamma^i} \quad (4.11d)$$

The interface velocity is assumed to be unknown during this derivation. Consequently, the term  $(\delta\mathbf{u}^i, \lambda)_{\Gamma^i}$  is introduced. Since the interface does not introduce any physical effects,  $\lambda$  is zero until further coupling to the structure is introduced in later chapters. As in the finite element method, integrating the stress term in the momentum equation Eq. (4.11a) by parts yields

$$-(\delta\mathbf{u}, \nabla \cdot \boldsymbol{\sigma})_{\Omega^f} = -(\delta\mathbf{u}, \boldsymbol{\sigma} \cdot \mathbf{n}^f)_{\Gamma^f} + (\nabla \cdot \delta\mathbf{u}, \boldsymbol{\sigma})_{\Omega^f} \quad (4.12)$$

For Dirichlet boundary conditions along  $\Gamma_D^f$ , the surface integral vanishes due to the vanishing test function. Neumann conditions are applied by letting  $\boldsymbol{\sigma} \cdot \mathbf{n}^f = \hat{\mathbf{h}}^f$  along  $\Gamma_N^f$ . Along the XFEM boundary  $\Gamma^i$ , where velocities and pressure jump from their physical values in  $\Omega^+$  to zero in  $\Omega^-$ , the traction  $\lambda$  corresponds to a real reaction force and is defined as

$$\boldsymbol{\sigma} \cdot \mathbf{n}^f - \lambda = \mathbf{0} \quad \text{in } \Gamma^i \quad (4.13)$$

The velocity matching condition along  $\Gamma^i$  is only enforced for the velocity field in  $\Omega^+$ . When the volume of  $\Omega^-$  changes, no artificial pressure is generated within  $\Omega^-$  at any time, as no flow is modeled in  $\Omega^-$ .

As mentioned above, the fluid equations need stabilizing terms for dominating convection and equal order interpolation as used in the following.

Using the fluid momentum operator  $F_M$  defined in Eqs. (2.29a) and (2.29b), the stabilized weak equations can be written as

$$0 = F_M(\delta\mathbf{u}, \mathbf{u}, p) - (\delta\mathbf{u}, \boldsymbol{\lambda})_{\Gamma^i} \quad (4.14a)$$

$$0 = F_C(\delta p, \mathbf{u}, p) \quad (4.14b)$$

$$0 = -(\delta\boldsymbol{\lambda}, \mathbf{u} - \mathbf{u}^i)_{\Gamma^i} \quad (4.14c)$$

$$0 = (\delta\mathbf{u}^i, \boldsymbol{\lambda})_{\Gamma^i} \quad (4.14d)$$

Neumann boundary integrals are already included in the stabilized fluid operator  $F_M$ . The continuity equation Eq. (4.14b) is unaffected by the Lagrange multiplier formulation. No stabilization terms appear on the Lagrange multiplier expression  $(\delta\mathbf{u}, \boldsymbol{\lambda})_{\Gamma^i}$ .

The Lagrange multiplier is discretized on the interface mesh. It is the slave side of the weak fluid-interface coupling, if the naming conventions of Mortar methods are adopted. Interface velocity  $\mathbf{u}^i$  and Lagrange multiplier  $\boldsymbol{\lambda}$  use the same approximation  $N_I^i$  as introduced in Eq. (2.73), such that

$$\boldsymbol{\lambda}^h = \sum_I N_I^i \boldsymbol{\lambda}_I \quad \text{and} \quad \delta\boldsymbol{\lambda}^h = \sum_I N_I^i \delta\boldsymbol{\lambda}_I \quad (4.15)$$

The proper relation between fixed-grid field velocity approximation and Lagrange multiplier approximation has been the topic of recent research efforts, see e.g. [Fernández-Méndez and Huerta, 2004; Flemisch and Wohlmuth, 2007; Ji and Dolbow, 2004; Moës et al., 2006]. In particular, not all approximation combinations are inf-sup stable. In this thesis, stable results have been obtained in a two-dimensional implementation, if stabilized, equal-order, and quadratic fluid elements ( $Q_2Q_2$ ) have been used in combination with linear shape functions for the Lagrange multiplier interpolation. The linear Lagrange multiplier elements have been placed at the intersection of the given interface with the fluid element edges as depicted in Figure 4.4. In the picture, the intersection points are derived from a smooth surface intersecting the fluid elements. The intersection between fluid edges and discrete surface mesh (the structure surface) is given in Chapter 6.

The discrete system to solve is

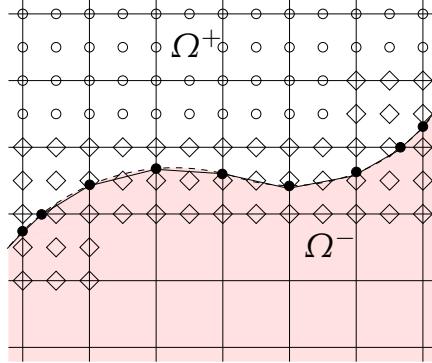
$$\delta\mathbf{u} [r_u(\mathbf{u}, p) + r_u^\lambda(\boldsymbol{\lambda})] = F_M(\delta\mathbf{u}^h, \mathbf{u}^h, p^h) - (\delta\mathbf{u}^h, \boldsymbol{\lambda}^h)_{\Gamma^i} \quad (4.16a)$$

$$\delta p \, r_p(\mathbf{u}, p) = F_C(\delta p^h, \mathbf{u}^h, p^h) \quad (4.16b)$$

$$\delta\boldsymbol{\lambda} [r_\lambda^\mathbf{u}(\mathbf{u}) + r_\lambda^{u^i}(\mathbf{u}^i)] = -(\delta\boldsymbol{\lambda}^h, \mathbf{u}^h - \mathbf{u}^{i,h})_{\Gamma^i} \quad (4.16c)$$

$$\delta\mathbf{u} \, r_{u^i}^\lambda(\boldsymbol{\lambda}) = (\delta\mathbf{u}^{i,h}, \boldsymbol{\lambda}^h)_{\Gamma^i} \quad (4.16d)$$

The discrete Lagrange multiplier unknown vector is denoted as  $\boldsymbol{\lambda}$ . After lin-



**Figure 4.4:** Thick structures with traction Lagrange Multiplier: Lagrange multiplier nodes ( $\bullet$ ) are added at intersection points between interface and fluid edges.

earization, the discrete system at Newton step  $k$  is

$$\begin{bmatrix} \mathbf{F}_{uu} & \mathbf{F}_{up} & \mathbf{M}_{u\lambda} & \mathbf{0} \\ \mathbf{F}_{pu} & \mathbf{F}_{pp} & \mathbf{0} & \mathbf{0} \\ \mathbf{M}_{\lambda u} & \mathbf{0} & \mathbf{0} & \mathbf{D}_{\lambda u^i} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_{u^i \lambda} & \mathbf{0} \end{bmatrix}_k \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{p} \\ \Delta \boldsymbol{\lambda} \\ \Delta \mathbf{u}^i \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_u + \mathbf{r}_u^\lambda \\ \mathbf{r}_p \\ \mathbf{r}_\lambda^u + \mathbf{r}_\lambda^{u^i} \\ \mathbf{r}_{u^i}^\lambda \end{bmatrix}_k \quad (4.17)$$

The Lagrange multiplier matrices are gained from linearization, that is by partial derivatives with respect to the discrete velocities  $\mathbf{u}$  and  $\mathbf{u}^i$  as

$$\delta \boldsymbol{\lambda} \mathbf{M}_{\lambda u} = - \frac{\partial (\delta \boldsymbol{\lambda}^h, \mathbf{u}^h)_{\Gamma^i}}{\partial \mathbf{u}} \quad (4.18)$$

$$\delta \boldsymbol{\lambda} \mathbf{D}_{\lambda u^i} = + \frac{\partial (\delta \boldsymbol{\lambda}^h, \mathbf{u}^{i,h})_{\Gamma^i}}{\partial \mathbf{u}^i} \quad (4.19)$$

and by using  $\mathbf{M}_{u\lambda} = \mathbf{M}_{\lambda u}^T$  and  $\mathbf{D}_{\lambda u^i} = \mathbf{D}_{u^i \lambda}^T$ . The fluid matrices  $\mathbf{F}_{uu}$ ,  $\mathbf{F}_{up}$ ,  $\mathbf{F}_{pu}$  and  $\mathbf{F}_{pp}$  are identical to those in Eq. (2.41).

If the interface velocity is given and prescribed as a Dirichlet condition along the entire interface  $\Gamma^i$ , corresponding columns and rows of the system matrix can be omitted and the known interface velocities only appear in the residual for  $\mathbf{r}_\lambda^{u^i}(\hat{\mathbf{u}}^i)$  with  $\hat{\mathbf{u}}^i$  being the given interface velocity. To emphasize that only fixed-grid unknowns connected to intersected elements are involved in the coupling, the fluid is split into the two sets as introduced in Eq. (4.10)

$$\begin{bmatrix} \mathbf{F}_{uu}^{SS} & \mathbf{F}_{up}^{SS} & \mathbf{F}_{uu}^{SC} & \mathbf{F}_{up}^{SC} & \mathbf{0} \\ \mathbf{F}_{pu}^{SS} & \mathbf{F}_{pp}^{SS} & \mathbf{F}_{pu}^{SC} & \mathbf{F}_{pp}^{SC} & \mathbf{0} \\ \mathbf{F}_{uu}^{CS} & \mathbf{F}_{up}^{CS} & \mathbf{F}_{uu}^{CC} & \mathbf{F}_{up}^{CC} & \mathbf{M}_{u\lambda}^C \\ \mathbf{F}_{pu}^{CS} & \mathbf{F}_{pp}^{CS} & \mathbf{F}_{pu}^{CC} & \mathbf{F}_{pp}^{CC} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{\lambda u}^C & \mathbf{0} & \mathbf{0} \end{bmatrix}_k \begin{bmatrix} \Delta \mathbf{u}^S \\ \Delta \mathbf{p}^S \\ \Delta \mathbf{u}^C \\ \Delta \mathbf{p}^C \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_u^S \\ \mathbf{r}_p^S \\ \mathbf{r}_u^C + \mathbf{r}_u^{\lambda,C} \\ \mathbf{r}_p^C \\ \mathbf{r}_\lambda^u + \mathbf{r}_\lambda^{u^i} \end{bmatrix}_k \quad (4.20)$$

The above system has several features that hamper an efficient three-dimensional implementation. The most severe problem is to find an appropriate, inf-sup stable Lagrange multiplier approximation and defining the proper location and structure of the Lagrange multiplier mesh. The problem is complicated by using hexahedral fluid elements, where complex intersection pattern emerge between given interface position and fluid element edges. Finally, the saddle point structure of the discrete system in Eq. (4.20) impedes the iterative solution of the equation system. For large scale flow simulations, such parallel, iterative solvers are mandatory. Hence, this approach has been given up in favor of an alternative weak formulation.

### 4.3.2 Cauchy stress based Lagrange multiplier

In the following, a stress based coupling approach is proposed, which has already been discussed for the Poisson problem in Section 3.3. The key point is that instead of using a traction vector field  $\lambda$  along the interface, an additional stress field  $\bar{\sigma}$ , that is approximated discontinuously between elements, is introduced as additional primary unknown.

The main purpose of the stress field  $\bar{\sigma}$  is its usage as Lagrange multiplier for the boundary constraints. Its test function  $\delta\bar{\sigma}$  weakly enforces the velocity constraint along the interface

$$(\delta\bar{\sigma} \cdot \mathbf{n}^f, \mathbf{u} - \mathbf{u}^i)_{\Gamma^i} \quad (4.21)$$

Hence,  $\delta\bar{\sigma} \cdot \mathbf{n}^f$  can be seen as the virtual traction force along the interface and has the same units as the virtual Lagrange multiplier field in Eq. (4.14c). The traction vector field  $\lambda$  along the interface is replaced with a symmetric tensor field multiplied by the fluid normal as  $\lambda = \bar{\sigma} \cdot \mathbf{n}^f$ .

For the six tensor components, a strain rate balance is added as

$$\bar{\gamma} - \gamma = \mathbf{0} \quad \text{in } \Omega^f \quad (4.22)$$

that is tested with the stress test function  $\delta\bar{\sigma}$  as

$$(\delta\bar{\sigma}, \bar{\gamma} - \gamma)_{\Omega^f} \quad (4.23)$$

The two strain rates  $\bar{\gamma}$  and  $\gamma$  are computed from the primary fluid unknowns

$$\bar{\gamma}(\bar{\sigma}, p) = \frac{1}{2\mu}(\bar{\sigma} + p\mathbf{I}) \quad (4.24)$$

$$\gamma(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) \quad (4.25)$$

This strain rate balance properly defines the additional stress in the fluid domain. The resulting task is therefore: find  $\mathbf{u}$ ,  $p$  and  $\tilde{\sigma}$  such that

$$\mathbf{r}_u(\mathbf{u}, p) = \mathbf{0} \quad \text{in } \Omega^f \quad (4.26a)$$

$$r_p(\mathbf{u}) = 0 \quad \text{in } \Omega^f \quad (4.26b)$$

$$\bar{\gamma} - \gamma = \mathbf{0} \quad \text{in } \Omega^f \quad (4.26c)$$

$$\mathbf{u} - \hat{\mathbf{u}}_D = \mathbf{0} \quad \text{in } \Gamma_D \quad (4.26d)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n}^f - \hat{h} = \mathbf{0} \quad \text{in } \Gamma_N \quad (4.26e)$$

$$\mathbf{u}^+ - \hat{\mathbf{u}}^i = \mathbf{0} \quad \text{in } \Gamma^i \quad (4.26f)$$

$$\tilde{\sigma} \cdot \mathbf{n}^f - \boldsymbol{\sigma} \cdot \mathbf{n}^f = \mathbf{0} \quad \text{in } \Gamma^i \quad (4.26g)$$

The velocity-dependent stress tensors  $\boldsymbol{\tau}$  and  $\boldsymbol{\sigma}$  in the fluid residual  $\mathbf{r}_u(\mathbf{u}, p)$  (Eq. (2.21a)) are computed as before

$$\boldsymbol{\tau} = 2\mu\gamma(\mathbf{u}) \quad (4.27)$$

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}(\mathbf{u}) \quad (4.28)$$

The combination of viscous stress  $\boldsymbol{\tau}$  and pressure  $p$  in the momentum equation Eq. (4.26a) is intentionally not replaced by the introduced Cauchy stress field  $\tilde{\sigma}$ . Hence, the stabilized fluid formulation remains unchanged and the additional stress approximation enters the fluid momentum equation only via boundary integrals resulting from partial integration of the stress term. Such a replacement, however, might become an option, if complex fluid materials are modeled as, e.g. in [Behr et al., 1993].

The weighted residual form is

$$0 = (\delta\mathbf{u}, \mathbf{r}_u(\mathbf{u}, p))_{\Omega^f} + (\delta\mathbf{u}, \boldsymbol{\sigma} \cdot \mathbf{n}^f - \hat{h})_{\Gamma_N^f} + (\delta\mathbf{u}, \boldsymbol{\sigma} \cdot \mathbf{n}^f - \tilde{\sigma} \cdot \mathbf{n}^f)_{\Gamma^i} \quad (4.29a)$$

$$0 = (\delta p, r_p(\mathbf{u}))_{\Omega^f} \quad (4.29b)$$

$$0 = -(\delta\tilde{\sigma} \cdot \mathbf{n}^f, \mathbf{u} - \mathbf{u}^i)_{\Gamma^i} \quad (4.29c)$$

$$0 = (\delta\mathbf{u}^i, \tilde{\sigma} \cdot \mathbf{n}^f)_{\Gamma^i} \quad (4.29d)$$

After integration by parts of the stress term within the fluid residual  $\mathbf{r}_u(\mathbf{u}, p)$ , the stabilized fluid system including the additional stress field becomes

$$0 = F_M(\delta\mathbf{u}, \mathbf{u}, p) - (\delta\mathbf{u}, \tilde{\sigma} \cdot \mathbf{n}^f)_{\Gamma^i} \quad (4.30a)$$

$$0 = F_C(\delta p, \mathbf{u}, p) \quad (4.30b)$$

$$0 = -(\delta\tilde{\sigma}, \bar{\gamma} - \gamma)_{\Omega^f} - (\delta\tilde{\sigma} \cdot \mathbf{n}^f, \mathbf{u} - \mathbf{u}^i)_{\Gamma^i} \quad (4.30c)$$

$$0 = (\delta\mathbf{u}^i, \tilde{\sigma} \cdot \mathbf{n}^f)_{\Gamma^i} \quad (4.30d)$$

Dirichlet and Neumann conditions away from the interface are employed as usual. In particular, Neumann conditions and fluid body forces are already included in  $F_M(\delta\mathbf{u}, \mathbf{u}, p)$  as defined in Eq. (2.29a).

In addition to the velocity and pressure discretization given in Eq. (2.27), the elementwise discontinuous stresses are approximated as

$$\bar{\sigma}^h(x) = \sum_L \phi_L^{\bar{\sigma}}(x) \bar{\sigma}_L \quad (4.31a)$$

$$\delta\bar{\sigma}^h(x) = \sum_L \phi_L^{\bar{\sigma}}(x) \delta\bar{\sigma}_L \quad (4.31b)$$

The stress field is enriched by the same enrichment functions as for nodal velocities and pressures, however, elementwise discontinuous shape functions  $N_L^{\bar{\sigma},e}(x)$  are applied. The stress approximation is therefore  $C^{-1}$ -continuous at inter-element boundaries. For the examples given at the end of this chapter, velocity, pressure and stress use the same order for their shape functions. Adopting the notation from [Gresho and Sani, 2000], the approximations for velocity, pressure, and stress are termed as  $Q_1 Q_1 Q_{-1}$  for trilinear,  $Q_2^{(20)} Q_2^{(20)} Q_{-2}^{(20)}$  for quadratic serendipity and  $Q_2 Q_2 Q_{-2}$  for fully triquadratic shape functions. Likewise, the approximation for linear and quadratic tetrahedrons are defined as  $P_1 P_1 P_{-1}$  and  $P_2 P_2 P_{-2}$ , respectively. Other element shapes like pyramids and wedges should approximate the stress in an analogous manner, however, such elements have not been tested and used in this work. The approximation spaces are gained from experience with the Hellinger-Reissner (HR)-principles for elastic materials and from the numerical experiments presented in Section 4.5.2. A rigorous mathematical analysis is still missing.

The discrete system to solve is therefore

$$\delta\mathbf{u} [\mathbf{r}_u(\mathbf{u}, \mathbf{p}) + \mathbf{r}_u^{\bar{\sigma}}(\bar{\sigma})] = \mathbf{F}_M(\delta\mathbf{u}^h, \mathbf{u}^h, p^h) - (\delta\mathbf{u}^h, \bar{\sigma}^h \cdot \mathbf{n}^f)_{\Gamma^i} \quad (4.32a)$$

$$\delta\mathbf{p} \mathbf{r}_p(\mathbf{u}, \mathbf{p}) = \mathbf{F}_C(\delta p^h, \mathbf{u}^h, p^h) \quad (4.32b)$$

$$\begin{aligned} \delta\bar{\sigma} \mathbf{r}_{\bar{\sigma}}(\mathbf{u}, \mathbf{p}, \bar{\sigma}, \mathbf{u}^i) &= -(\delta\bar{\sigma}^h, \bar{\gamma}(\bar{\sigma}^h, p^h) - \gamma(\mathbf{u}^h))_{\Omega^f} \\ &\quad - (\delta\bar{\sigma}^h \cdot \mathbf{n}^f, \mathbf{u}^h - \mathbf{u}^{i,h})_{\Gamma^i} \end{aligned} \quad (4.32c)$$

$$\delta\mathbf{u}^i \mathbf{r}_{u^i}^{\bar{\sigma}}(\bar{\sigma}) = (\delta\mathbf{u}^{i,h}, \bar{\sigma}^h \cdot \mathbf{n}^f)_{\Gamma^i} \quad (4.32d)$$

Velocity and pressure nodal unknowns are denoted as  $\mathbf{u}$  and  $\mathbf{p}$  as before; The newly introduced element stress unknowns are denoted by  $\bar{\sigma}$ . The global linearized discrete system including the additional stress unknowns is given as

$$\left[ \begin{array}{ccc|c|c} \mathbf{F}_{uu} & \mathbf{F}_{up} & \mathbf{G}_{u\bar{\sigma}} & \mathbf{0} & \\ \mathbf{F}_{pu} & \mathbf{F}_{pp} & \mathbf{0} & \mathbf{0} & \\ \mathbf{K}_{\bar{\sigma}u} + \mathbf{G}_{\bar{\sigma}u} & \mathbf{K}_{\bar{\sigma}p} & \mathbf{K}_{\bar{\sigma}\bar{\sigma}} & \mathbf{G}_{\bar{\sigma}u^i} & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{G}_{u^i\bar{\sigma}} & \mathbf{0} & \end{array} \right]_k \left[ \begin{array}{c} \Delta\mathbf{u} \\ \Delta\mathbf{p} \\ \Delta\bar{\sigma} \\ \Delta\mathbf{u}^i \end{array} \right] = - \left[ \begin{array}{c} \mathbf{r}_u + \mathbf{r}_u^{\bar{\sigma}} \\ \mathbf{r}_p \\ \mathbf{r}_{\bar{\sigma}} \\ \mathbf{r}_{u^i}^{\bar{\sigma}} \end{array} \right]_k \quad (4.33)$$

The submatrices  $\mathbf{F}_{uu}$ ,  $\mathbf{F}_{up}$ ,  $\mathbf{F}_{pu}$  and  $\mathbf{F}_{pp}$  denote the discrete fluid system as before. The remaining submatrices result from the linearization of the boundary

integral terms

$$\delta \mathbf{u} \mathbf{G}_{u\bar{\sigma}} = - \frac{\partial (\delta \mathbf{u}^h, \bar{\sigma}^h \cdot \mathbf{n}^f)_{\Gamma^i}}{\partial \bar{\sigma}} \quad (4.34)$$

$$\delta \bar{\sigma} \mathbf{G}_{\bar{\sigma}u} = - \frac{\partial (\delta \bar{\sigma}^h \cdot \mathbf{n}^f, \mathbf{u}^h)_{\Gamma^i}}{\partial u} \quad (4.35)$$

$$\delta \bar{\sigma} \mathbf{G}_{\bar{\sigma}u^i} = + \frac{\partial (\delta \bar{\sigma}^h \cdot \mathbf{n}^f, \mathbf{u}^{i,h})_{\Gamma^i}}{\partial u^i} \quad (4.36)$$

$$\delta \mathbf{u}^i \mathbf{G}_{u^i\bar{\sigma}} = + \frac{\partial (\delta \mathbf{u}^{i,h}, \bar{\sigma}^h \cdot \mathbf{n}^f)_{\Gamma^i}}{\partial \bar{\sigma}} \quad (4.37)$$

and domain integral terms

$$\delta \bar{\sigma} \mathbf{K}_{\bar{\sigma}u} = + \frac{\partial (\delta \bar{\sigma}^h, \gamma(\mathbf{u}^h))_{\Omega^f}}{\partial u} \quad (4.38)$$

$$\delta \bar{\sigma} \mathbf{K}_{\bar{\sigma}p} = - \frac{\partial (\delta \bar{\sigma}^h, \bar{\gamma}(p^h, \bar{\sigma}^h))_{\Omega^f}}{\partial p} \quad (4.39)$$

$$\delta \bar{\sigma} \mathbf{K}_{\bar{\sigma}\bar{\sigma}} = - \frac{\partial (\delta \bar{\sigma}^h, \bar{\gamma}(p^h, \bar{\sigma}^h))_{\Omega^f}}{\partial \bar{\sigma}} \quad (4.40)$$

Note that the boundary integrals are already linear relations. For the employed Newtonian material model with constant and uniform viscosity and density, the domain integrals for the strain rates are linear as well.

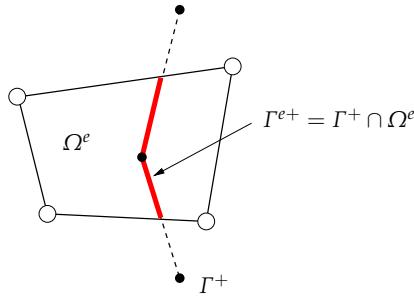
As introduced during the stress discretization, the stress unknowns are element unknowns, meaning that their shape functions are discontinuous at inter-element boundaries. This allows to write separately for each element  $e$ :

$$\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^e \Delta \bar{\sigma}_e + (\mathbf{K}_{\bar{\sigma}u}^e + \mathbf{G}_{\bar{\sigma}u}^e) \Delta \mathbf{u}_e + \mathbf{K}_{\bar{\sigma}p}^e \Delta p_e + \mathbf{G}_{\bar{\sigma}u^i}^e \Delta \mathbf{u}_e^i = -\mathbf{r}_{\bar{\sigma}}^e \quad (4.41)$$

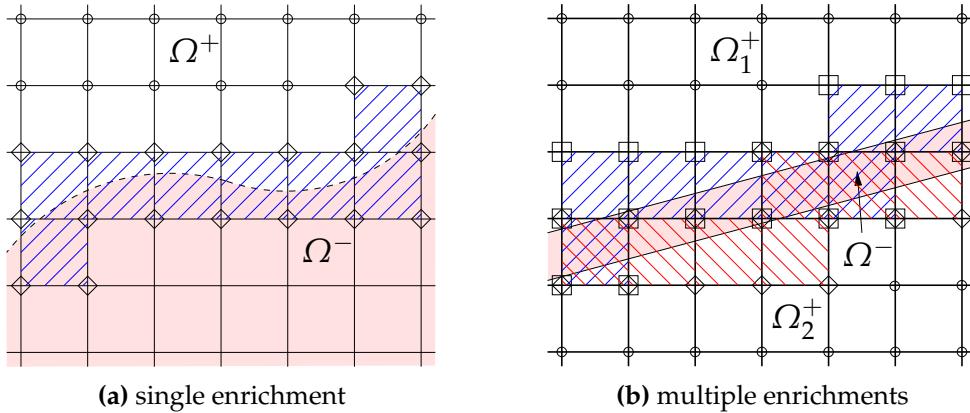
Here,  $\Delta \mathbf{u}_e$ ,  $\Delta p_e$  denote the velocity and pressure nodal unknowns belonging to the current element  $e$  and  $\Delta \bar{\sigma}_e$  the element unknowns for the stress approximation. The vector  $\Delta \mathbf{u}_e^i$  contains all interface velocities belonging to interface elements, which intersect the current fluid element  $e$ . For one fluid element  $e$ , this setup is depicted in Figure 4.5. The boundary integrals are integrated separately for each fluid element  $e$  using boundary integration cells.

The introduced stress field couples into the fluid and interface momentum equation only via the boundary integrals along  $\Gamma^i$ , namely via  $\mathbf{G}_{u\bar{\sigma}}$  and  $\mathbf{G}_{u^i\bar{\sigma}}$ , respectively. Hence, although the stress field is properly defined in all elements, the stress unknowns influence the velocity and pressure solution only in intersected elements. This observation allows to conclude that (i) the element stress equation can be omitted for non-intersected elements; and (ii) the element stress unknowns can be condensed on the element level.

The first conclusion leaves element stress unknowns only in intersected elements as shown in Figure 4.6. Since void enrichments are exclusively used,



**Figure 4.5:** Definition of element boundary for boundary integrals.



**Figure 4.6:** Enrichment for element stresses: stress unknowns are only present in intersected elements. For multiple interfaces, multiple stress enrichments indicated by multiple fill patterns are required as shown on the right.

the stress is approximated only by void enrichment functions and standard or continuous enrichments are unnecessary.

The second conclusion allows the following condensation process for intersected elements. The stress increments for Newton step  $k$  of element  $e$  can be expressed as

$$\Delta\bar{\sigma}_e = (\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^e)^{-1}(-r_{\bar{\sigma}}^e - (\mathbf{K}_{\bar{\sigma}u}^e + \mathbf{G}_{\bar{\sigma}u}^e)\Delta\mathbf{u}_e - \mathbf{K}_{\bar{\sigma}p}^e\Delta\mathbf{p}_e - \mathbf{G}_{\bar{\sigma}u^i}^e\Delta\mathbf{u}_e^i) \quad (4.42)$$

Let the following element coupling matrices  $\mathbf{C}^e$  for the fluid velocity unknowns be defined as

$$\mathbf{C}_{uu}^e = -\mathbf{G}_{u\bar{\sigma}}^e(\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^e)^{-1}(\mathbf{K}_{\bar{\sigma}u}^e + \mathbf{G}_{\bar{\sigma}u}^e) \quad (4.43)$$

$$\mathbf{C}_{up}^e = -\mathbf{G}_{u\bar{\sigma}}^e(\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^e)^{-1}\mathbf{K}_{\bar{\sigma}p}^e \quad (4.44)$$

$$\mathbf{C}_{uu^i}^e = -\mathbf{G}_{u\bar{\sigma}}^e(\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^e)^{-1}\mathbf{G}_{\bar{\sigma}u^i}^e \quad (4.45)$$

and, similarly, for the interface velocity unknowns as

$$\mathbf{C}_{u^i u}^e = -\mathbf{G}_{u^i \bar{\sigma}}^e (\mathbf{K}_{\bar{\sigma} \bar{\sigma}}^e)^{-1} (\mathbf{K}_{\bar{\sigma} u}^e + \mathbf{G}_{\bar{\sigma} u}^e) \quad (4.46)$$

$$\mathbf{C}_{u^i p}^e = -\mathbf{G}_{u^i \bar{\sigma}}^e (\mathbf{K}_{\bar{\sigma} \bar{\sigma}}^e)^{-1} \mathbf{K}_{\bar{\sigma} p}^e \quad (4.47)$$

$$\mathbf{C}_{u^i u^i}^e = -\mathbf{G}_{u^i \bar{\sigma}}^e (\mathbf{K}_{\bar{\sigma} \bar{\sigma}}^e)^{-1} \mathbf{G}_{\bar{\sigma} u^i}^e \quad (4.48)$$

Let further the residual terms  $\dot{r}_u^e$  and  $\dot{r}_{u^i}^e$  be defined as

$$\dot{r}_u^e = -\mathbf{G}_{u \bar{\sigma}}^e (\mathbf{K}_{\bar{\sigma} \bar{\sigma}}^e)^{-1} r_{\bar{\sigma}}^e \quad (4.49)$$

$$\dot{r}_{u^i}^e = -\mathbf{G}_{u^i \bar{\sigma}}^e (\mathbf{K}_{\bar{\sigma} \bar{\sigma}}^e)^{-1} r_{\bar{\sigma}}^e \quad (4.50)$$

With these definitions, the condensed global system can be assembled ( $\mathbf{A}_e$ ) from element entries as

$$\mathbf{A}_e \begin{bmatrix} F_{uu}^e + C_{uu}^e & F_{up}^e + C_{up}^e & C_{uu^i}^e \\ F_{pu}^e & F_{pp}^e & \mathbf{0} \\ C_{u^i u}^e & C_{u^i p}^e & C_{u^i u^i}^e \end{bmatrix}_k \mathbf{A}_e \begin{bmatrix} \Delta u_e \\ \Delta p_e \\ \Delta u_e^i \end{bmatrix} = -\mathbf{A}_e \begin{bmatrix} r_u^e + r_{u^i}^{\bar{\sigma}, e} + \dot{r}_u^e \\ r_p^e \\ r_{u^i}^{\bar{\sigma}, e} + \dot{r}_{u^i}^e \end{bmatrix}_k \quad (4.51)$$

From an implementation point of view, it can be useful to assemble the coupling matrices  $\mathbf{C}^e$  and the additional residual terms  $\dot{r}^e$  separately into global sparse matrices  $\mathbf{C}$  and residual vectors, respectively. The global system will have modified entries only for fluid nodal unknowns belonging to intersected elements. Splitting into constrained (set  $\mathcal{C}$ ) and unconstrained fluid unknowns (set  $\mathcal{S}$ ), the global system is written as

$$\begin{bmatrix} F_{uu}^{SS} & F_{up}^{SS} & F_{uu}^{SC} & F_{up}^{SC} & \mathbf{0} \\ F_{pu}^{SS} & F_{pp}^{SS} & F_{pu}^{SC} & F_{pp}^{SC} & \mathbf{0} \\ F_{uu}^{CS} & F_{up}^{CS} & F_{uu}^{CC} + C_{uu}^{CC} & F_{up}^{CC} + C_{up}^{CC} & C_{uu^i}^{CC} \\ F_{pu}^{CS} & F_{pp}^{CS} & F_{pu}^{CC} & F_{pp}^{CC} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & C_{u^i u} & C_{u^i p} & C_{u^i u^i} \end{bmatrix}_k \begin{bmatrix} \Delta u^S \\ \Delta p^S \\ \Delta u^C \\ \Delta p^C \\ \Delta u^i \end{bmatrix} = -\begin{bmatrix} r_u^S \\ r_p^S \\ r_u^C + r_{u^i}^{\bar{\sigma}, C} + \dot{r}_u^C \\ r_p^C \\ r_{u^i}^{\bar{\sigma}, i} + \dot{r}_{u^i}^i \end{bmatrix}_k \quad (4.52)$$

If all interface velocities are given as Dirichlet conditions, no interface velocity increments are computed and the fluid system reduces to

$$\begin{bmatrix} F_{uu}^{SS} & F_{up}^{SS} & F_{uu}^{SC} & F_{up}^{SC} \\ F_{pu}^{SS} & F_{pp}^{SS} & F_{pu}^{SC} & F_{pp}^{SC} \\ F_{uu}^{CS} & F_{up}^{CS} & F_{uu}^{CC} + C_{uu}^{CC} & F_{up}^{CC} + C_{up}^{CC} \\ F_{pu}^{CS} & F_{pp}^{CS} & F_{pu}^{CC} & F_{pp}^{CC} \end{bmatrix}_k \begin{bmatrix} \Delta u^S \\ \Delta p^S \\ \Delta u^C \\ \Delta p^C \end{bmatrix} = -\begin{bmatrix} r_u^S \\ r_p^S \\ r_u^C + r_{u^i}^{\bar{\sigma}, C} + \dot{r}_u^C \\ r_p^C \end{bmatrix}_k \quad (4.53)$$

Without the split into constrained and unconstrained degrees of freedom, the short notation for Eq. (4.53) is

$$\begin{bmatrix} F_{uu} + C_{uu} & F_{up} + C_{up} \\ F_{pu} & F_{pp} \end{bmatrix}_k \begin{bmatrix} \Delta u \\ \Delta p \end{bmatrix} = -\begin{bmatrix} r_u + r_{u^i}^{\bar{\sigma}} + \dot{r}_u \\ r_p \end{bmatrix}_k \quad (4.54)$$

---

**Algorithm 4.1** Build global tangent stiffness matrix and residual

---

```

for each fluid element  $e$  do
    if element intersected then
        if stress condensation then
            extract nodal values from global solution vector
            recover element values using old solution and increment
        else
            extract nodal and element values from global solution vector
        end if
        build element matrix and residual
        if stress condensation then
            condense stress
            store stress-related element matrices at element
            assemble condensed system into global system
        else
            assemble uncondensed system into global system
        end if
    else
        extract nodal values from global solution vector
        perform domain integration for element matrix and residual
        assemble element system into global system
    end if
end for

```

---

Using an incremental Newton-Raphson iteration, the condensed element stress requires a special update procedure. If system Eq. (4.54) has been solved, the updated solution  $[u^T \ p^T]_{k+1}$  is computed as given in Eq. (2.42). For the element stress update, the element stress increment is computed using Eq. (4.42) and the resulting stress increment is added to the stress solution from iteration step  $k$  to obtain the improved element stress unknowns

$$\bar{\sigma}_{k+1} = \bar{\sigma}_k + \Delta\bar{\sigma}(\Delta u, \Delta p) \quad (4.55)$$

For this purpose, the element matrices required in Eq. (4.42) are stored at intersected elements during the Newton-Raphson iteration. Algorithm 4.1 summarizes the evaluation of element integrals, when element stress condensation is applied. For debugging purpose, the element stress in intersected elements can be left in the global system. This is indicated by an additional check, whether condensation should be performed or not. For the same reason, the element stress can be applied to all elements instead of only to the intersected elements. While both debugging features may influence the iterative solution of the non-linear fluid system, they do not affect the converged solution as discussed above and as shown in the example section.

### 4.3.3 Viscous stress and pressure based Lagrange multipliers

In deviation from the previous hybrid approach, it is also possible to introduce four primary unknown fields, namely  $\mathbf{u}$ ,  $p$ ,  $\bar{\tau}$  and  $\bar{p}$ . This approach applies the split of the Cauchy stress into pressure and deviatoric stress also to the element based Cauchy stress. The additional pressure  $\bar{p}$  is then used to enforce that the trace of the deviatoric stress  $\bar{\tau}$  or the derived strain rate  $\bar{\gamma}$  is zero, which is analogue to the incompressibility constraint for the velocity field. The strong form for this four field problem is defined as

$$\mathbf{r}_u(\mathbf{u}, p) = \mathbf{0} \quad \text{in } \Omega \quad (4.56\text{a})$$

$$r_p(\mathbf{u}) = 0 \quad \text{in } \Omega \quad (4.56\text{b})$$

$$\bar{\gamma} - \gamma = \mathbf{0} \quad \text{in } \Omega \quad (4.56\text{c})$$

$$-\bar{p} + p = 0 \quad \text{in } \Omega \quad (4.56\text{d})$$

$$\text{tr}(\bar{\gamma}) = 0 \quad \text{in } \Omega \quad (4.56\text{e})$$

$$\mathbf{u} - \hat{\mathbf{u}} = \mathbf{0} \quad \text{in } \Gamma^i \quad (4.56\text{f})$$

$$\underline{\sigma} \cdot \mathbf{n}^f - \hat{h} = \mathbf{0} \quad \text{in } \Gamma_N \quad (4.56\text{g})$$

$$\underline{\sigma} \cdot \mathbf{n}^f - \bar{\sigma} \cdot \mathbf{n}^f = \mathbf{0} \quad \text{in } \Gamma^i \quad (4.56\text{h})$$

Velocity and pressure based values are computed as before, namely as

$$\underline{\sigma}(\mathbf{u}, p) = -p \mathbf{I} + \underline{\tau}(\mathbf{u}) \quad (4.57)$$

$$\underline{\tau}(\mathbf{u}) = 2\mu \underline{\gamma}(\mathbf{u}) \quad (4.58)$$

$$\underline{\gamma}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \quad (4.59)$$

The element stress and element pressure based Cauchy stress  $\bar{\sigma}$  and strain rate  $\bar{\gamma}$  are computed using

$$\bar{\sigma}(\bar{\tau}, \bar{p}) = -\bar{p} \mathbf{I} + \bar{\tau} \quad (4.60)$$

$$\bar{\gamma}(\bar{\tau}) = \frac{1}{2\mu} \bar{\tau} \quad (4.61)$$

The stabilized fluid system after integration by parts including the additional stress field as given as

$$0 = F_M(\delta \mathbf{u}, \mathbf{u}, p) - (\delta \mathbf{u}, -\bar{p} \mathbf{n}^f + \bar{\tau} \cdot \mathbf{n}^f)_{\Gamma^i} \quad (4.62\text{a})$$

$$0 = F_C(\delta p, \mathbf{u}, p) \quad (4.62\text{b})$$

$$0 = -(\delta \bar{\tau}, \bar{\gamma} - \gamma)_{\Omega^f} + (\text{tr}(\delta \bar{\gamma}), \bar{p} - p)_{\Omega} - (\delta \bar{\tau} \cdot \mathbf{n}^f, \mathbf{u} - \mathbf{u}^i)_{\Gamma^i} \quad (4.62\text{c})$$

$$0 = (\delta \bar{p}, \text{tr}(\bar{\gamma}))_{\Omega^f} + (\delta \bar{p} \mathbf{n}^f, \mathbf{u} - \mathbf{u}^i)_{\Gamma^i} \quad (4.62\text{d})$$

$$0 = (\delta \mathbf{u}^i, -\bar{p} \mathbf{n}^f + \bar{\tau} \cdot \mathbf{n}^f)_{\Gamma^i} \quad (4.62\text{e})$$

In the previous, Cauchy stress based formulation, the Cauchy stress represents the full stress tensor. Due to the split into pressure and deviatoric viscous stress, it is now necessary to enforce that the trace of the strain rate  $\text{tr}(\bar{\gamma})$  is zero. This is equivalent to the incompressibility constraint for the velocity field. This additional constraint is enforced by the element pressure. Both viscous stress and pressure are required for the surface integrals to obtain the complete surface traction vector.

The complete fluid-interface system including the interface constraint becomes

$$\begin{bmatrix} F_{uu} & F_{up} & G_{u\bar{\tau}} & G_{u\bar{p}} & 0 \\ F_{pu} & F_{pp} & 0 & 0 & 0 \\ K_{\bar{\tau}u} + G_{\bar{\tau}u} & K_{\bar{\tau}p} & K_{\bar{\tau}\bar{\tau}} & K_{\bar{\tau}\bar{p}} & G_{\bar{\tau}u^i} \\ G_{\bar{p}u} & 0 & K_{\bar{p}\bar{\tau}} & 0 & G_{\bar{p}u^i} \\ 0 & 0 & G_{u^i\bar{\tau}} & G_{u^i\bar{p}} & 0 \end{bmatrix}_k \begin{bmatrix} \Delta u \\ \Delta p \\ \Delta \bar{\tau} \\ \Delta \bar{p} \\ \Delta u^i \end{bmatrix} = - \begin{bmatrix} r_u + r_u^{\bar{\tau}, \bar{p}} \\ r_p \\ r_{\bar{\tau}} \\ r_{\bar{p}} \\ r_{u^i}^{\bar{\tau}, \bar{p}} \end{bmatrix}_k \quad (4.63)$$

The matrix structure is equivalent to the previous approach. Element pressures and element viscous stresses are approximated elementwise discontinuously in exactly the same way as the Cauchy element stress in the previous section. The approximations for velocity, continuous pressure, viscous stress and element pressure are  $Q_1 Q_1 Q_{-1} Q_{-1}$  for trilinear,  $Q_2^{(20)} Q_2^{(20)} Q_{-2}^{(20)} Q_{-2}^{(20)}$  for quadratic serendipity and  $Q_2 Q_2 Q_{-2} Q_{-2}$  for full triquadratic shape functions. Likewise, the approximation for linear and quadratic tetrahedrons are defined as  $P_1 P_1 P_{-1} P_{-1}$  and  $P_2 P_2 P_{-2} P_{-2}$ , respectively.

Additional pressure and viscous stress couple into the momentum equation only via boundary integrals along  $\Gamma^i$ . Hence, they can be omitted for non-intersected elements without affecting the velocity and nodal pressure solution. In addition, it is again possible to condense the element variables at the element level

$$\begin{bmatrix} \Delta \bar{\tau} \\ \Delta \bar{p} \end{bmatrix} = \begin{bmatrix} K_{\bar{\tau}\bar{\tau}} & K_{\bar{\tau}p} \\ K_{\bar{p}\bar{\tau}} & 0 \end{bmatrix}^{-1} \left( - \begin{bmatrix} r_{\bar{\tau}} \\ r_{\bar{p}} \end{bmatrix} - \begin{bmatrix} K_{\bar{\tau}u} + G_{\bar{\tau}u} \\ G_{\bar{p}u} \end{bmatrix} \Delta u_e - \begin{bmatrix} K_{\bar{\tau}p} \\ 0 \end{bmatrix} \Delta p_e - \begin{bmatrix} G_{\bar{\tau}u^i} \\ G_{\bar{p}u^i} \end{bmatrix} \Delta u_e^i \right) \quad (4.64)$$

Inversion of the matrix is performed using a direct solver (Amesos UMF-PACK, [Heroux and Willenbring, 2003]) on the element level, where the zero diagonal term does not pose particular problems. The algorithmic implementation is analogue to Algorithm 4.1. The condensed system has shown — at least for low Reynolds number cases considered in this thesis — the same optimal convergence properties as the Cauchy stress Lagrange multiplier.

In summary, the alternative approach has shown no particular advantages or disadvantages compared to the first approach. Therefore the Cauchy stress approach is used for all three-dimensional examples, since less element stress

unknowns are present making the Cauchy stress approach less costly. However, if a mathematical analysis of the stress based Lagrange multiplier is undertaken, the pressure-stress approach should receive attention, too.

## 4.4 Implementation and Algorithms

In the following, an overview on implementation aspects is given. The structure of this section loosely follows the tasks typically encountered, when moving interfaces with given displacement are encountered: (i) generation of interface mesh and integration cells, (ii) establishing DOF-distributions and enrichment, (iii) numerical integration of element matrices, (iv) solution of the assembled fluid system, (v) visualization of the computed results

The first two points represent the setup phase. Its steps are detailed in Algorithm 4.2. For transient problems with fixed boundary positions as consid-

---

### Algorithm 4.2 XFEM setup phase

---

```

locate intersected elements using parallel search trees
generate boundary and domain integration cells
identify required enrichments and setup DOFs
allocate parallel vectors and matrices matching the DOF distribution
if interface moves then
    extrapolate DOFs from old to new DOF distribution → Algorithm 5.1
end if
```

---

ered in this chapter, this setup is performed before the fluid time loop starts. Within the time loop, the nonlinear solution process is similar to the standard FEM. The only difference is the modified element integration as indicated in Algorithm 4.3.

The implementation is done in the multiphysics code Baci that is developed at the Institute for Computational Mechanics at Technische Universität München. The parallel features are largely based on the Open-Source library Trilinos [Heroux et al., 2005] on top of which element libraries and a parallel mesh and unknown management are implemented.

### 4.4.1 Interface localization

One of the basic questions that has to be answered during XFEM computations is how a position  $x$  is located relatively to the interfaces  $\Gamma_i^i$ . The main application is the evaluation of the enrichment function  $\bar{\psi}(x)$  for each interface.

---

**Algorithm 4.3** Newton loop

---

```

for each newton step  $k$  do
    apply Dirichlet conditions to solution vector  $[\mathbf{u} \; \mathbf{p}]_k^T$ 
    compute residual and system matrix Algorithm 4.1
    check convergence based on some norm of residual
    if converged then
        stop Newton loop
    end if
    apply Dirichlet conditions to residual and matrix
    solve for increment  $[\Delta\mathbf{u} \; \Delta\mathbf{p}]_k^T$ 
    update solution vector
    store old increment for stress update in elements
end for

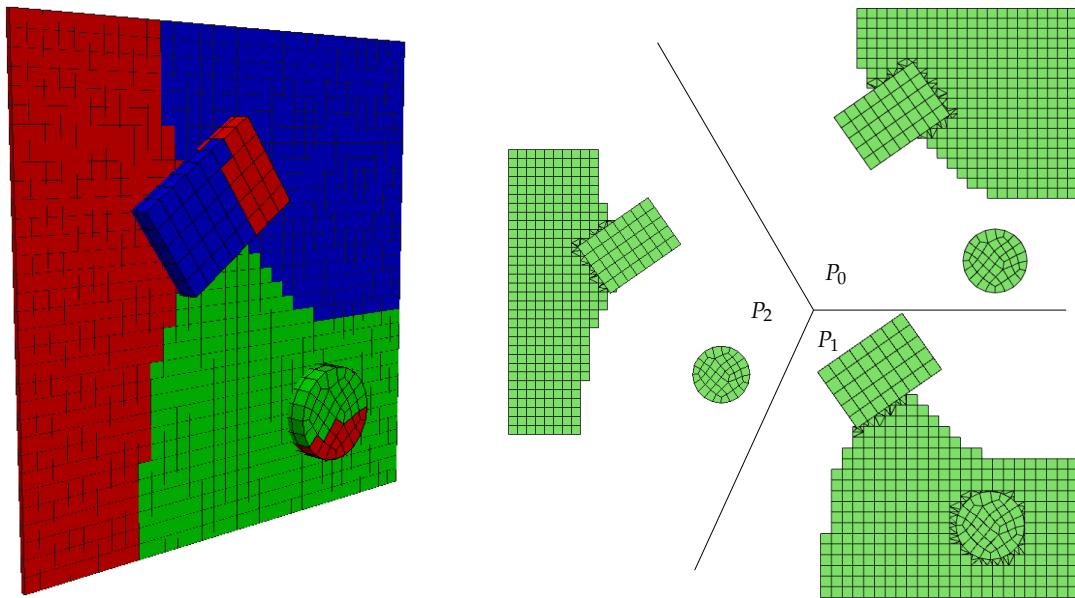
```

---

The first task is to identify the position of the interface. In the proposed FSI scheme, the structure surface serves as a Lagrangian interface marker. Hence, no other means of locating the interface is required. This is in contrast to applications like two phase flow or crack propagation, where the zero contour of the level-set is the only information, where the interface is located. See references [Belytschko et al., 2001; Chessa and Belytschko, 2003] for examples of each application. The second task is the definition of inside and outside or in front and behind the interface. For the FSI scheme, the surface normal of the structure surface serves as the reference. For level-set functions, the sign of the function identifies the regions.

For the present implementation, a parallel search tree is used to answer introductory question. For partitioned FSI simulations, the three-dimensional fluid and structure meshes are distributed independently on the available processors. This is exemplary shown for three processors on the left in Figure 4.7. A cut through the fluid mesh reveals the partitioning into three processors. The structure surface mesh is mirrored on all processors, which as depicted on the right in Figure 4.7. Note that for the given three-dimensional example, the structure surface is closed. The interior volume elements are not distributed. The mirroring happens only once at the beginning of the simulation, assuming that the structure surface mesh remains unchanged. For subsequent time-steps, only the surface displacement has to be distributed to all processors, which is efficiently done using the parallel vectors included in the Trilinos library. In the case of structural mesh changes, the surface mesh needs to be distributed again.

In summary, a level-set function is not required for the proposed FSI approach. Nevertheless, the level-set function can be used in addition to Lagrangian interfaces as proposed in [Legay et al., 2006]. Here the level-set is

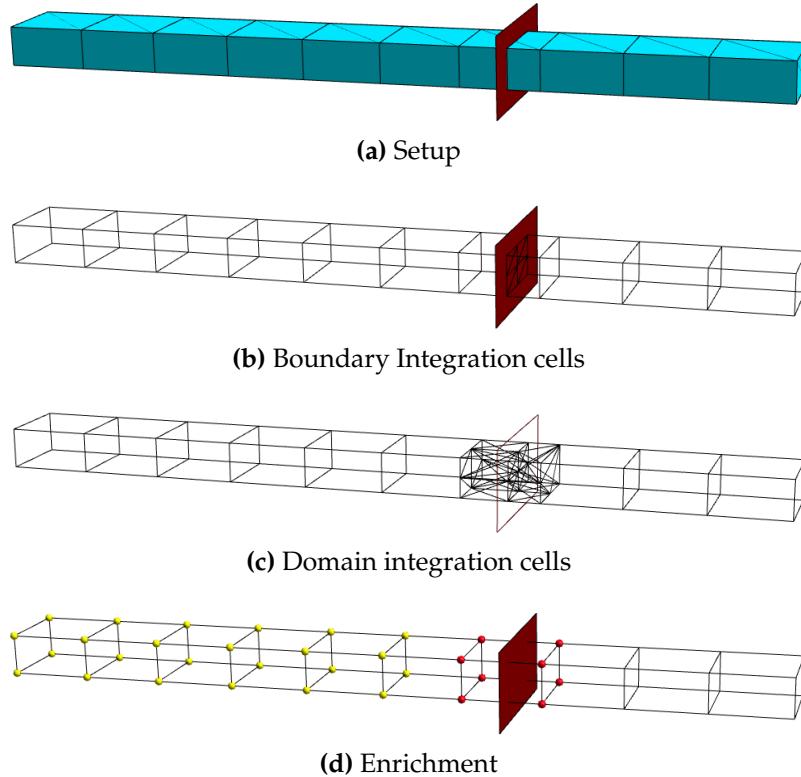


**Figure 4.7:** Parallel distribution of 3d fluid mesh and FSI surface: The redundant interface mesh allows completely parallel integration cell generation and enrichment function evaluation.

generated based on the most recent interface position and is subsequently used for defining separate regions. For parallel simulations, a practical property is that level-set functions are present as discrete field on all processors and thereby it is possible to identify the relative position with respect to an interface even if the interface itself is located on another processor. However, the complexity of managing level-sets increases, if multiple interface or continua are present. In addition, open boundaries requires additional level-sets to locate the edges of open boundaries. Whether parallel search trees or parallel level-sets are more effective in terms of memory, parallel re-initialization and parallel communication, remains an open question. In addition, load balancing taking the nodal connectivity and the expensive integration of intersected elements into account (see Section 3.4) has not been optimized, yet.

#### 4.4.2 Integration cell generation

With the interface mesh and its updated position in time, the subsequent generation of integration cells is performed completely in parallel. The intersection processes is supported by a local search tree on each processor that covers all interface meshes plus the local fluid mesh. With the help of the search tree, intersection candidates are found quickly and only a small number of elements is really tested for possible intersection with the fluid-structure interface [Mayer et al., 2009].



**Figure 4.8:** Fundamental steps for generating integration cells and enrichment

The intersection process consists of two steps, which is illustrated in Figure 4.8. The setup for the isostatic pressure example given in Section 4.5.2 is used as an example. The fluid mesh and the intersecting quad4 boundary element are shown in Figure 4.8a. In the first step, the boundary integration cells are generated such that the cells are aligned with surface mesh elements and that cell edges respect both the surface mesh element edges and the fluid element edges and faces (Figure 4.8b). The positions of the cell corners are stored in fluid element coordinates  $\xi^\Omega$ . In the second step, domain integration cells are generated. The generation is based on a constrained Delaunay tetrahedralization, which is implemented using the external library TetGen [Si, 2010a,b]. The domain for tetrahedralization is the fluid element domain in element coordinates  $\xi^\Omega$ . The boundary cells represent the constraining surface in the constrained Delaunay tetrahedralization (CDT3) such that no domain cell is intersected by a boundary cell. The resulting tetrahedral domain cells are shown in Figure 4.8c. For a minimal number of integration points, as few integration cells as possible are generated (see Section 3.4). If higher order integration cells are required, the tetrahedralization is first performed using linear boundary and domain integration cells. Higher order nodes along tetrahedron edges are then moved onto the curved interface  $\Gamma^i$  as described in [Mayer et al., 2009].

In terms of computation time, the generation of both linear and quadratic integration cells is much less costly than the numerical integration of the same element. The critical part is the robustness of the implementation of the subdivision scheme. The complex implementation requires rigorous testing, since any complex intersection situation will occur with increasing likelihood, if the number of nodes and elements increases.

#### 4.4.3 Enrichment and unknown management

In the three-dimensional FE fluid formulation as described in Section 2.1, there are four unknowns per node (three velocity components and the nodal pressure) and no element unknowns at all times. Without adaptive mesh refinement, the size of global and element vectors as well as the parallel distribution do not change. In absence of adaptivity, the mesh is written once for output and subsequently, only the solution vectors need to be written to disk.

In contrast, the XFEM formulation with multiple interfaces introduces more than the usual unknowns on some nodes or no unknowns on other nodes. In addition, enriched element unknowns have to be treated that are present in the stress based Lagrange multiplier formulation. Hence, the introduction of fully and partially enriched fields requires to keep track of the meaning of each node's unknowns over time.

For this purpose, a special identifier termed *DofKey* is introduced. The basic idea is that the tuple (*NodeId*,*Field*,*EnrType*,*Label*) uniquely defines the meaning of a specific unknown. The *NodeId* is a global identifier that occurs exactly once in the parallel simulation, i.e. it is unique across all processors. The *Field* identifies the involved physical scalar quantities like temperature, pressure and a velocity components. It can also contain tensor components like  $\tau_{xy}$ . Each Field is considered as a scalar field. Their individual meaning, for example as a components of a vector or second order tensor field, will only be manifested when evaluating the weak form. The *EnrType* defines the kind of enrichment. As introduced in Section 4.2.2, standard as well as interface and FSI void enrichments are required. The last identifier, the *Label*, enumerates the involved interfaces in the simulation, if thin structures are considered.

This concept is expressed in C++ using three classes as

$$\begin{array}{c}
 \left( \text{NodeId}, \left( \text{Field}, \underbrace{\left( \text{EnrType}, \text{Label} \right)}_{\text{Enrichment}} \right) \right) \\
 \underbrace{\qquad\qquad\qquad}_{\text{FieldEnr}} \\
 \underbrace{\qquad\qquad\qquad}_{\text{DofKey}}
 \end{array}$$

NodeId and Label are implemented using unsigned integers, while Field and EnrType are implemented using C++ enumerates types (enums). Each class implements the relational operators ‘equal’, ‘not equal’, ‘less than’, and ‘more than’. This allows to use the classes within vectors, sets and maps as defined in the C++ standard template library (STL). Subsequently, mathematical definitions on DofKey sets can be readily expressed using STL sets and maps. The same DofKey class is used to manage element unknowns for intersected elements.

In addition, the ability to serialize each DofKey object and distribute the object on parallel machines is implemented. Such parallel communication is necessary, if interfaces cross processor boundaries. Then the implementation of the enrichment processes described in Section 4.2.2 requires additional parallel communication to obtain matching enrichment situations on each processor.

This concept of handling nodal and element unknowns has proven to be applicable to all encountered complex enrichment situations. Although it is difficult to measure source code readability and readability depends also on personal taste, the resulting source code is easier to comprehend, as if the same concept would be expressed with plain integers and arrays instead of classes. Similar concepts have for example been described in [Tabarraei and Sukumar, 2008].

#### 4.4.4 Visualization

For visualization of complex flows, optimized sequential or even parallel post-processors are required to manage the large data-sets that are generated in transient computations. An example for such a program is ParaView (v3.4) [Ahrens et al., 2005; Henderson, 2007] that has been used to visualize most of the three-dimensional simulation results in this thesis.

Most current programs are unable to visualize the sharp, discontinuous solution, because the information about the discontinuity cannot readily be expressed in the post-processor software. In particular, the concept of integration cells is usually not present in current FE post-processors. However, a sharp solution can be visualized only with integration cells or similar subdivision and by using the enriched DOFs. A rather ad hoc treatment of cells as elements without connectivity leads to a constantly changing mesh. For large scale computation, such constantly changing meshes are impractical due to the increased memory and computation costs. Nevertheless, the ability to see integration cells and the sharp solution is useful in the implementation and test phase. For that purpose, the research pre- and post processor Gmsh [Geuzaine and Remacle, 2009] is used to display the changing meshes.

For certain visualizations, it is sufficient to display smeared discontinuities, i.e. the correct physical nodal values are interpolated linearly without consideration of the discontinuity. In this work, a smeared output is generated in the simulation code such that only a fixed grid with a constant number of unknowns is written to file in each time step. Then visualization and animation features can then be applied as usual and the post-processing performance is identical to standard finite element methods. Problems occur, if the post-processor computes derived quantities near the interface. For instance, mass flow or impulse rate computations through implicit surfaces give incorrect results. Similarly, particle tracer receive wrong element velocities near the interface such that particles may cross the moving FSI surface and are stuck inside the structure. Such computations would have to be moved into the actual simulation code, where the complete discontinuous approximation is present. This leads to a tight coupling of simulation and post-processing, since the entire computation has to be repeated, if a different visualization is requested.

Consequently, existing post-processors have to be extended to allow integration cells and enriched approximations. This includes file formats to exchange enriched data between simulation code and post-processor.

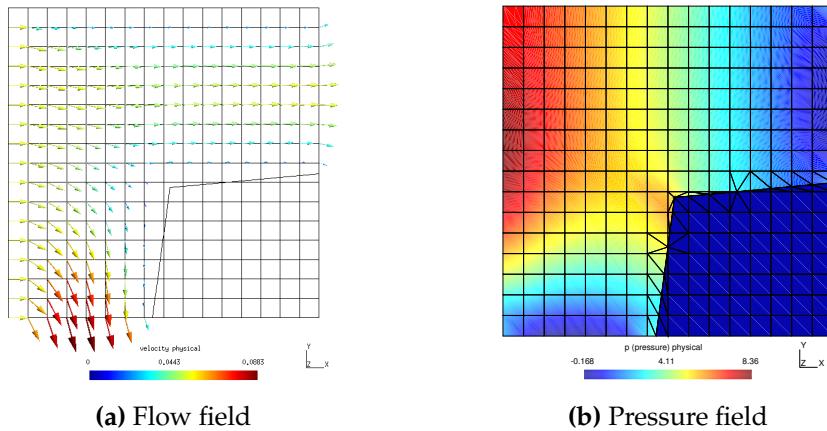
## 4.5 Numerical examples

In the following section, simulation results of stationary and instationary fluid flow around embedded discretized structures are shown. The intersected elements are shown using the triangular and tetrahedral integration cells to properly display the sharp interface in visualized scalar fields as discussed in the previous section.

The results are computed using two different implementations. If not indicated otherwise, the results are computed by a three-dimensional implementation, where the new stress Lagrange multiplier have been used exclusively. With the three-dimensional code, also two-dimensional examples have been computed by letting all velocities normal to the  $x$ - $y$  plane be zero. A two-dimensional implementation using equal order quad4 and quad9 fluid elements allows to test the traction Lagrange multiplier method. Here, the interface mesh is discretized by linear line elements.

### 4.5.1 Qualitative examples

The intention of these initial examples is to demonstrate the flexibility with respect to structural shapes that can be used in the simulation. Coarse meshes have been used to show essential features of the enrichment, integration cell



**Figure 4.9:** Stationary flow around bulky structure

distribution and to demonstrate that the XFEM discontinuities and Lagrange multiplier formulations work for arbitrary fluid mesh densities.

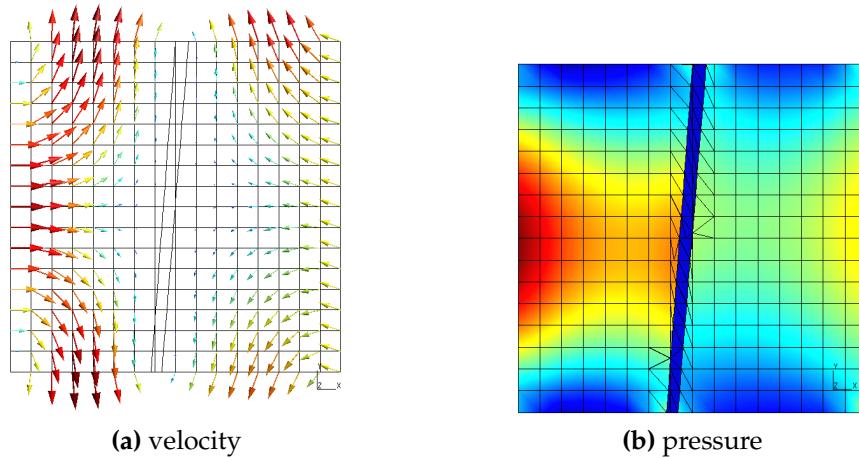
### Flow against a bulky structure

The first example, shown in Figure 4.9, illustrates a flow around a structural corner. The obstacle is located in the lower right corner. The inflow from the left boundary ( $\mathbf{u} = (0.05, 0.0, 0.0)^T$ ) and the wall boundary at the top are standard Dirichlet boundary conditions. The remaining boundaries are of Neumann (zero traction) type.

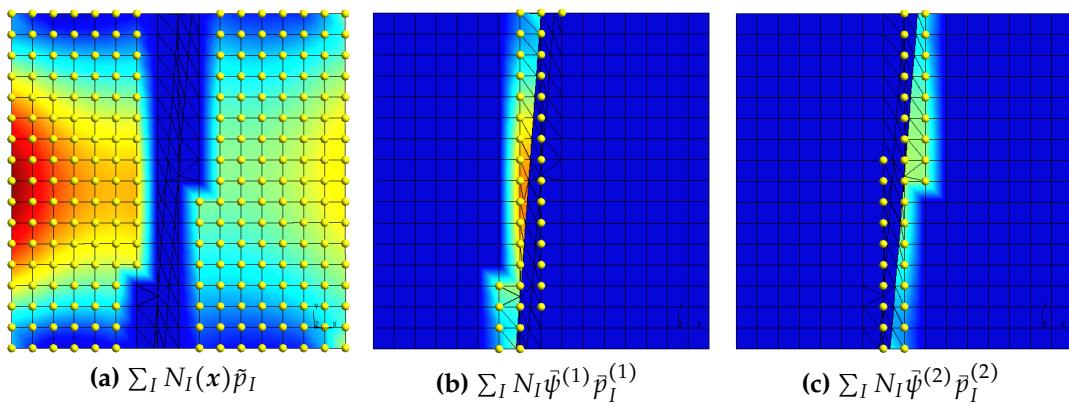
No fluid flow is computed within the structural domain in the lower right corner. In addition, smooth velocity and pressure fields are computed using the XFEM enrichment with fully integrated integration cells, although the interface arbitrarily intersects fluid elements. The velocity along the fluid-structure interface is zero as enforced weakly using the traction Lagrange multipliers. The triangular integration cells in Figure 4.9b allow to visualize the pressure discontinuity without smearing the solution.

### Flow around a thin structure

In the second example, different flow fields develop around a thin structure. Hereby, the structure thickness is smaller than the fluid element size. On the left side, a horizontal parabolic inflow is applied, while diagonal inflow enters the domain on the right side. Figure 4.10a and Figure 4.10b illustrate that the velocity and the pressure of the flow right and left of the structure are decoupled from each other and also from the small fictitious fluid domain between them.



**Figure 4.10:** Flow around thin structure.



**Figure 4.11:** Flow around thin structure: pressure enrichment. Bright dots indicate, which nodes are used for a particular enrichment.

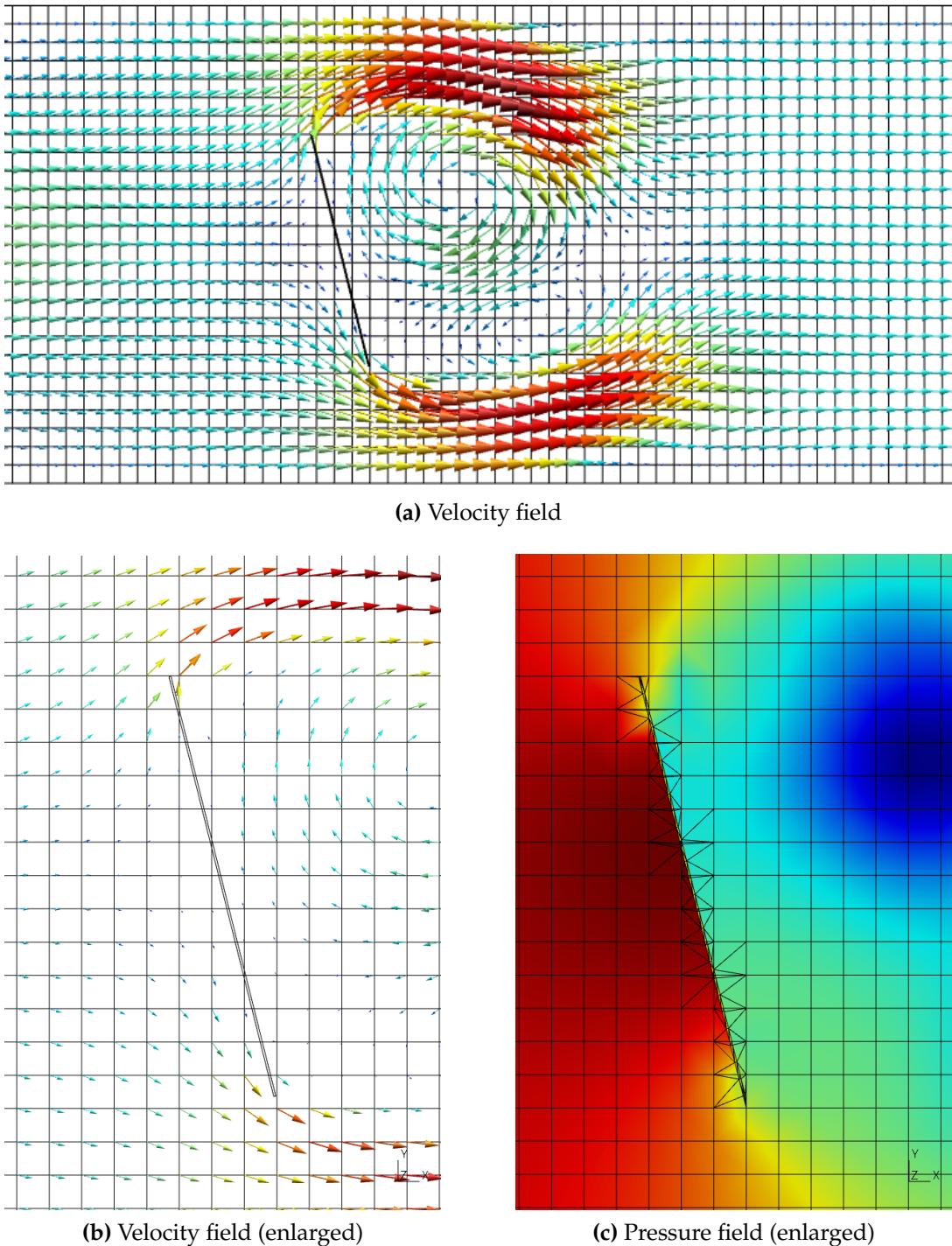
The pressure solution is composed of three distinct sets of degrees of freedom with their corresponding approximation function. The three sets of enrichments given by

$$p^h(x) = \sum_I N_I(x) \tilde{p}_I + \sum_J N_J(x) \bar{\psi}^{(1)}(x) \bar{p}_J^{(1)} + \sum_K N_K(x) \bar{\psi}^{(2)}(x) \bar{p}_K^{(2)}$$

are visualized in Figure 4.11a-4.11c. The standard DOFs, depicted in Figure 4.11a occupy most of the domain. Near the surface the first and the second enrichment overlap each other (Figure 4.11b and 4.11c). The sum of these enrichments gives the pressure field as displayed in Figure 4.10b.

## Flow around thin, closed structures

The following examples demonstrate how closed structures with thin parts are simulated. The first example is shown in Figure 4.12. The structure is



**Figure 4.12:** Flow around thin, closed structure: Example 1.

approximately ten times thinner than the fluid element edge length. In the center, the pressure jumps from the upstream side value to zero in the fictitious domain and then to the low pressure on the downstream side. For the shown coarse mesh, the coarse fluid mesh improperly resolves the pressure discontinuity for such sharp ending of the structure domain. Nevertheless, the principal decoupling of physical and fictitious domain is preserved. Despite of the unresolved thin endings, the wall boundary condition is appropriately enforced. Hence, such coarse setup could be the initial solution for an  $h$ -adaptive simulation and flow features of interest can be resolved by mesh refinement. The applied FSI enrichment is located at the top and bottom end of the thin domain as discussed in Section 4.2.2.

The transition between FSI enrichment and enrichment for each interface is smooth and therefore not visible in the final solution. This is demonstrated in the second example shown in Figure 4.13. Here, the FSI enrichment is applied at the upper and lower straight sides of the structure. The joints are not visible in the resulting flow field.

Both examples are computed using hex20 fluid elements and condensed stress Lagrange multiplier.

### 4.5.2 3D patch tests

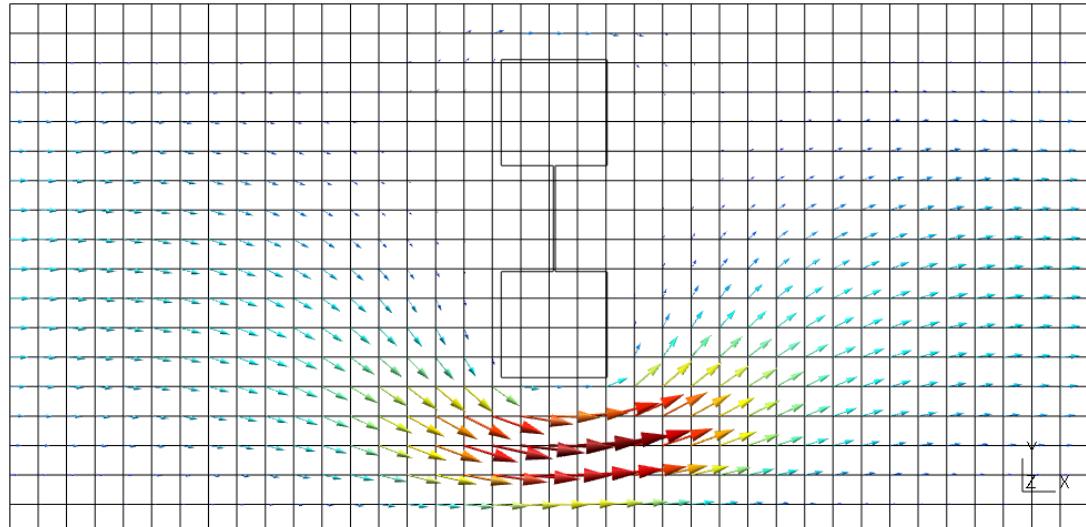
#### Pure shear flow

The first example verifies the correctness of viscous stress computations and interface forces. For this purpose, a shear flow with a linear velocity distribution in the  $y$ -direction and consequently, a constant viscous stress  $\sigma_{xy}$  component is simulated. The physical fluid domain has the dimensions  $1.0\text{ m} \times 0.5\text{ m} \times 0.1\text{ m}$  and is modeled by a larger computational fluid domain, which is intersected by an embedded plane wall at  $y = 0.5\text{ m}$  as shown in Figure 4.14a.

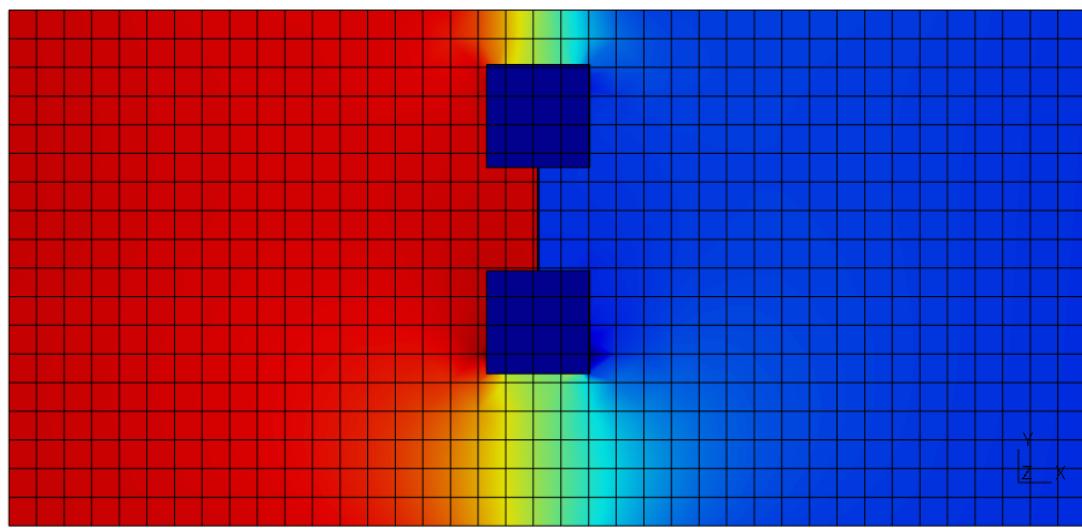
The fluid density is  $\rho^f = 1.0\text{ kg/m}^3$  and the kinematic viscosity is  $\nu = 10^{-3}\text{ m}^2/\text{s}$ . At the top, the velocity in  $x$ -direction is prescribed as  $u_x = 1.0\text{ m/s}$ ; the  $y$ -component is set to zero. On the left and right sides, the velocity component in  $y$ -direction is set to zero to achieve a horizontal inflow. In addition, no flow in  $z$ -direction is allowed at all nodes. The resulting velocity solution is shown in Figure 4.14b.

The chosen setup results in a constant shear rate and constant viscous stress in the fluid domain. The pressure and its gradient are zero. In addition, all Cauchy stress components but  $\sigma_{xy}$  (and, by angular momentum conservation, also  $\sigma_{yx}$ ) are zero. The analytic value of  $\sigma_{xy}$  is

$$\sigma_{xy} = \tau_{xy} = \rho^f \nu \left( \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right) = \nu (2.0\text{ s}^{-1} + 0.0\text{ s}^{-1}) = 0.002\text{ N/m}^2 \quad (4.65)$$

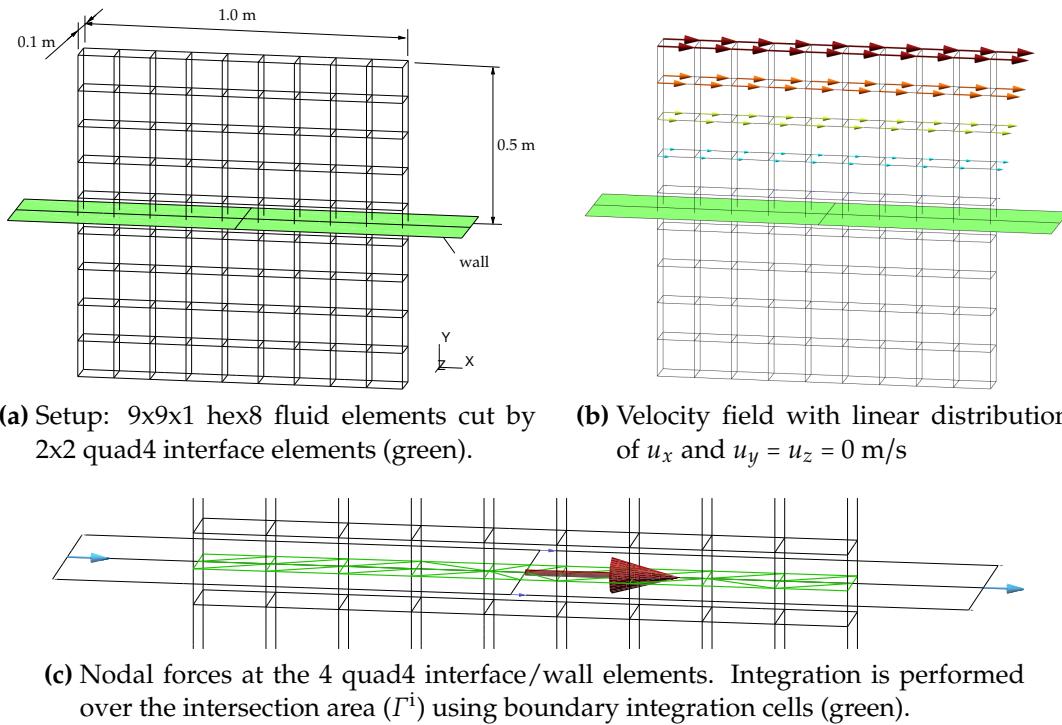


(a) Velocity field



(b) Pressure field

**Figure 4.13:** Flow around thin, closed structure: Example 2.



**Figure 4.14:** Shear flow with linear velocity distribution and constant viscous stress.

The computed value for  $\bar{\sigma}_{xy}$  in the intersected elements matches the analytic value above up to numerical precision.

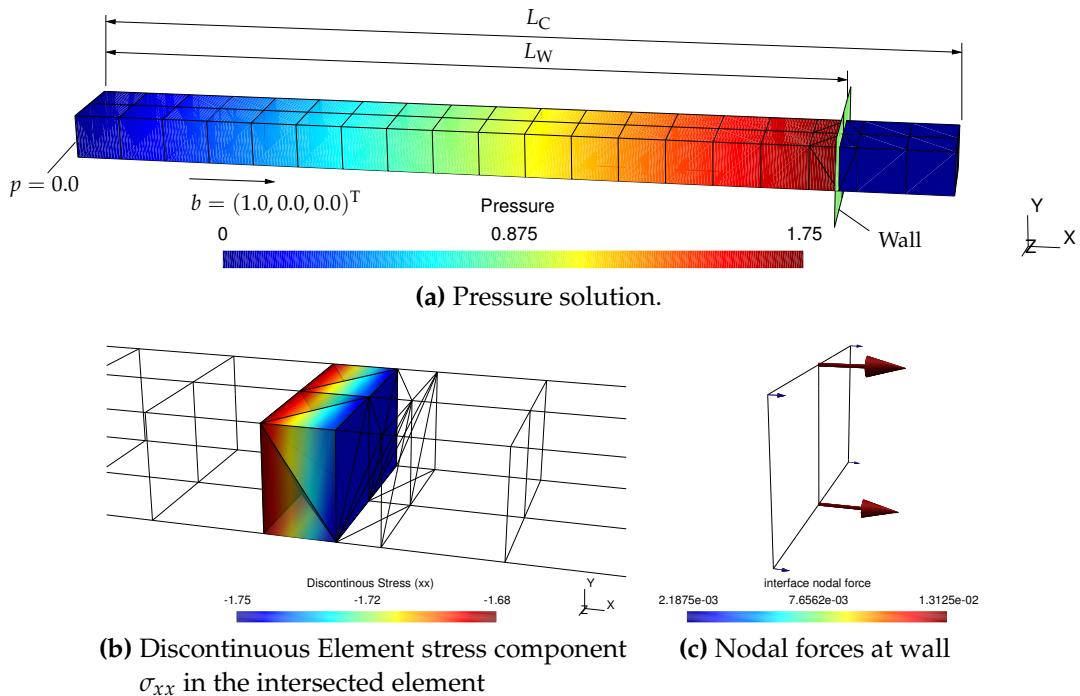
With the envisioned FSI approach in mind, the interface is given as a discrete mesh consisting of four quad4 elements. The interface velocity  $\mathbf{u}^i$  and its test function  $\delta\mathbf{u}^i$  are discretized using the shape function  $N_K^{u^i}$ . The integral  $(\delta\mathbf{u}^i, \boldsymbol{\sigma} \cdot \mathbf{n}^f)_{\Gamma^i}$  yields the nodal forces for each node  $K$  at the interface mesh as

$$f_K^i = \int_{\Gamma^i} N_K^{u^i} \boldsymbol{\sigma} \cdot \mathbf{n}^f d\mathbf{x} \quad (4.66)$$

The computed nodal forces and the boundary integration cells used for integration are shown in Figure 4.14c. The analytic value for the force in  $y$ -direction on the ‘wet’ intersection area  $A^i = 0.1 \text{ m}^2$  is  $F_y = \sigma_{yx} A^i = 0.2 \times 10^{-3} \text{ N}$ . Adding all  $y$ -components of the computed nodal forces of the interface mesh (Figure 4.14c), the exact same value (up to numerical precision) is obtained.

### Hydrostatic pressure

The next example verifies that linear pressure solutions can be represented exactly and that the exact pressure solution leads to correct interface forces due to the pressure acting on the wall. For this purpose, a body force  $\hat{\mathbf{b}} = \rho^f(1.0 \text{ m/s}^2, 0.0, 0.0)^T$  applied to fluid in a channel leads to a hydrostatic pressure field as depicted in Figure 4.15. Velocities in  $y$  and  $z$  direction are con-



**Figure 4.15:** Hydrostatic pressure distribution and resulting wall forces.

strained such that all velocities remain zero and therefore, no viscous stresses are present in the solution.

The computational fluid domain  $\Omega$  of the channel has the dimensions  $2.0 \text{ m} \times 0.2 \text{ m} \times 0.1 \text{ m}$  and is modeled with  $19 \times 2 \times 1$  hex8 fluid elements. The intersecting wall consists of two quad4 elements and intersects the channel length  $L_w = 1.75 \text{ m}$  with the bigger part being the physical fluid domain  $\Omega^f$ . The fluid density is  $\rho^f = 1.0 \text{ kg/m}^3$ . With the given body force  $\hat{b}$  and zero pressure at the left-hand channel ending, the analytic pressure at the interface is  $p = b_x L_w = 1.75 \text{ N/m}^2$ . The setup and the pressure solution are given in Figure 4.15a. The solution of the element stress component  $\sigma_{xx}$  in the intersected element is plotted in Figure 4.15b, the resulting interface force is shown in Figure 4.15c.

From the absence of viscous forces it follows that

$$p(x) = -\sigma_{xx}(x) = -\sigma_{yy}(x) = -\sigma_{zz}(x) \quad \text{in } \Omega^+ \quad (4.67)$$

To match the linear pressure distribution in this example exactly, at least a (tri-)linear stress approximation is required. For elementwise constant approximation functions for the stress field,  $\sigma_{xx}$  can not be equal to  $1.75 \text{ N/m}^2$  along the interface and equal to the linear pressure distribution at the same time. Consequently, the interface forces would be inexact even for this simple example. For the result shown in Figure 4.15, the same polynomial order

is used for the stress shape functions as for the velocity and pressure fields ( $Q_1 Q_1 Q_{-1}$ ). Hence, linear stress components can be exactly represented.

## Conclusion

The linear pressure example and the shear flow example with constant viscous stresses allow the conclusion that the stress approximation should be chosen to be of the same or higher polynomial order as the pressure approximation. At the same time, the stress approximation should have the same or higher polynomial order as the velocity derivatives used for the viscous stress computation. In the present implementation with equal order velocity-pressure approximations, the pressure dictates the minimal polynomial order for the stress approximation. For other elements, e.g. the Taylor-Hood element, the velocity derivatives might dictate the minimal stress approximation. Future mathematical analysis should provide more understanding for this ‘rule of thumb’.

### 4.5.3 3D benchmark computations

#### Jeffery-Hamel flow

In the following, the Jeffery-Hamel flow is used to study the convergence behavior of the proposed weak Dirichlet constraint approach. The Jeffery-Hamel flow is a flow between two converging walls, for which a two-dimensional steady-state solution with only radial velocity components can be established [Hamel, 1916; Jeffery, 1915; Rosenhead, 1940].

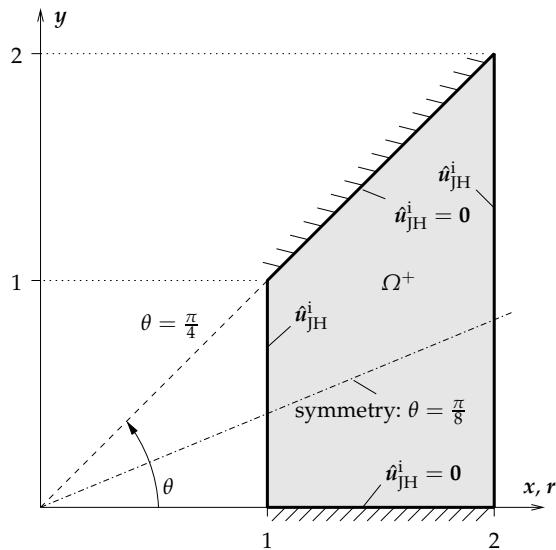
The radial velocity  $u_r(\theta)$  is given only implicitly as solution of a nonlinear ordinary differential equation (ODE) of the radial velocity. The problem can be expressed by ODEs with varying order and corresponding boundary conditions, which also influence how analytic solutions can be computed. Here, the discussion on solution strategies in [Corless and Assefa, 2007] is adopted, where the following third-order ODE is used

$$u_r''' + 4u_r' + 2u_r'u_r = 0 \quad , \quad 0 \leq \theta \leq \alpha = \frac{\pi}{4} \quad (4.68)$$

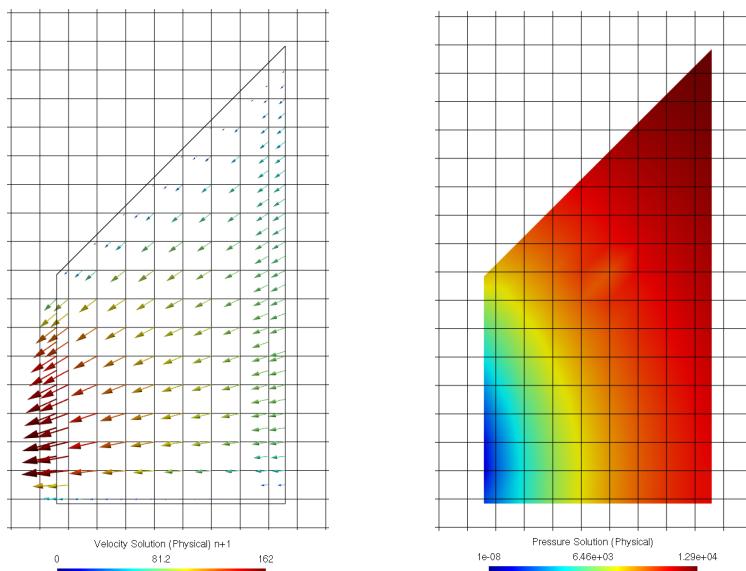
The boundary conditions are chosen such that a converging flow as shown in Figure 4.16 is described. In particular, the setup in Figure 4.16a requires the following boundary conditions

$$u_r(0) = 0 \quad u_r\left(\frac{\pi}{8}\right) = -170, \quad u_r'\left(\frac{\pi}{8}\right) = 0$$

The solution is symmetric with respect to  $\theta = \frac{\pi}{8}$  and the maximum velocity at the symmetry line at radius  $r = 1$  is set to  $-170$ . With the kinematic viscosity



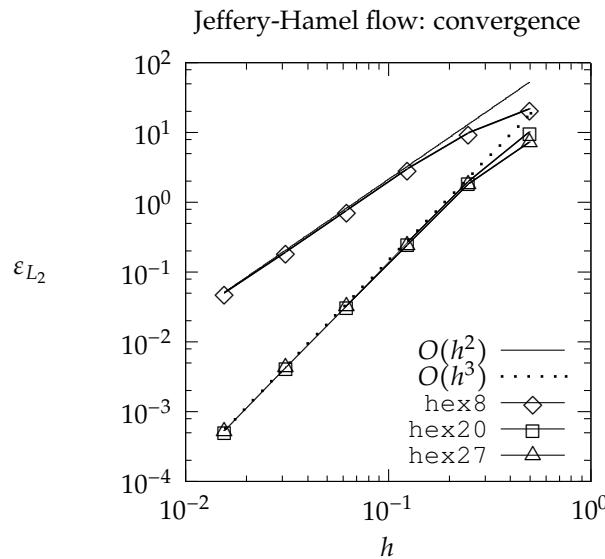
(a) Setup



(b) Velocity solution on 2. refinement level.

(c) Pressure solution on 2. refinement level.

**Figure 4.16:** Jeffery-Hamel flow: Setup and example solutions. Convergence study using the two-dimensional Jeffery-Hamel flow solution. The actual computation is performed using 3D hexahedral elements. All Dirichlet conditions are applied using the hybrid embedded DBC approach.



**Figure 4.17:** Jeffery-Hamel flow: convergence rates.

set to  $\nu = 1$ , the Reynolds-number is computed by

$$\text{Re} = -\frac{3}{4} \int_0^\alpha u_r(\theta) d\theta$$

as defined in [Corless and Assefa, 2007]. For the given velocity boundary conditions, the Reynolds number is approximately  $\text{Re} \approx 85$ . The computation of the reference solution is given in Appendix B.

All Dirichlet conditions are applied using the new stress Lagrange multiplier formulation. Since all boundaries prescribe velocities, it is necessary to prescribe the pressure at an arbitrary node within the fluid domain. The actual computation is performed using 3D hexahedral elements. The corresponding approximation error for the velocity field is computed by

$$\varepsilon_{L_2} = \| \mathbf{u}^h - \mathbf{u}^{\text{exact}} \|_{L_2} = \sqrt{\int_{\Omega_f} |\mathbf{u}^h - \mathbf{u}^{\text{exact}}|^2 dx} \quad (4.69)$$

are shown in Figure 4.17. It can be seen that the error decreases with optimal rates for all tested element types.

Special care is required for the computation of the stabilization parameters for the NS equations, since refinement in two dimensions results in rectangular element shapes of the 3D elements. As stabilization parameter  $\tau_{\text{PSPG}}^e$  of the NS equation uses the volume based element length scale  $h_e^V$  (Eq. (2.33)), elements with changing aspect ratios will influence the optimal convergence. Refining also in the third dimension (here  $z$ ) or reducing the thickness of the computational domain and correcting the decreasing volume by normalizing the domain thickness leads to the optimal convergence presented in Figure 4.17.

### Benchmark computations - 3D implementation

For further validation, the benchmark cases in [Schäfer and Turek, 1996] were studied. Here, incompressible, laminar flows ( $Re = 20\text{-}100$ ) over a cylinder and a cuboid are computed. In particular, the stationary two-dimensional case 2D-1, the instationary two-dimensional case 2D-3 and the three-dimensional benchmark cases 3D-1Q and 3D-1Z were computed. The setup, boundary conditions and results are taken without modifications from the original publication, but will be repeated here for convenience.

Instead of defining the obstacles using Dirichlet conditions and boundary aligned meshes, the cylinder/cuboid is discretized using a fixed number of 8-node hexahedral elements, which results in an interface mesh of quad4 elements. For each benchmark, the number of structural elements, and therefore interface elements, is kept constant for all fluid refinement levels.

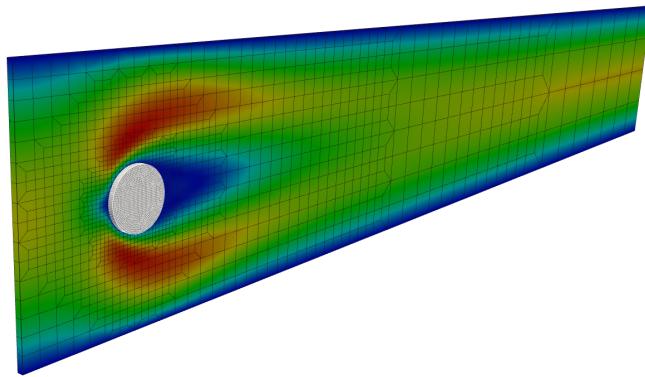
**2D benchmarks** The two-dimensional benchmarks—calculated using the three-dimensional implementation—were used to demonstrate that the developed approach converges to the correct solution for linear and quadratic element shape functions. Furthermore, it is demonstrated that stress condensation leads to identical results with less unknowns.

The benchmark 2D-1 consists of a two-dimensional channel with no-slip conditions at the upper and lower wall, a parabolic inflow with a maximum velocity of 0.3 m/s and a zero-stress Neumann condition at the outflow. The two-dimensional setup is modeled with one layer of hexahedral elements in the third dimension. Refinement of the fluid mesh is performed only in the first two dimensions<sup>1</sup>. An example solution is given in Figure 4.18.

The measure used for comparison are the lift ( $c_{\text{lift}}$ ), the drag ( $c_{\text{drag}}$ ), and the pressure difference ( $\Delta p$ ) between front and back side of the cylinder. Note that the three measures may not converge monotonically in an adaptive computation. The lift is especially sensitive since it is the sum of large opposing forces in positive and negative  $y$ -direction, where the sum is close to zero as shown in the following tables. Thus, small mesh variations in the not yet converged solution may lead to relatively large variations around the reference solution. Similar effects can be observed also for the pressure difference. Nevertheless, sufficient refinement leads to the reference solution for all presented examples. Note further, that the reference solution is a compilation of

---

<sup>1</sup>To make the number of unknowns comparable to the reference computations, the number of unknowns are multiplied by the factor 8/20, since three-dimensional (hex20) elements instead of two-dimensional (quad8) elements are used. Likewise, a factor of 4/8 is needed to compare the number of DOFs for the hex8 element with linear, quadrilateral quad4 elements in [Schäfer and Turek, 1996].



**Figure 4.18:** Benchmark case 2D-1: velocity field. Shown is the coarsest mesh with hex8 elements.

several simulations in the reference publication, hence, no number of DOFs is given as reference value.

The results of the computations are given in Table 4.1. For all element types, the computed solutions agree with the given reference values. The necessary number of unknowns to achieve this accuracy is within the range of reference computations.

The main conclusion drawn from these results is that condensation of stress unknowns does not change results as exemplary shown for hex8 and

**Table 4.1:** Results for the stationary benchmark case 2D-1.

element type	$N_{u,p}^{\text{DOF}}$	$N_{\sigma}^{\text{DOF}}$	$c_{\text{drag}}$	$c_{\text{lift}}$	$\Delta p / (\text{N/m}^2)$
hex8, $\bar{\sigma}$ uncond.	12040	2832	5.6066	0.0093	0.1147
	40416	5712	5.5893	0.0093	0.1149
	146328	11376	5.5814	0.0105	0.1156
hex8, $\bar{\sigma}$ cond.	12040	0	5.6066	0.0093	0.1147
	40416	0	5.5893	0.0093	0.1149
	146328	0	5.5814	0.0105	0.1156
hex20, $\bar{\sigma}$ uncond.	14884	3840	5.5739	0.0063	0.1169
	48020	7440	5.5766	0.0108	0.1159
	166388	14880	5.5753	0.0107	0.1175
hex20, $\bar{\sigma}$ cond.	14884	0	5.5739	0.0063	0.1169
	48020	0	5.5766	0.0108	0.1159
	166388	0	5.5753	0.0107	0.1175
Ref. lower bound	-	-	5.5700	0.0104	0.1172
Ref. upper bound	-	-	5.5900	0.0110	0.1176

**Table 4.2:** Results for the instationary benchmark case 2D-3.

element type	$N_{u,p}^{\text{DOF}}$	$\Delta t/\text{s}$	$c_{\text{drag,max}}$	$c_{\text{lift,max}}$	$\Delta p/(\text{N/m}^2)$
hex20, $\bar{\sigma}$ condensed	3444	0.008	3.1828	0.3944	-0.1078
	10488	0.008	2.7824	0.1308	-0.1141
	30504	0.008	2.9273	0.4861	-0.1077
	102504	0.008	2.9470	0.4733	-0.1093
Ref. lower bound	-	-	2.9300	0.4700	-0.1150
Ref. upper bound	-	-	2.9700	0.4900	-0.1050

hex20 elements. When comparing the condition number of condensed and uncondensed global system matrices, the condensed matrices have a reduced condition number for all results presented in this thesis. Since the computational cost of condensation is negligible when compared with numerical integration, stress condensation is performed for all remaining examples.

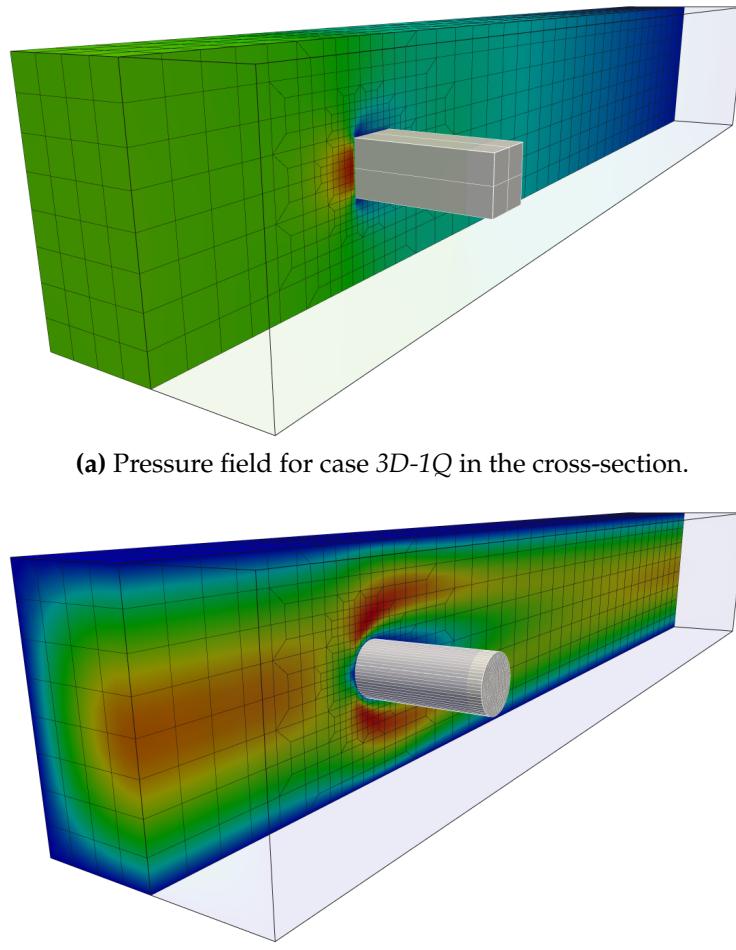
The setup for benchmark 2D-3 is almost identical to case 2D-1. The only difference is a dynamic inflow condition, where the parabolic inflow over the height ( $y$ -direction) is given as

$$U(x = 0, y, t) = 4U_m y(H - y) \sin(\pi t/8 \text{ s})/H^2, \quad V = 0 \quad (4.70)$$

with  $H = 0.41 \text{ m}$  and  $U_m = 1.5 \text{ m/s}$ . The simulated time span is 8 s. Measures are the maximum drag ( $c_{\text{drag,max}}$ ) and lift values ( $c_{\text{drag,max}}$ ) in this time span as well as the pressure difference between front and rear end of the cylinder at  $t = 8 \text{ s}$ . For time discretization, the one-step- $\theta$  ( $\theta = 0.66$ ) and the second order BDF2 scheme is applied.

The results of the instationary two-dimensional computations are given in Table 4.2. Note that only the BDF2 results are given as they were more accurate as expected from a second order scheme. Good agreement with the reference results is achieved also for the instationary case.

**3D benchmark** The three-dimensional benchmarks 3D-1Q and 3D-1Z have almost identical setups: the fluid domain is  $0.41 \times 0.41 \times 2.5$  with ‘no-slip’ conditions on the upper, lower and side walls. A three-dimensional parabolic inflow is applied using standard Dirichlet conditions and zero-traction Neumann conditions are applied at the outflow. The cases 3D-1Q and 3D-1Z differ in the shape of the immersed structure: a cuboid with dimensions  $0.1 \times 0.1 \times 0.41$  in case 3D-1Q and a cylinder with radius 0.05 and length 0.41 in case 3D-1Z. For illustration, example solutions for both cases are given in Figure 4.19. The results of both three-dimensional computations are given in Table 4.3 and Table 4.4.



**Figure 4.19:** 3D benchmark cases 3D-1Q and 3D-1Z

Most results match the reference values for drag  $c_{\text{drag}}$ , lift  $c_{\text{lift}}$  and pressure difference between front and rear end of the structure  $\Delta p$  within the given tolerances. The remaining disagreements are contributed to the insufficient spatial resolution (due to limited computational resources) when compared to the reference computations in [Schäfer and Turek, 1996]. Nevertheless, the accuracy is comparable to results computed with similar numbers of unknowns. Case 3D-1Q demonstrates that the method can deal with sharp corners of the interface constraint, even though a pressure singularity exists at the two front edges of the cuboid. Case 3D-1Z was used to test the new formulation on hexahedral and tetrahedral element shapes. For all tested element shapes, the solution tends towards the reference solution.

Figure 4.19 highlights again, that the fluid discretization does not constrain the element size of the interface mesh. The surface elements of the cubical structure are two to four times bigger than the intersected fluid elements. For the cylinder, these surface elements are smaller than the fluid

**Table 4.3:** Results for benchmark case 3D-1Q.

element type	$N_{u,p}^{\text{DOF}}$	$c_{\text{drag}}$	$c_{\text{lift}}$	$\Delta p / (\text{N/m}^2)$
hex20, $\bar{\sigma}$ condensed	27936	7.5465	0.1926	0.1761
	88036	7.5408	0.0759	0.1736
Ref. lower bound	-	7.5000	0.0600	0.1720
Ref. upper bound	-	7.7000	0.0800	0.1800

**Table 4.4:** Results for benchmark case 3D-1Z.

element type	$N_{u,p}^{\text{DOF}}$	$c_{\text{drag}}$	$c_{\text{lift}}$	$\Delta p / (\text{N/m}^2)$
tet4, $\bar{\sigma}$ condensed	66268	6.1534	0.0356	0.1645
	240708	6.1533	0.0115	0.1744
tet10, $\bar{\sigma}$ condensed	69724	6.1979	0.0506	0.1703
	246780	6.1619	0.0064	0.1751
hex20, $\bar{\sigma}$ condensed	28112	6.2093	0.1238	0.1645
	89156	6.1572	0.0050	0.1664
hex27, $\bar{\sigma}$ condensed	74936	6.2026	0.0736	0.1659
	169592	6.1567	0.0067	0.1663
Ref. lower bound	-	6.0500	0.0080	0.1650
Ref. upper bound	-	6.2500	0.0100	0.1750

elements. The fine structure resolution is only needed in circumferential direction to approximate the circular shape. The interface resolution in axial direction had no influence on the lift and drag values, which are sums over all nodal forces of the interface mesh. However, if the force distribution on the surface is of interest, then a finer interface mesh in axial direction is necessary as well. Such force distribution is required for FSI computations with flexible structures. In combination with the examples in Section 4.5.2, the results indicate that correct surface forces can be computed and that no leakage for any ratio of surface to fluid element sizes exists.

# CHAPTER 5

## Moving implicit fluid surfaces

This chapter introduces moving fluid boundaries into the developed XFEM formulation. Introductory remarks highlight the specific difficulties when using an Eulerian formulation in combination with moving boundaries. Subsequently, a Ghost fluid method for moving boundaries is described, which is then extended to include impermeable wall boundary conditions.

### 5.1 Introduction

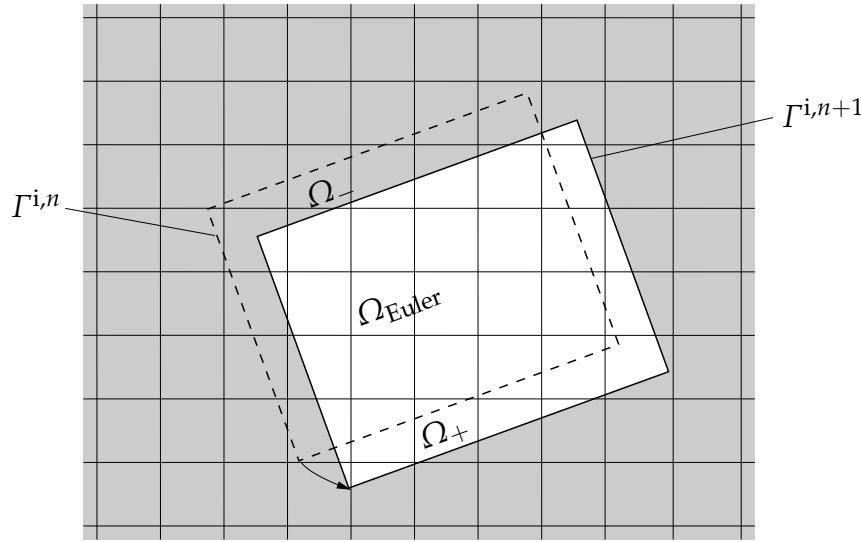
In transient FSI problems, the position of the domain boundary  $\Gamma^i$  depends on  $x$  and  $t$ . For fixed-grid methods with moving implicit interfaces, two time-integration approaches to this problem can be found in literature: a (discontinuous) FE space-time formulation or an Eulerian fluid formulation with traditional finite difference time-stepping. The latter approaches are sometimes combined with a (local) ALE approach with repeated projections onto a fixed Eulerian fluid grid.

A space-time method for implicit FSI interfaces has been recently proposed in [Zilian and Legay, 2008]. The used stabilized space-time formulation for the incompressible NS equations is based on works as [Hughes and Hulbert, 1988; Tezduyar et al., 1992a,b]. For general remarks on space-time techniques, see also [Lew et al., 2003]. The space-time formulation provides a consistent formulation of the evolving implicit interface in space and time. A key property in the context of XFEM is that domain integration cells are aligned with the interface in space *and* time. This avoids all ambiguities resulting from traditional finite difference time stepping techniques as described in the remainder of this chapter. However, space-time methods require more computational resources than finite difference time stepping schemes with respect

to both computation time and computer memory. The particular complication for space-time methods in combination with XFEM is the numerical integration, which requires to ‘triangulate’ the four-dimensional space-time domain. The three-dimensional space-time formulation — two space dimensions plus one time dimension — presented in [Zilian and Legay, 2008] may be relatively easy to extend to four dimensions from a mathematical point of view. However, the four-dimensional implementation faces several practical problems. Many existing geometric algorithms, for instance for mesh generation and triangulation, would require extensions or even a rewrite. In addition, the structure field has also been treated with a space-time formulation in [Zilian and Legay, 2008]. Since the initially stated goal of this thesis is an implementation for three spatial dimensions without modifications of the structure formulation, the present work concentrates on traditional time-stepping methods.

For finite-difference time-stepping, the ALE formulation allows a reference movement that is independent of the convective fluid formulation. The reference movement is necessary to let the fluid mesh follow the structure surface. However, an overly strong deformation of the reference mesh eventually requires remeshing, which is one of the primary motivations for research in fixed grid methods. An example for a combined fixed-grid / arbitrary Lagrangian Eulerian (FG/ALE) approach has been described in [Codina et al., 2009] and references therein. The implicit interfaces at time step  $n$  can arbitrarily intersect fluid elements and integration cells are generated accordingly. To advance to the new time step, fluid elements and integration cells within a certain distance from the interface deform to follow the interface using the ALE formulation. Away from the interface the mesh velocity is zero and the ALE approach reduces to the Eulerian formulation. After solution for the new time step values, velocities and accelerations are projected from the deformed fluid mesh onto the undeformed background grid such that the interface now intersects the next layer of fluid elements. The projection is combined with the additional constraint that the projected velocity  $u_{\text{projected}}^{h,n}$  is discretely incompressible for the new interface configuration. With the FG/ALE approach most of the fluid domain remains undeformed and remeshing is avoided. However, projection between non-fitting meshes is now required at every time step near the interface. Even though only a relatively small domain along the implicit interface is affected, the introduced error must be considered for the entire accuracy and stability of the transient solution.

A purely Eulerian description of the flow field has a long tradition in FD and FV methods. It is influenced by works as [Fedkiw et al., 1999; Sussman, 2003; Sussman et al., 1994] and references therein. Summaries can be found for example in [Osher and Fedkiw, 2003; Sethian, 1999]. The key observation



**Figure 5.1:** Deforming observed domain on a fixed fluid grid.

is that after time-discretization as presented in Section 2.1.2, the conservation equations are enforced at  $\Omega^{n+1}$  without taking the deformation of the observed domain from step  $n$  to  $n + 1$  into account. From a continuum mechanics point of view, it is questionable to transform the integral equations of mass and momentum conservation into their local or differential equation forms, because the integral boundaries over the domain are time-dependent. The differential equations, however, are required as a starting point for the weighted-residual based finite element formulation.

For illustration, Figure 5.1 shows a deforming domain at two time instances. The domain that is under observation at the old and at the new time step is denoted as  $\Omega_{\text{Euler}}$ :

$$\Omega_{\text{Euler}} = \Omega^{n+1} \cap \Omega^n \quad (5.1)$$

The part that is under observation at the new time step  $n + 1$  but was not observed at the old time step  $n$  is denoted as

$$\Omega_+ = \Omega^{n+1} \setminus \Omega_{\text{Euler}} \quad (5.2)$$

while the part that is not observed at time step  $n + 1$  is denoted as

$$\Omega_- = \Omega^n \setminus \Omega_{\text{Euler}} \quad (5.3)$$

Fluid material movement in  $\Omega_{\text{Euler}}$  is properly described using the Eulerian approach. Points in  $\Omega_-$  are not under observation anymore and hence may not influence the solution. The critical point is that the discretization of partial time derivatives at a spatial point  $x$  in  $\Omega_+$  requires velocities and accelerations

at the old time step level. However, the flow was not observed and computed at the old time step, hence, no history values are present at newly observed nodes. One possible solution to this dilemma is to guess or to estimate what happened at the old time step.

Methods to estimate history values are described for example in [Hong et al., 2007; Osher and Fedkiw, 2003] and references therein. The basic idea is to extend the velocity field from the last position of the interface into a larger Ghost-fluid domain until all previously unknown nodes are initialized with history values. This guess is guided by mechanical and engineering insight into the modeled problem. In addition, incompressible flow requires an incompressible old flow field in the new interface configuration  $\Gamma^{i,n+1}$ . To ensure discrete incompressibility, a projection step with incompressibility constraints is often added. The required projection as e.g. described in [Houzeaux and Codina, 2001] is of the form: for a given  $\hat{\mathbf{u}}^{\text{guess},n}$ , find  $\mathbf{u}^n$  in domain  $\Omega^{n+1}$  such that

$$(\delta \mathbf{u}, \hat{\mathbf{u}}^{\text{guess},n} - \mathbf{u}^n)_{\Omega^{n+1}} = 0 \quad (5.4a)$$

$$(\delta p, \nabla \cdot \mathbf{u}^n)_{\Omega^{n+1}} = 0 \quad (5.4b)$$

For the two step BDF2 scheme, the velocity  $\mathbf{u}^{n-1}$  has to be treated accordingly. Projections of old accelerations, as needed in the OST methods, have not been tested, but may be necessary as well. If equal order shape functions for velocity and pressure are employed, additional (PSPG) stabilization is required. The quality of the guess and the applied projections are critical components for stability and temporal convergence of the transient computation.

One key insight from existing estimates is that properties of separate domains should not be mixed, i.e. the missing old values from one side of the interface should not be used as history values for the other side and vice versa. Applied to the FSI problem, it should be avoided to get any information from the structural domain. It might be tempting to interpolate the structure velocity onto the fluid mesh to get missing old velocities and accelerations. Although this might work for special cases, where both continua have similar properties, it generally leads to dependencies between the applied numerical schemes for each field. For instance, interpolating the velocity field of a structure that changes its volume would produce a fluid velocity field that is not divergence free. Such inherent coupling is present in methods like the immersed finite element method (IFEM) [Zhang et al., 2004] or the distributed Lagrange multiplier / fictitious domain (DLM/FD) [Yu, 2005], where for incompressible flows, only incompressible structures can be modeled. To avoid such dependencies, relying on strict surface coupling guides the following derivation.

## 5.2 A Ghost-Fluid method for the XFEM

### 5.2.1 Setup

In the following, the Ghost-fluid method [Hong et al., 2007; Kang et al., 2000] is adapted to the specific needs of the proposed fixed-grid FSI approach. The techniques introduced in Chapter 4 allow to describe the implicit boundaries of the observed fluid domain by enriched approximation functions. In addition, velocity conditions can be applied weakly along such interfaces. The extension in this section is the introduction of the moving interface.

The following assumptions are made for this extension: (i) a pure Eulerian formulation is applied, (ii) the observed fluid domain may move independently from any element boundaries, hence (iii) the fluid mesh exists regardless of whether the flow at some point is observed, (iv) the fluid solution is sought only in  $\Omega^{f,n+1}$ , and (v) a smooth movement of the surface of the observed domain is assumed in time. The last point allows reasonable estimates for old time step values of newly uncovered nodes.

### 5.2.2 Discretization

For moving interfaces, the enrichment function  $\psi(x, t)$  is a function of space and time. Hence, the approximation function is also space and time dependent as  $\phi(x, t) = N(x)\psi(x, t)$ . The computational domain, i.e. the fluid mesh, does not change; the changing geometry of the observed domain is incorporated in the time dependent enrichment function.

If time is discretized first, a purely spatial problem has to be discretized in space subsequently. This approach corresponds to the time-discretization as introduced in Section 2.1 and is repeated here for the OST as

$$\frac{\mathbf{u}^{n+1}(x) - \mathbf{u}^n(x)}{\Delta t} = \theta \mathbf{a}^{n+1}(x) + (1 - \theta) \mathbf{a}^n(x) \quad (5.5)$$

The semi-discrete strong form is tested and discretized in  $\Omega^{n+1}$  using the Bubnov-Galerkin method, i.e. test and trial functions are discretized with the same approximation functions. Hereby, the solution is sought on domain  $\Omega^{f,n+1}$  at time step  $n + 1$ . The resulting velocity discretization is

$$\mathbf{u}^{h,n+1}(x) = \sum_L \phi_L^{n+1}(x) \mathbf{u}_L^{n+1} = \sum_L N_L(x) \psi_L^{n+1}(x) \mathbf{u}_L^{n+1} \quad (5.6)$$

$$\delta \mathbf{u}^{h,n+1}(x) = \sum_L \phi_L^{n+1}(x) \delta \mathbf{u}_L^{n+1} = \sum_L N_L(x) \psi_L^{n+1}(x) \delta \mathbf{u}_L^{n+1} \quad (5.7)$$

$$\mathbf{u}^{h,n}(x) = \sum_L \phi_L^{n+1}(x) \mathbf{u}_L^n = \sum_L N_L(x) \psi_L^{n+1}(x) \mathbf{u}_L^n \quad (5.8)$$

Since the whole residuum is tested with the same test function, the new and the old time step have the same approximation function. As mentioned in the previous section, this approach requires to estimate velocity unknowns in  $\Omega_+^{n+1}$  such that  $\mathbf{u}^{h,n}(x)$  can be approximated with the new test function. Nodal pressures and element stresses  $\bar{\sigma}^h$  are also discretized using  $\psi^{n+1}$ . Having only one enrichment per time step implies that the triangulation for the numerical integration of the weak form respects only the position of the discontinuity at time-step  $n + 1$ .

In contrast to Eq. (5.8), reference [Fries and Zilian, 2009] suggests to use the old enrichment function value to approximate the old velocity in space such that

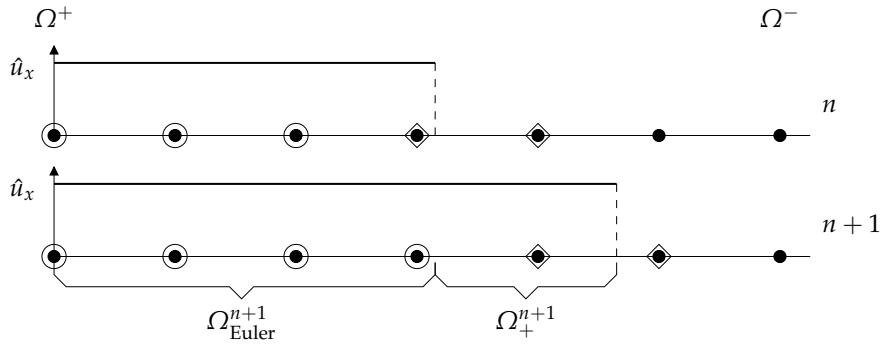
$$\mathbf{u}^{h,n}(x) = \sum_L \phi_L^n(x) \mathbf{u}_L^n = \sum_L N_L(x) \psi_L^n(x) \mathbf{u}_L^n \quad (5.9)$$

For the OST method, the old acceleration requires the old time step approximation, too. For the BDF2 scheme, the velocity  $\mathbf{u}^{n-1}$  would then require the appropriate  $n - 1$  enrichment function  $\psi^{n-1}$ . The unknown new velocity and the test function are approximated as in Eqs. (5.6) and (5.7). The problematic term in the weak form of the NS equation is

$$(\delta\mathbf{u}, \mathbf{u}^{\text{hist}})_{\Omega^{n+1}} = (\delta\mathbf{u}, \mathbf{u}^n + \Delta t(1 - \theta)\mathbf{a}^n)_{\Omega^{n+1}} \quad (5.10)$$

Using approximation Eq. (5.9), test approximation  $\phi_L^{n+1}(x)$  and old velocity approximation function  $\phi_L^n(x)$  are different for moving interfaces. Both, new and old time step approximations are present in the same term and fluid elements are intersected by new and old discontinuous functions. For the integral in Eq. (5.10), this would require to create integration cells respecting both discontinuities, which was also acknowledged in [Fries and Zilian, 2009]. This form is used in other works on transient XFEM problems, too, see e.g. [Chessa and Belytschko, 2003; Chessa et al., 2002]. Applying this approach does not require guessing of old values  $\mathbf{u}^n$  as the number of old unknowns fits the number of old approximation functions.

In summary, two choices seem to exist to discretize velocity  $\mathbf{u}^{h,n}(x)$ . The first is comparable with the Ghost fluid method, since the old velocity uses the new approximation and interface position only and requires an estimate for old time step values. The second requires new and old previous interface locations, however, no estimates are required. The following one-dimensional example shall show, whether both or only one of the two approaches can be used for the envisioned fixed-grid approach.



**Figure 5.2:** Moving domain boundary on an one-dimensional fixed fluid grid with Eulerian description of the fluid material movement.

### 5.2.3 A one-dimensional example

#### Setup

The following one-dimensional example setup demonstrates that only the Ghost fluid approach leads to a consistent solution for moving interfaces. For that purpose, the three-dimensional NS equations are reduced to an one-dimensional problem as

$$\rho_f \frac{\partial u_x}{\partial t} = - \frac{\partial p}{\partial x} \quad (5.11a)$$

$$\frac{\partial u_x}{\partial x} = 0 \quad (5.11b)$$

Ignoring the required stabilization (due to equal order velocity and pressure) for this thought experiment, the discrete weak form for this problem after integration by parts is given as

$$\rho_f (\delta u_x^h, u_x^{h,n+1})_{\Omega^{f,n+1}} - (\delta u_{x,x}^h, p^{h,n+1})_{\Omega^{f,n+1}} = \frac{\rho_f}{\Theta} (\delta u_x^h, u_x^{\text{hist},h,n})_{\Omega^{f,n+1}} \quad (5.12a)$$

$$(\delta p^h, u_{x,x}^{h,n+1})_{\Omega^{f,n+1}} = 0 \quad (5.12b)$$

The one-dimensional equation is used to model a flow along the \$x\$-axis as depicted in Figure 5.2. On the left, an inflow \$\hat{u}\_x(t)\$ is prescribed using standard Dirichlet conditions. On the right, the XFEM boundary with a zero-Neumann condition is applied, where the flow can leave the observed domain \$\Omega^f\$. Fluid material velocity and interface velocity are independent of each other. Putting the inf-sup condition and the resulting pressure stabilization aside, the solution to differential equation is uniquely defined; the velocity \$u\_x^h(x,t)\$ in the entire domain is constant in space equaling \$\hat{u}\_x(t)\$. For a time-invariant Dirichlet value \$\hat{u}\_x\$, the solution \$u\_x^h(x,t)\$ is constant in space and time. The pressure field is spatially constant. For stationary flow, the

pressure is zero due to the zero-traction condition on the right side; any flow acceleration would result in a constant pressure gradient and, therefore, a linear pressure distribution with zero pressure on the Neumann boundary.

To compare the two approaches, the implicit surface moves with the constant velocity  $u^i$  in positive  $x$ -direction. The interface motion introduces the problematic domain  $\Omega_+^{n+1}$ , where no old velocity values are known. The two time instances are shown in Figure 5.2. Pressure and velocity at the new time step as well as the respective test functions are enriched using the new approximation  $\phi^{n+1}$  as in Eq. (5.6) and Eq. (5.7).

### Comparison

Using the approach with old approximation functions (Eq. (5.9)) leads to the following problem: in  $\Omega_+^{n+1}$ , the integral Eq. (5.10) is zero, since

$$\bar{\psi}^n(x) = 0 \quad \text{in } \Omega_+^{n+1} \quad (5.13)$$

This implies that the velocity in  $\Omega_+^{n+1}$  accelerates from zero to  $\hat{u}_x(t)$  within one timestep, since the incompressibility constraint allows only a spatially constant velocity field for the  $n + 1$  solution. The pressure gradient balances the missing old velocity in Eq. (5.11a) such that a linear pressure distribution with zero pressure at the Neumann surface is generated. Since the pressure gradient is proportional to  $u_x^{n+1}/\Delta t$ , decreasing time step sizes do not reduce the problem. Using smaller time steps reduces the problematic domain, however, the flow in  $\Omega_+^{n+1}$  accelerates in a shorter time span and an even higher pressure gradient is generated. In other words, the error does not vanish for decreasing time step sizes. Changing the spatial resolution does not affect this conclusion.

In contrast, the Ghost-fluid method requires an estimate, what happened in the previously unobserved domain. For the example problem, the old velocity  $u^n$  in  $\Omega_{\text{Euler}}^{f,n+1}$  is used to initialize the missing fluid nodes values, since this is the only allowed solution for the incompressible one-dimensional flow at time step  $n$ . With this guess, no artificial pressure gradient is generated for any time-step size and spatial resolution. The approach that has also been used for three-dimensional computations is to extrapolate the fluid history values from the interface in normal direction until every node that requires old values has been initialized. After the necessary projection to ensure incompressibility, any time-integration scheme can be applied [Osher and Fedkiw, 2003]. In three-dimensional simulations, the extrapolation and projection may generate inaccurate history velocity and acceleration fields. Applied to the one-dimensional example, a variation of the exact old velocity in  $\Omega_+^{n+1}$  also introduces an artificial pressure gradient. However, the artificial pressure gradient is independent of the time-step size, hence, the error reduces

linearly with the time step size. For smaller time step sizes, the interface moves a smaller distance, such that the extrapolation error also reduces.

Both methods compute the same result, if the interface moves to the left. At the new time step level, the test function  $\delta u^{h,n+1}(x)$  with enrichment function  $\psi^{n+1}(x)$  does not cover the previously observed domain  $\Omega_-^{n+1}$  and, consequently, no contribution from old time steps is computed in the abandoned domain  $\Omega_-^{n+1}$ .

## Conclusion

This example suggests that the Ghost fluid approach with appropriate estimates is more suitable for moving boundaries on Eulerian grids than using multiple enrichments. Beyond the convergence aspect, respecting multiple discontinuities would also increase the implementation effort for generation of integration cells.

It is emphasized again that the reason for this apparent choice is the inappropriate use of the Eulerian formulation for moving boundaries, not the XFEM formulation. Such choice is also present in other XFEM implementations, see for instance the solidification problem presented in [Chessa et al., 2002]. There, a Lagrangian formulation is used, while a moving interface represents changing material properties. However, the strong form in Lagrangian description is established assuming a fixed reference in material coordinates with time-invariant domain boundaries. It could be argued that the computational domain is a fixed domain and a transient discontinuous function is described in a non-changing reference. However, most XFEM methods use integrals that respect the discontinuity by using integration cells, hence, a time-dependent integral boundary still exists and the argument does not solve the inappropriate use of the Eulerian or Lagrangian method for moving interfaces. Hence, using the Lagrangian approach to interfaces that move with respect to the material coordinates seems questionable in the same way as using the Eulerian formulation for moving interfaces. Nevertheless, as the following results and the results from the cited literature suggest [e.g. Hong et al., 2007; Osher and Fedkiw, 2003], using the Eulerian formulation still allows to implement a consistent numerical method, as long as reasonable guesses for  $\Omega_+^{n+1}$  are possible. Likewise, using only the new interface position for the approximation function and estimating old time step values properly, the method presented in [Réthoré et al., 2005] leads to a converging and energy conserving method for crack propagation in Lagrangian formulation.

Sometimes, the discussed problems remain unnoticed, if the difference between the fields on each side of the moving interface are relatively small. Using old time step values from the other sides fluid, for example in two

phase flow or for FSI problems involving thin structures with continuous normal velocities, the velocity difference between each side of the discontinuity might be small enough such that above observations do not lead to such prohibitive errors as in the presented one-dimensional example. However, the incompressible NS combined with an one-sided strong discontinuity penalizes small inaccuracies with strong pressure oscillations. The pressure oscillations hamper the intended FSI application, since the pressure contributes strongly to the FSI forces applied on the structure surface.

Rigorous studies on conservation of mass and momentum that depend on the projection and the extrapolation, are still missing. As an alternative, the FG/ALE approach presented in [Codina et al., 2009] may—after the required projection—provide a better old time step value on the new domain configuration  $\Omega^{n+1}$ . Eventually, enriched space-time methods provide mathematically the cleanest way, but their implementation and computational extra costs may still be to high for large scale computational fluid dynamics (CFD) computations. Hence, the Ghost fluid approach is the preferred choice for the present fixed-grid scheme and has been exclusively used. The numerical experience is that stable solutions can be obtained for time-integration that include numerical damping (BDF2 and OST with  $\theta > 0.5$ ).

### 5.2.4 Implementation

The algorithmic implementation is shown in Algorithm 5.1. The setup phase for moving boundaries is given in Algorithm 5.2. Subsequently, the Newton-Raphson iteration is performed as before (Algorithm 4.3).

The following pressure-stabilized Petrov Galerkin (PSPG) stabilization is added to the projection in Eq. (5.4)

$$\sum_e \tau_{\text{PSPG}}^e (\nabla \delta p, \nabla p)_{\Omega_e^{\text{f},n+1}} \quad (5.14)$$

The value for the stabilization parameter  $\tau_{\text{PSPG}}^e$  is the same as for the flow computations, see Eq. (2.36).

---

#### Algorithm 5.1 Ghost-fluid update at beginning of timestep $n + 1$

---

allocate unknown vectors with number of DOF required at time step  $n+1$   
 extrapolate  $\mathbf{u}^{\text{h},n}$ ,  $\mathbf{u}^{\text{h},n-1}$  and  $\mathbf{a}^{\text{h},n}$  to all nodes without history values  
 select set of nodes that contribute to non-incompressible discrete velocities  
 project velocity  $\mathbf{u}$  to incompressible space (Eqs. (5.4))

---

---

**Algorithm 5.2 XFEM setup phase for moving boundaries**

---

```

XFEM setup phase → Algorithm 4.2
if interface moves then
    extrapolate DOFs from old to new DOF distribution → Algorithm 5.1
end if

```

---

## 5.3 Treatment of moving walls

In the previous section, movement of the fluid material and movement of the interface had on purpose not been related to each other. The velocity of the interface  $\mathbf{u}^i$  has been independent of the fluid material velocity  $\mathbf{u}^f$  at the interface. Accordingly, the extrapolated fluid material velocity has been independent of the speed of the interface itself.

Only when the interface represents an impermeable wall, the fluid material velocity and the interface velocity are equal as requested in Eq. (2.66) and Eq. (2.68).

### 5.3.1 Interface movement

The interface deformation is described by the Lagrangian formulation as introduced in Section 2.3.2. In particular, the interface deformation is described by a displacement  $\mathbf{d}^i(\mathbf{X}, t)$  and a corresponding velocity  $\mathbf{u}^i(\mathbf{X}, t)$ . For the Lagrangian formulation, the material time derivative  $\dot{\mathbf{d}}^i = \mathbf{u}^i$  reduces to a partial derivative

$$\frac{\partial \mathbf{d}^i(\mathbf{X}, t)}{\partial t} = \mathbf{u}^i(\mathbf{X}, t) \quad (5.15)$$

The interface velocity  $\mathbf{u}^i$  is discretized in time by the one-step- $\theta$  method such that

$$\frac{\mathbf{d}^{i,n+1} - \mathbf{d}^{i,n}}{\Delta t} = \theta^i \mathbf{u}^{i,n+1} + (1 - \theta^i) \mathbf{u}^{i,n} \quad (5.16)$$

With the help of Eq. (5.16), the interface velocity  $\mathbf{u}^{i,n+1}$  can be expressed by the given interface displacement  $\hat{\mathbf{d}}^{i,n+1}$

$$\mathbf{u}^{i,n+1} = \frac{1}{\Delta t \theta^i} (\mathbf{d}^{i,n+1} - \mathbf{d}^{i,\text{hist}}) \quad (5.17)$$

with the old time step (or history) values defined as

$$\mathbf{d}^{i,\text{hist}} = \mathbf{d}^{i,n} + \Delta t (1 - \theta^i) \mathbf{u}^{i,n} \quad (5.18)$$

In the presented simulations, the simulation starts from a resting state, that is the initial interface displacement  $\mathbf{d}^{i,0}$  and the initial velocity  $\mathbf{u}^{i,0}$  are zero.

Note that  $\theta^i$  can be chosen independently from the fluid time integration factor  $\theta$ . For second order accuracy,  $\theta^i$  has to be chosen as  $\theta^i = 0.5$  resulting in the undamped, second order trapezoidal rule. The combination of coarse FE meshes and large time step sizes may introduce intermittent oscillations, which have their source for example in inexact extrapolation of history values as discussed in the previous section. In such cases, using  $\theta^i = 0.66$  or even a Backward-Euler method ( $\theta^i = 1$ ) damps intermittent oscillations, but also limits the overall temporal order to first order [Förster et al., 2006]. For the presented simulation,  $\theta^i = 0.5$  has been used, if not indicated otherwise.

The spatial discretization of the interface mesh is given in Eq. (2.73).

### 5.3.2 Coupling of fluid velocity with interface displacement

Using Eq. (5.17) and Eq. (5.18), it is now possible to insert  $\mathbf{u}^{i,n+1}(\mathbf{d}^{i,n+1})$  into the fluid residual equations Eq. (4.16c) and Eq. (4.32c).

**Traction based Lagrange multiplier** For the traction based Lagrange multiplier approach, the discrete system to solve is

$$\delta\mathbf{u} \mathbf{r}_u(\mathbf{u}, \mathbf{p}) + \delta\mathbf{u} \mathbf{r}_u^\lambda(\boldsymbol{\lambda}) = F_M(\delta\mathbf{u}^h, \mathbf{u}^h, p^h) - (\delta\mathbf{u}^h, \boldsymbol{\lambda}^h)_{\Gamma^i} \quad (5.19a)$$

$$\delta\mathbf{p} \mathbf{r}_p(\mathbf{u}, \mathbf{p}) = F_C(\delta p^h, \mathbf{u}^h, p^h) \quad (5.19b)$$

$$\delta\boldsymbol{\lambda} \mathbf{r}_\lambda^u(\mathbf{u}) + \delta\boldsymbol{\lambda} \mathbf{r}_\lambda^{d^i}(\mathbf{d}^i) = -(\delta\boldsymbol{\lambda}^h, \mathbf{u}^h - \frac{1}{\Delta t \theta^i} (\mathbf{d}^{i,h} - \mathbf{d}^{i,h,\text{hist}}))_{\Gamma^i} \quad (5.19c)$$

$$\delta\mathbf{d}^i \mathbf{r}_{d^i}^\lambda(\boldsymbol{\lambda}) = (\delta\mathbf{d}^{i,h}, \boldsymbol{\lambda}^h)_{\Gamma^i} \quad (5.19d)$$

The linearization process is almost unchanged. The only difference is that the interface displacement is now the interface unknown.

$$\begin{bmatrix} \mathbf{F}_{uu} & \mathbf{F}_{up} & \mathbf{M}_{u\lambda} & \mathbf{0} \\ \mathbf{F}_{pu} & \mathbf{F}_{pp} & \mathbf{0} & \mathbf{0} \\ \mathbf{M}_{\lambda u} & \mathbf{0} & \mathbf{0} & \mathbf{D}_{\lambda d^i} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_{d^i \lambda} & \mathbf{0} \end{bmatrix}_k \begin{bmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{p} \\ \Delta\boldsymbol{\lambda} \\ \Delta\mathbf{d}^i \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_u + \mathbf{r}_u^\lambda \\ \mathbf{r}_p \\ \mathbf{r}_\lambda^u + \mathbf{r}_\lambda^{d^i} \\ \mathbf{r}_{d^i}^\lambda \end{bmatrix}_k \quad (5.20)$$

with  $\mathbf{M}_{u\lambda} = \mathbf{M}_{\lambda u}^T$  as given in Eq. (4.18) and

$$\mathbf{D}_{\lambda d^i} = \frac{1}{\Delta t \theta^i} \mathbf{D}_{\lambda u i} \quad (5.21)$$

$$\mathbf{D}_{d^i \lambda} = \mathbf{D}_{u^i \lambda} \quad (5.22)$$

If the interface displacement is given and prescribed as a Dirichlet condition for all interface displacement unknowns  $\mathbf{d}$ , column and row of the system matrix can be omitted and the known interface displacement and the time

history terms only appear in the residual  $r_{\lambda}^{d^i}(\mathbf{u}, \hat{\mathbf{d}}^i)$  with  $\hat{\mathbf{d}}^i$  being the given interface displacement. The resulting reduced system is then

$$\begin{bmatrix} F_{uu} & F_{up} & M_{u\lambda} \\ F_{pu} & F_{pp} & \mathbf{0} \\ M_{\lambda u} & \mathbf{0} & \mathbf{0} \end{bmatrix}_k \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{p} \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} r_u + r_u^\lambda \\ r_p \\ r_\lambda^u + r_\lambda^{d^i} \end{bmatrix}_k \quad (5.23)$$

**Stress based Lagrange multiplier** The discrete system is posed assuming again that the interface displacement is unknown

$$\delta \mathbf{u} \cdot r_u(\mathbf{u}, \mathbf{p}) + \delta \mathbf{u} \cdot r_{\bar{\sigma}}(\bar{\sigma}, \mathbf{d}^i) = F_M(\delta \mathbf{u}^h, \mathbf{u}^h, p^h) - (\delta \mathbf{u}^h, \bar{\sigma}^h \cdot \mathbf{n}^f)_{\Gamma^i} \quad (5.24a)$$

$$\delta \mathbf{p} \cdot r_p(\mathbf{u}, \mathbf{p}) = F_C(\delta p^h, \mathbf{u}^h, p^h) \quad (5.24b)$$

$$\begin{aligned} \delta \bar{\sigma} \cdot r_{\bar{\sigma}}^{d^i}(\mathbf{u}, \mathbf{p}, \bar{\sigma}, \mathbf{d}^i) &= - (\delta \bar{\sigma}^h, \bar{\gamma}(\bar{\sigma}^h, p^h) - \gamma(\mathbf{u}^h))_{\Omega^f} \\ &\quad - (\delta \bar{\sigma}^h \cdot \mathbf{n}^f, \mathbf{u}^h - \frac{1}{\Delta t \theta^i} (\mathbf{d}^{i,h} - \mathbf{d}^{i,h,\text{hist}}))_{\Gamma^i} \end{aligned} \quad (5.24c)$$

$$\delta \mathbf{d}^i \cdot r_{d^i}(\bar{\sigma}) = (\delta \mathbf{d}^{i,h}, \bar{\sigma}^h \cdot \mathbf{n}^f)_{\Gamma^i} \quad (5.24d)$$

Linearization at Newton-Raphson iteration step  $k$  is performed with respect to the discrete unknowns  $\mathbf{u}$ ,  $\mathbf{p}$ ,  $\bar{\sigma}$  and  $\mathbf{d}^i$ . With changing interface position the interface normal changes as well. If the displacement is treated as unknown field the dependency of the normal with respect to the displacement has to be linearized accordingly. In particular, the boundary integral in the stress residual is linearized

$$\begin{aligned} &\frac{\partial - (\delta \bar{\sigma}^h \cdot \mathbf{n}^f, \mathbf{u}^h - \frac{1}{\Delta t \theta^i} (\mathbf{d}^{i,h} - \mathbf{d}^{i,h,\text{hist}}))_{\Gamma^i}}{\partial \mathbf{d}^i} \\ &= \underbrace{- (\delta \bar{\sigma}^h \cdot \mathbf{n}^f, - \frac{1}{\Delta t \theta^i} \frac{\partial \mathbf{d}^{i,h}}{\partial \mathbf{d}^i})_{\Gamma^i}}_{\delta \bar{\sigma} \mathbf{G}_{\bar{\sigma} d^i}} - \underbrace{(\delta \bar{\sigma}^h \cdot \frac{\partial \mathbf{n}^f(\mathbf{d}^{i,h})}{\partial \mathbf{d}^i}|_k, \mathbf{u}_k^h - \frac{1}{\Delta t \theta^i} (\mathbf{d}_k^{i,h} - \mathbf{d}^{i,h,\text{hist}}))_{\Gamma^i}}_{\delta \bar{\sigma} \mathbf{G}_{\bar{\sigma} d^i}^n} \end{aligned} \quad (5.25)$$

The first matrix part is similar to the linearization in Eq. (4.36). The only difference is the factor from the interface time discretization

$$\mathbf{G}_{\bar{\sigma} d^i} = \frac{1}{\Delta t \theta^i} \mathbf{G}_{\bar{\sigma} u^i} \quad (5.26)$$

The second matrix part stems from the displacement dependent normal vector. All remaining boundary terms in Eq. (5.24a) and Eq. (5.24d) that contain surface normals have to be linearized accordingly. The resulting tangent ma-

trices are named  $\mathbf{G}_{ud^i}^n$  and  $\mathbf{G}_{d^i d^i}^n$ . The complete discrete system is therefore

$$\mathbf{A}_e \begin{bmatrix} \mathbf{F}_{uu} & \mathbf{F}_{up} & \mathbf{G}_{u\bar{\sigma}} & \mathbf{G}_{ud^i}^n \\ \mathbf{F}_{pu} & \mathbf{F}_{pp} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{\bar{\sigma}u} + \mathbf{G}_{\bar{\sigma}u} & \mathbf{K}_{\bar{\sigma}p} & \mathbf{K}_{\bar{\sigma}\bar{\sigma}} & \mathbf{G}_{\bar{\sigma}d^i} + \mathbf{G}_{\bar{\sigma}d^i}^n \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_{d^i\bar{\sigma}} & \mathbf{G}_{d^i d^i}^n \end{bmatrix}_k \mathbf{A}_e \begin{bmatrix} \Delta \mathbf{u}_e \\ \Delta \mathbf{p}_e \\ \Delta \bar{\sigma}_e \\ \Delta d^i_e \end{bmatrix} = -\mathbf{A}_e \begin{bmatrix} \mathbf{r}_u + \mathbf{r}_{\bar{\sigma}} \\ \mathbf{r}_p \\ \mathbf{r}_{\bar{\sigma}} \\ \mathbf{r}_{d^i} \end{bmatrix}_k \quad (5.27)$$

For elementwise condensation, the stress increments of one element can again be expressed as

$$\Delta \bar{\sigma}_e = (\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^e)^{-1} (-\mathbf{r}_{\bar{\sigma}}^{d^i,e} - (\mathbf{K}_{\bar{\sigma}u}^e + \mathbf{G}_{\bar{\sigma}u}^e) \Delta \mathbf{u}_e - \mathbf{K}_{\bar{\sigma}p}^e \Delta \mathbf{p}_e - (\mathbf{G}_{\bar{\sigma}d^i}^e + \mathbf{G}_{\bar{\sigma}d^i}^{n,e}) \Delta d^i_e) \quad (5.28)$$

As before, the element stress couples into the momentum equations for fluid and interface only via the matrices  $\mathbf{G}_{u\bar{\sigma}}$  and  $\mathbf{G}_{d^i\bar{\sigma}}$ , respectively. In preparation, the following element matrices  $\mathbf{C}^e$  are defined for the fluid velocity unknowns

$$\mathbf{C}_{ud^i}^e = -\mathbf{G}_{u\bar{\sigma}}^e (\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^e)^{-1} (\mathbf{G}_{\bar{\sigma}d^i}^e + \mathbf{G}_{\bar{\sigma}d^i}^{n,e}) + \mathbf{G}_{ud^i}^n \quad (5.29)$$

and, similarly, for the interface displacement unknowns

$$\mathbf{C}_{d^i d^i}^e = -\mathbf{G}_{d^i\bar{\sigma}}^e (\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^e)^{-1} (\mathbf{G}_{\bar{\sigma}d^i}^e + \mathbf{G}_{\bar{\sigma}d^i}^{n,e}) + \mathbf{G}_{d^i d^i}^n \quad (5.30)$$

The corresponding residual terms based on the stress residual are defined as

$$\mathbf{r}_u^{d^i,e} = -\mathbf{G}_{u\bar{\sigma}}^e (\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^e)^{-1} \mathbf{r}_{\bar{\sigma}}^{d^i,e} \quad (5.31)$$

$$\mathbf{r}_{d^i}^e = -\mathbf{G}_{d^i\bar{\sigma}}^e (\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^e)^{-1} \mathbf{r}_{\bar{\sigma}}^{d^i,e} \quad (5.32)$$

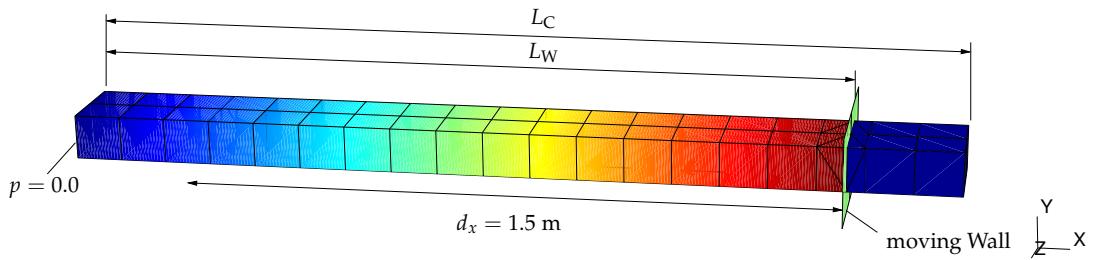
The remaining condensed matrices are defined in Section 4.3.2. The system after stress condensation is then

$$\mathbf{A}_e \begin{bmatrix} \mathbf{F}_{uu}^e + \mathbf{C}_{uu}^e & \mathbf{F}_{up}^e + \mathbf{C}_{up}^e & \mathbf{C}_{ud^i}^e \\ \mathbf{F}_{pu}^e & \mathbf{F}_{pp}^e & \mathbf{0} \\ \mathbf{C}_{d^i u}^e & \mathbf{C}_{d^i p}^e & \mathbf{C}_{d^i d^i}^e \end{bmatrix}_k \mathbf{A}_e \begin{bmatrix} \Delta \mathbf{u}_e \\ \Delta \mathbf{p}_e \\ \Delta d^i_e \end{bmatrix} = -\mathbf{A}_e \begin{bmatrix} \mathbf{r}_u^e + \mathbf{r}_{\bar{\sigma}}^{d^i,e} + \mathbf{r}_{d^i}^e \\ \mathbf{r}_p^e \\ \mathbf{r}_{\bar{\sigma}}^{d^i,e} + \mathbf{r}_{d^i}^e \end{bmatrix}_k \quad (5.33)$$

If the interface displacement is explicitly given, the last column and row of the system matrix can be omitted and all derivatives of the normal vector with respect to the interface displacements vanish. The reduced system with separately assembled fluid and condensed coupling matrices is given as

$$\begin{bmatrix} \mathbf{F}_{uu} + \mathbf{C}_u & \mathbf{F}_{up} + \mathbf{C}_{up} \\ \mathbf{F}_{pu} & \mathbf{F}_{pp} \end{bmatrix}_k \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{p} \end{bmatrix} = -\begin{bmatrix} \mathbf{r}_u + \mathbf{r}_{\bar{\sigma}} + \mathbf{r}_{d^i} \\ \mathbf{r}_p \end{bmatrix}_k \quad (5.34)$$

This system is used for the partitioned FSI implementation in the following chapter.



**Figure 5.3:** Force solution for various time-integration schemes for  $\Delta t = 0.05$  s

### 5.3.3 Example

A simple example using the Ghost-fluid method and stress Lagrange multiplier verifies the correctness of the proposed scheme. The fluid domain setup is shown in Figure 5.3 and is similar to the setup in Section 4.5.2. However, no body force is applied, instead the interface including wall boundary conditions is moved in an oscillatory way with prescribed interface displacement given by

$$d_x^i(t) = -\hat{d}_x^i \frac{1}{2} (1 - \cos(2\pi \frac{t}{T})) \quad (5.35)$$

The oscillation amplitude is denoted by  $\hat{d}_x^i$  and the periodic time by  $T$ . The resulting velocity  $u_x^i$  and acceleration  $a_x^i$  of the interface is therefore

$$u_x^i(t) = -\hat{d}_x^i \frac{1}{2} \sin(2\pi \frac{t}{T}) \frac{2\pi}{T} \quad (5.36)$$

$$a_x^i(t) = -\hat{d}_x^i \frac{1}{2} \cos(2\pi \frac{t}{T}) \frac{2\pi}{T} \frac{2\pi}{T} \quad (5.37)$$

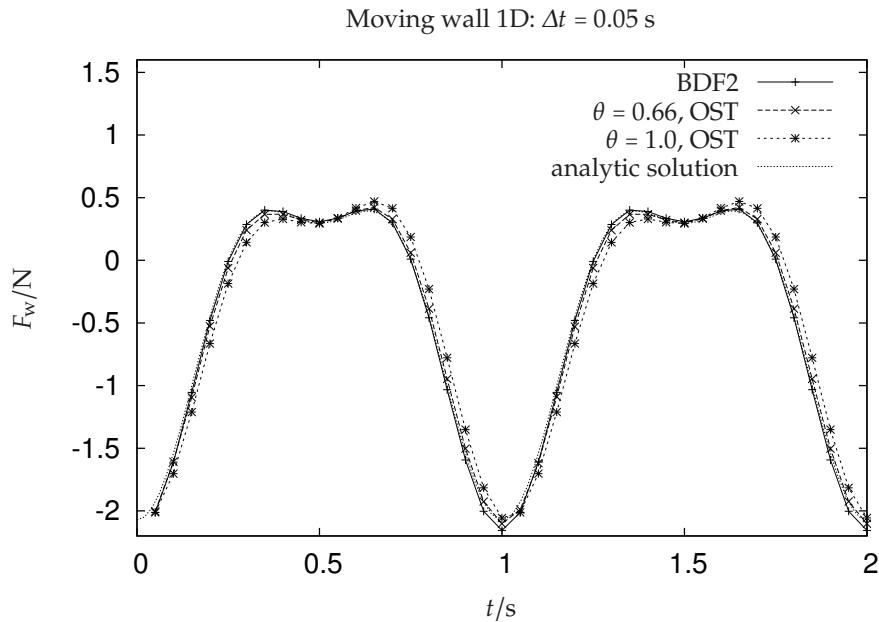
The channel size is again  $2.0 \text{ m} \times 0.2 \text{ m} \times 0.1 \text{ m}$  and the amplitude was chosen as  $\hat{d}_x^i = 1.5 \text{ m}$ . The initial position of the wall is again  $L_w^0 = 1.75 \text{ m}$ , however  $L_w$  is now a function of  $t$  as

$$L_w(t) = L_w^0 + d(t) \quad (5.38)$$

The wet surface, that interacts with the fluid is therefore  $A_w = 0.2 \text{ m} \times 0.1 \text{ m}$ . The prescribed oscillation has a periodic time  $T = 1 \text{ s}$ .

The left side of the channel is again a zero-traction boundary. The fluid is pushed out of the channel and pulled in again by the moving wall. The flow speed is constant in space and should be the same as the speed of the moving wall. As the spatial solution can be represented exactly, no discretization error occurs. The force on the moving wall is solely determined by the pressure as given in Eq. (4.67) and the resulting wall force  $F_{\text{wall}}^h$  is computed from the discrete interface nodal forces  $f_K^i$  as

$$F_w^h = \sum_K f_K^i \quad (5.39)$$



**Figure 5.4:** Force solution for three time-integration schemes for  $\Delta t = 0.05$  s

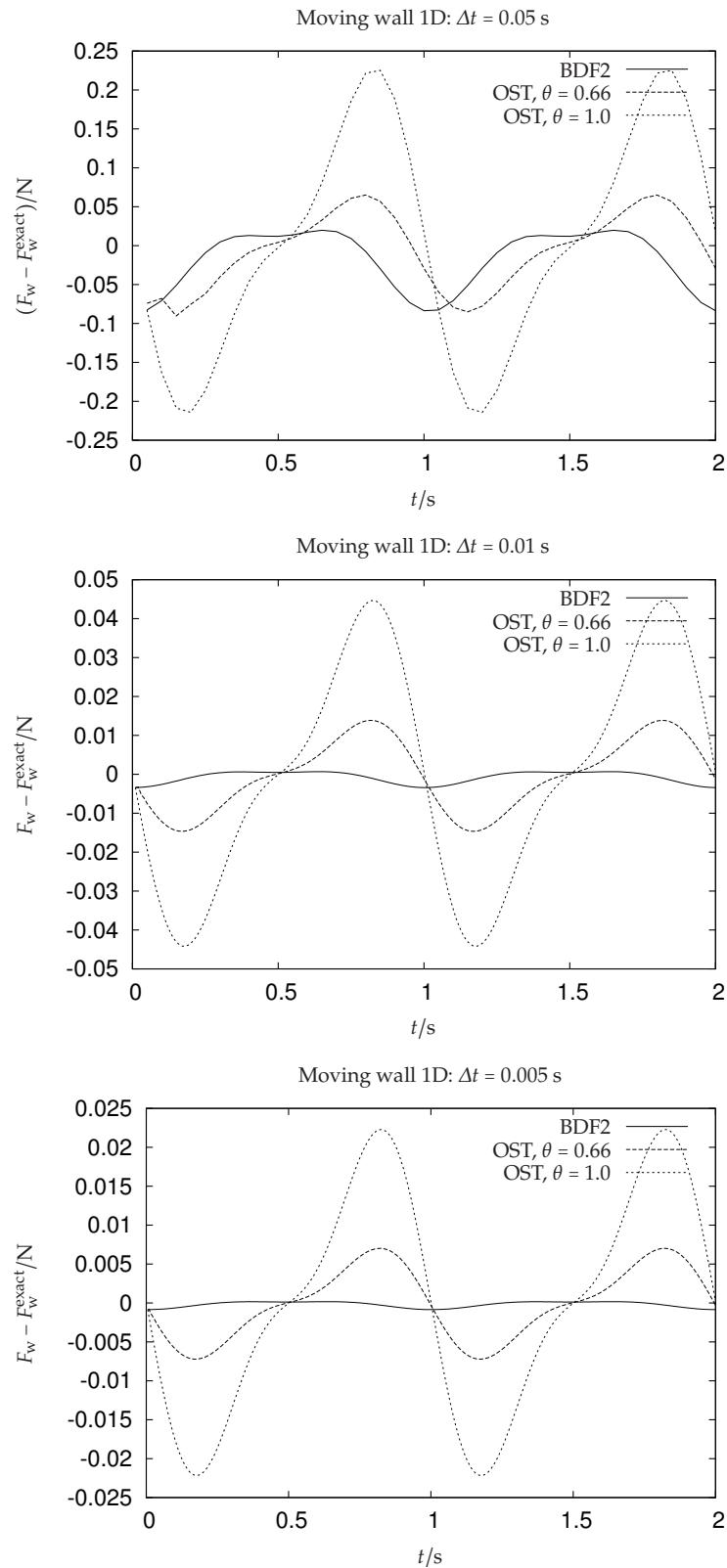
The interface nodal forces  $f_K^i$  are computed as defined in Eq. (4.66). Since for this example the acceleration is proportional to the pressure gradient only, an analytic solution for  $F_w$  can be given as

$$\begin{aligned}
 F_w^{\text{exact}}(t) &= \int_{\Gamma_i} \boldsymbol{\sigma} \cdot \mathbf{n}^f d\mathbf{x} \\
 &= A_w p \\
 &= A_w \frac{\partial p(t)}{\partial x} L_w(t) \\
 &= A_w \rho^f a_x^i(t) L_w(t)
 \end{aligned} \tag{5.40}$$

For starting the simulation, a backward Euler step ( $\theta = 1.0$ ) was used in the first time step. The wall displacement was prescribed as given in Eq. (5.35), however, the wall velocity was computed using Eq. (5.17) with  $\theta^i = 0.5$  instead of the analytical velocity. For the initial step  $\theta^i = 1.0$  was used.

Three time-stepping schemes, namely the BDF2 scheme and the OST with  $\theta = 0.66$  and  $\theta = 1.0$ , have been tested for their convergence properties. Figure 5.4 shows the result of the computed wall force compared to the analytical solution for time step sizes  $\Delta t = 0.05$  s. For smaller time steps, the three solutions become indistinguishable. Thus, in Figure 5.5 the difference of the solution to the analytical solutions for three different time step sizes have been plotted.

Here, the lagging of the first order schemes behind the correct solution is clearly visible. The initial wiggles due to the start up by a backward Euler method can be observed for  $\theta = 0.66$ . These wiggles are subsequently



**Figure 5.5:** Force difference between computed and analytical wall force

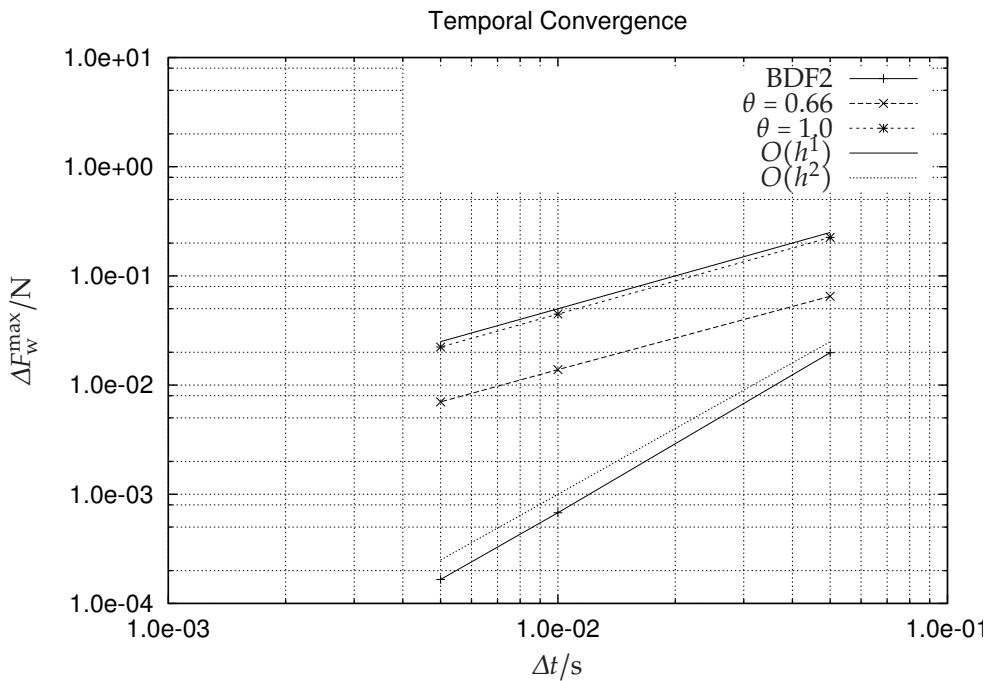


Figure 5.6: Temporal convergence for three time-integration schemes

damped by the diffusive behavior of the damped OST scheme. It is also visible, that the second order BDF2 scheme converges faster than the first order schemes, where the backward Euler scheme has the largest error when compared with the other schemes. The convergence diagram in Figure 5.6 confirms the observed convergence behavior. For the temporal convergence diagram, the maximal force difference between computed and exact solution has been plotted over the time step size. The maximal force difference occurs for each scheme differently in the range between  $t = 0.7$  s and  $t = 0.8$  s. Alternatively, the force difference at a fixed point in time for all three schemes can be chosen, which results in the same convergence behavior.

It can be seen that all schemes converge with optimal order. For the OST method, only  $\theta = 0.5$  would produce second order convergence. However,  $\theta = 0.5$  removes numerical diffusion from the OST time integration and initial errors introduced by the backward Euler step are undamped and eventually lead to divergence. Using  $0.5 < \theta \leq 1$  performs better than a pure backward Euler method, but only first order convergence is obtained. Thus for most numerical results, the BDF2 has been employed. Note that the achieved second order convergence depends also on the second order scheme that has been used for the interface velocity ( $\theta^i = 0.5$ ). With a first order interface displacement time discretization, the overall order would be limited to first order. This finding is in agreement with the discussion on FSI time integration schemes in [Fürster, 2007].

This example shows that temporal convergence is optimal, if the correct old velocities have been estimated. A rigorous study that includes the effect of erroneous estimates is still missing. More complex examples without analytic solutions should be tested to show, whether the convergence is disturbed by using faulty guesses for the old time-step fields.



# CHAPTER 6

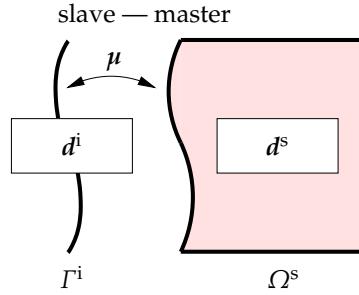
## Coupled fluid-structure system

The fixed-grid XFSI system is the combination of the derived single field approaches and their coupling the additional interface mesh. For integrating the single field approaches, first a weak interface-structure coupling using Mortar techniques is introduced. Then the monolithic and partitioned FSI schemes are described for both traction and stress Lagrange multipliers, where the new stress Lagrange multiplier technique simplifies and improves both the monolithic and the partitioned coupling, when compared with the traction Lagrange multiplier approach. Finally, extensions towards improved fluid solutions near the interface are presented and examples illustrate versatility of the XFSI approach.

### 6.1 Interface-structure coupling

As the most general case, the interface mesh and the structure surface use independent displacement approximations. Hence, a coupling mechanism for non-matching boundary meshes is required. For this problem, the Mortar methods for non-matching grids [Bernardi et al., 1993, 1994] is adopted.

In the Mortar method, the kinematic matching condition between interface and structure surface in Eq. (2.69) is enforced weakly by using a Lagrange multiplier field  $\mu$ . The Lagrange multiplier corresponds to the surface traction  $\mu$  defined in Eq. (2.71). The setup for the structure-interface coupling is depicted in Figure 6.1. The strong form of the structural momentum equation including the interface condition in Eq. (2.69) with the three primary un-



**Figure 6.1:** Mortar coupling between structural surface and interface

knowns  $d^s$ ,  $d^i$  and  $\mu$  is given as

$$\mathbf{r}_{ds}(d^s) = \mathbf{0} \quad \text{in } \Omega^s \quad (6.1a)$$

$$d^s - d^i = \mathbf{0} \quad \text{in } \Gamma^i \quad (6.1b)$$

$$\underline{\sigma}^s \cdot \mathbf{n}^s - \mu = \mathbf{0} \quad \text{in } \Gamma^i \quad (6.1c)$$

$$d^s - \hat{d}_D = \mathbf{0} \quad \text{in } \Gamma_D^s \quad (6.1d)$$

$$\underline{\sigma}^s \cdot \mathbf{n}^s - \hat{h}^s = \mathbf{0} \quad \text{in } \Gamma_N^s \quad (6.1e)$$

Here the structure surface  $\Gamma^s$  is split into  $\Gamma_D^s$  with Dirichlet conditions,  $\Gamma_N^s$  with Neumann conditions and  $\Gamma^i$ , where the FSI conditions are located.

The weighted residual form of Eq. (6.1) reads as

$$0 = (\delta d^s, \mathbf{r}_{ds}(d^s))_{\Omega^s} + (\delta d, \underline{\sigma}^s \cdot \mathbf{n}^s - \hat{h}^s)_{\Gamma_N^s} + (\delta d, \underline{\sigma}^s \cdot \mathbf{n}^s - \mu)_{\Gamma^i} \quad (6.2a)$$

$$0 = -(\delta \mu, d^s - d^i)_{\Gamma^i} \quad (6.2b)$$

$$0 = (\delta d^i, \mu)_{\Gamma^i} \quad (6.2c)$$

The interface displacement is assumed to be unknown, hence, a test function  $\delta d^i$  is required. Since the interface does not add any physical effects, the interface traction  $\mu$  is zero as long as no fluid is coupled to the interface. Integration by parts of the stress term in Eq. (6.2a) gives

$$-(\delta d^s, \nabla \cdot \underline{\sigma}^s)_{\Omega^s} = -(\delta d^s, \underline{\sigma}^s \cdot \mathbf{n}^s)_{\Gamma^s} + (\nabla \cdot \delta d^s, \underline{\sigma}^s)_{\Omega^s}$$

With the structure weak form term  $S(\delta d^s, d^s)$  as defined in Section 2.2, the weak form with interface constraints is written as

$$0 = S(\delta d^s, d^s) - (\delta d^s, \mu)_{\Gamma^i} \quad (6.3a)$$

$$0 = -(\delta \mu, d^s - d^i)_{\Gamma^i} \quad (6.3b)$$

$$0 = (\delta d^i, \mu)_{\Gamma^i} \quad (6.3c)$$

Neumann conditions and volumetric forces are included in  $S(\delta d^s, d^s)$ .

The structural displacement is discretized in Section 2.2.2 and the interface displacement is discretized in Section 2.3.2. The corresponding structure and interface shape functions are denoted as  $N^s$  and  $N^i$ , respectively. Both approximations are repeated here for easier reference:

$$\mathbf{d}^{s,h} = \sum_I N_I^s \mathbf{d}_I^s \quad \text{and} \quad \delta \mathbf{d}^{s,h} = \sum_I N_I^s \delta \mathbf{d}_I^s \quad (6.4)$$

$$\mathbf{d}^{i,h} = \sum_I N_I^i \mathbf{d}_I^i \quad \text{and} \quad \delta \mathbf{d}^{i,h} = \sum_I N_I^i \delta \mathbf{d}_I^i \quad (6.5)$$

In the Mortar method, the Lagrange multiplier is discretized with the shape functions and node set of the so called slave side. For the partitioned FSI scheme presented later, the interface  $\Gamma^i$  has to be the slave side and the structural surface the master side. Consequently, trial and test functions of the Lagrange multiplier field are discretized with the interface shape functions as

$$\boldsymbol{\mu}^h = \sum_I N_I^i \boldsymbol{\mu}_I \quad \text{and} \quad \delta \boldsymbol{\mu}^h = \sum_I N_I^i \delta \boldsymbol{\mu}_I \quad (6.6)$$

If the interface is curved, then the structure surface elements may not coincide with the interface elements. Here, a projection  $P^\mu$  is required to project the discrete structure surface displacement onto the interface mesh to integrate the weak interface condition. Such projection is a complex three-dimensional task and is a critical ingredient in a robust Mortar implementation, see e.g. [Puso, 2004; Puso et al., 2008; Wohlmuth, 2001]. Including the projection, the discrete nonlinear system are given as

$$\delta \mathbf{d}^s [\mathbf{r}_{ds}^s(\mathbf{d}^s) + \mathbf{r}_{ds}^\mu(\boldsymbol{\mu})] = S(\delta \mathbf{d}^{s,h}, \mathbf{d}^{s,h}) - (\delta \mathbf{d}^{s,h}, \boldsymbol{\mu}^h)_{\Gamma^i} \quad (6.7a)$$

$$\delta \boldsymbol{\mu} [\mathbf{r}_\mu^{ds}(\mathbf{d}^s) + \mathbf{r}_\mu^{di}(\mathbf{d}^i)] = -(\delta \boldsymbol{\mu}^h, P^{\mu s,h} \mathbf{d}^{s,h} - \mathbf{d}^{i,h})_{\Gamma^i} \quad (6.7b)$$

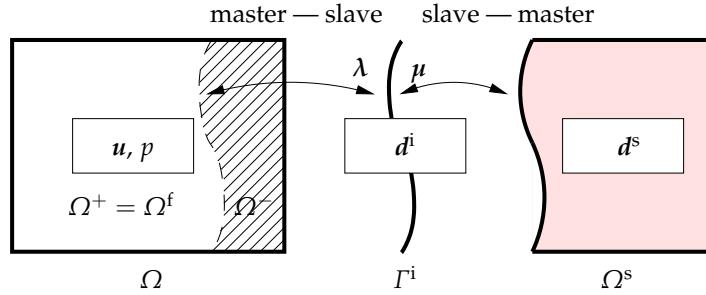
$$\delta \mathbf{d} \mathbf{r}_{di}^\mu(\boldsymbol{\mu}) = (\delta \mathbf{d}^{i,h}, \boldsymbol{\mu}^h)_{\Gamma^i} \quad (6.7c)$$

The virtual work defined in Eq. (6.7b) allows to define two coupling matrices  $\mathbf{M}_{\mu ds}$  and  $\mathbf{D}_{\mu di}$  as

$$0 = \delta \boldsymbol{\mu}_I \left[ \underbrace{\int_{\Gamma^i} N_I^i P^\mu N_J^s dx \mathbf{d}_J^s}_{\mathbf{M}_{\mu ds}} - \underbrace{\int_{\Gamma^i} N_I^i N_K^i dx \mathbf{d}_K^i}_{\mathbf{D}_{\mu di}} \right] \quad (6.8)$$

The matrix  $\mathbf{M}_{\mu ds}$  is generally rectangular because of the non-fitting surface discretizations, while  $\mathbf{D}_{\mu di}$  is a square matrix. The respective transposed matrices are denoted as  $\mathbf{M}_{ds \mu} = \mathbf{M}_{\mu ds}^T$  and  $\mathbf{M}_{di \mu} = \mathbf{M}_{\mu di}^T$ .

The mortar coupling introduces only linear relations such that the linearization of the interface structure system does not pose particular problems.



**Figure 6.2:** Coupled FSI system with traction Lagrange multipliers.

The structure system is linearized in Section 2.2. The linearized interface-structure system in incremental formulation is therefore given as

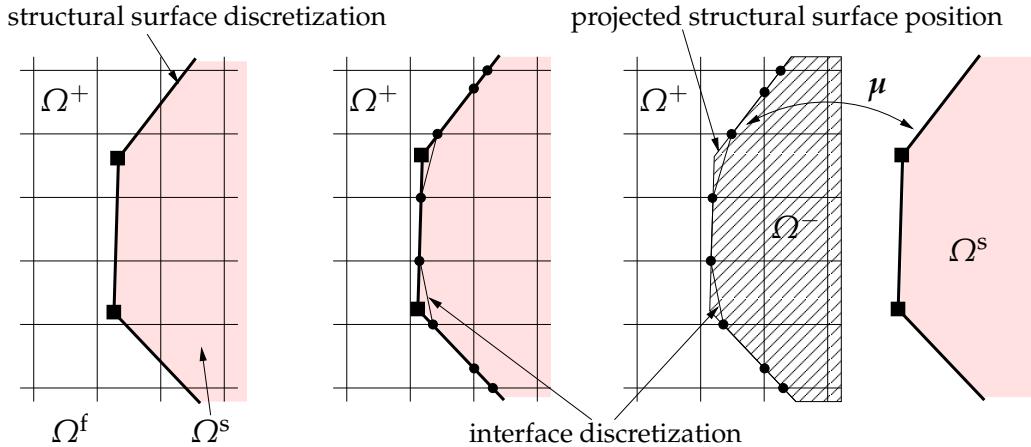
$$\begin{bmatrix} 0 & D_{d^i \mu} & 0 & 0 \\ D_{\mu d^i} & 0 & M_{\mu d^s} & 0 \\ 0 & M_{d^s \mu} & S_{d^s d^s}^{BB} & S_{d^s d^s}^{BD} \\ 0 & 0 & S_{d^s d^s}^{DB} & S_{d^s d^s}^{DD} \end{bmatrix}_k \begin{bmatrix} \Delta d^i \\ \Delta \mu \\ \Delta d_B^s \\ \Delta d_D^s \end{bmatrix} = - \begin{bmatrix} r_{d^i}^\mu \\ r_\mu^{d^i} + r_\mu^{d^s} \\ r_{d^s}^{s,B} + r_{d^s}^{\mu,B} \\ r_{d^s}^{s,D} \end{bmatrix}_k \quad (6.9)$$

Here, the structure DOFs are split into two sets: set  $\mathcal{B}$  contains all displacement unknowns attached to boundary nodes on the structure FSI surface  $\Gamma^{s,FSI,h}$  and set  $\mathcal{D}$  contains all remaining structure DOFs. The split reveals that only structure surface nodes are involved in the weak structure interface coupling. After acknowledging this fact, the following shorter writing is used

$$\begin{bmatrix} 0 & D_{d^i \mu} & 0 \\ D_{\mu d^i} & 0 & M_{\mu d^s} \\ 0 & M_{d^s \mu} & S_{d^s d^s} \end{bmatrix}_k \begin{bmatrix} \Delta d^i \\ \Delta \mu \\ \Delta d^s \end{bmatrix} = - \begin{bmatrix} r_{d^i}^\mu \\ r_\mu^{d^i} + r_\mu^{d^s} \\ r_{d^s}^s + r_{d^s}^\mu \end{bmatrix}_k \quad (6.10)$$

## 6.2 FSI system with traction Lagrange multipliers

The FSI approach using traction Lagrange multipliers is published in [Gerstenberger and Wall, 2008b]. The fully coupled FSI system using three distinct meshes in  $\Omega$ ,  $\Omega^s$  and  $\Gamma^i$ , respectively, is shown in Figure 6.2. The time-discretization may be different for fluid, structure and interface variables and has been presented separately in Section 2.1.2, 2.2.2 and 5.3.1. After fluid and structure time discretization, the coupled fluid-interface-structure system in-



**Figure 6.3:** Interface construction using the discrete structural surface: On the left, the structural surface and the fluid mesh is shown. Lagrange nodes ( $\bullet$ ) are located on the intersection between structural surface and fluid edges.

cluding the fluid stabilization becomes

$$0 = F_M(\delta \mathbf{u}, \mathbf{u}, p) - (\delta \mathbf{u}, \boldsymbol{\lambda})_{\Gamma^i} \quad (6.11a)$$

$$0 = F_C(\delta p, \mathbf{u}, p) \quad (6.11b)$$

$$0 = -(\delta \boldsymbol{\lambda}, \mathbf{u} - \frac{1}{\Delta t \theta^i} (\mathbf{d}^{i,h} - \mathbf{d}^{i,h,\text{hist}}))_{\Gamma^i} \quad (6.11c)$$

$$0 = (\delta \mathbf{d}^i, \boldsymbol{\lambda} + \boldsymbol{\mu})_{\Gamma^i} \quad (6.11d)$$

$$0 = -(\delta \boldsymbol{\mu}, \mathbf{d}^s - \mathbf{d}^i)_{\Gamma^i} \quad (6.11e)$$

$$0 = S(\delta \mathbf{d}^s, \mathbf{d}^s) - (\delta \mathbf{d}^s, \boldsymbol{\mu})_{\Gamma^i} \quad (6.11f)$$

If the traction based Lagrange multiplier for fluid-interface coupling is employed as described in Section 4.3.1, the fluid-interface coupling dictate the interface discretization. In particular, the interface discretization was derived from the intersection points of a smooth interface crossing the fluid element edges. In the XFSI approach, the surface mesh of the Lagrangian structure defines the location of the interface. Consequently, the interface is constructed from the intersection of structural surface edges with fluid element edges as shown in Figure 6.3. For moving structures, the interface mesh has to be recreated in every nonlinear FSI step. The structural surface position is used for the creation of integration cells.

In the fluid-interface-structure system, the interface hosts both Lagrange multiplier fields  $\boldsymbol{\lambda}$  and  $\boldsymbol{\mu}$  and the interface displacement  $\mathbf{d}^i$ . In other words, the interface is the slave side for both, the fluid-interface coupling and the structure-interface coupling. The three interface variables  $\boldsymbol{\lambda}$ ,  $\mathbf{d}^i$  and  $\boldsymbol{\mu}$  have the same number of nodes and use the same approximation functions  $N_I^i$ .

Hence, matrix  $D_{d^i\mu}$  is identical to  $D_{d^i\lambda}$ . The discrete equivalent of Eq. (2.72) is

$$D_{d^i\lambda} \boldsymbol{\lambda} + D_{d^i\mu} \boldsymbol{\mu} = \mathbf{0} \quad (6.12)$$

which reduces to

$$\boldsymbol{\lambda} = -\boldsymbol{\mu} \quad (6.13)$$

The linearized monolithic system is a combination of the linear systems in Eq. (4.17) and Eq. (6.10) with all linearized matrix entries defined as before

$$\begin{bmatrix} F_{uu} & F_{up} & M_{u\lambda} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ F_{pu} & F_{pp} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ M_{\lambda u} & \mathbf{0} & \mathbf{0} & D_{\lambda d^i} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & D_{d^i\lambda} & \mathbf{0} & D_{d^i\mu} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & D_{\mu d^i} & \mathbf{0} & M_{\mu d^s} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & M_{d^s\mu} & S_{d^s d^s} \end{bmatrix}_k \begin{bmatrix} \Delta u \\ \Delta p \\ \Delta \lambda \\ \Delta d^i \\ \Delta \mu \\ \Delta d^s \end{bmatrix} = - \begin{bmatrix} r_u + r_u^\lambda \\ r_p \\ r_\lambda^\lambda + r_\mu^{d^i} \\ r_{d^i}^{d^i} + r_{d^s}^{d^s} \\ r_\mu^{d^i} + r_\mu^{d^s} \\ r_{d^s}^s + r_{d^s}^\mu \end{bmatrix}_k \quad (6.14)$$

Correct scaling of coupling matrices using Eq. (5.16) allows to solve for velocities in the fluid field and displacements on the interface mesh and the structural domain in one monolithic system. The necessary scaling factors to couple displacements and velocities are contained within the coupling matrices  $D_{\lambda d^i}$  and the interface residual terms as shown in Section 5.3.2.

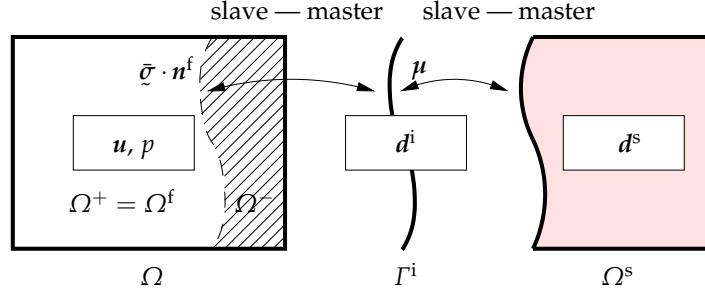
As discussed in Section 4.3.1, the traction Lagrange multiplier approach between fluid and interface is difficult to extend to three-dimensional problems. Hence, the presented monolithic approach has been used only for two-dimensional simulations as given in the example section. For three-dimensional simulations, the new coupling approach is used in the following.

### 6.3 FSI system with stress Lagrange multipliers

The stress based Lagrange multiplier approach replaces the traction  $\lambda$  by the hybrid stress formulation. Since the stress field is defined on the background grid, the background fluid grid is termed the slave side of the fluid-interface coupling<sup>1</sup>. This setup is shown in Figure 6.4. The corresponding weak form

---

<sup>1</sup>Even though the stress field approximation is different from velocity and pressure approximation, the element based shape functions are defined on the same grid as the fluid. Hence, the term ‘slave side’ seems appropriate.



**Figure 6.4:** Coupled FSI system with proposed stress Lagrange multipliers.

after time discretization is

$$0 = F_M(\delta u, u, p) - (\delta u, \bar{\sigma} \cdot \mathbf{n}^f)_{\Gamma^i} \quad (6.15a)$$

$$0 = F_C(\delta p, u, p) \quad (6.15b)$$

$$0 = -(\delta \bar{\sigma}, \bar{\gamma} - \gamma)_{\Omega^f} - (\delta \bar{\sigma} \cdot \mathbf{n}^f, u - \frac{1}{\Delta t \theta^i} (d^{i,h} - d^{i,h,hist}))_{\Gamma^i} \quad (6.15c)$$

$$0 = (\delta d^i, \bar{\sigma} \cdot \mathbf{n}^f + \mu)_{\Gamma^i} \quad (6.15d)$$

$$0 = -(\delta \mu, d^s - d^i)_{\Gamma^i} \quad (6.15e)$$

$$0 = S(\delta d^s, d^s) - (\delta d^s, \mu)_{\Gamma^i} \quad (6.15f)$$

Almost all ingredients of this weak form have already been defined in Section 5.3.2 and Section 6.1. The only addition is the weak traction balance Eq. (6.15d), which is introduced by testing the strong form in Eq. (2.72) with the test function of the interface displacement \$\delta d^i\$.

The monolithic system with condensed element stresses is given as

$$\begin{bmatrix} F_{uu} + C_{uu} & F_{up} + C_{up} & C_{ud^i} & 0 & 0 \\ F_{pu} & F_{pp} & 0 & 0 & 0 \\ C_{d^i u} & C_{d^i p} & C_{d^i d^i} & D_{d^i \mu} & 0 \\ 0 & 0 & D_{\mu d^i} & 0 & M_{\mu d^s} \\ 0 & 0 & 0 & M_{d^s \mu} & S_{d^s d^s} \end{bmatrix}_k \begin{bmatrix} \Delta u \\ \Delta p \\ \Delta d^i \\ \Delta \mu \\ \Delta d^s \end{bmatrix} = - \begin{bmatrix} r_u + r_{\bar{u}}^{\bar{\sigma}} + \overset{c}{r}_u \\ r_p \\ r_{d^i}^{\bar{\sigma}} + \overset{c}{r}_{d^i} + r_{d^i}^{\mu} \\ r_{\mu}^{d^i} + r_{\mu}^{d^s} \\ r_{d^s}^s + r_{d^s}^{\mu} \end{bmatrix}_k \quad (6.16)$$

The matrices  $C$  and coupling forces  $\overset{c}{r}$  stem from the stress condensation process, which is described in detail in Section 4.3.2 and Section 5.3.2. Splitting

Eq. (6.16) into surface and interior DOFs reveals the surface coupling:

$$\begin{bmatrix} \mathbf{F}_{uu}^{SS} & \mathbf{F}_{up}^{SS} & \mathbf{F}_{uu}^{SC} & \mathbf{F}_{up}^{SC} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{pu}^{SS} & \mathbf{F}_{pp}^{SS} & \mathbf{F}_{pu}^{SC} & \mathbf{F}_{pp}^{SC} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{uu}^{CS} & \mathbf{F}_{up}^{CS} & \mathbf{F}_{uu}^{CC} + \mathbf{C}_{uu}^{CC} & \mathbf{F}_{up}^{CC} + \mathbf{C}_{up}^{CC} & \mathbf{C}_{ud^i}^C & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{pu}^{CS} & \mathbf{F}_{pp}^{CS} & \mathbf{F}_{pu}^{CC} & \mathbf{F}_{pp}^{CC} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{d^i u}^C & \mathbf{C}_{d^i p}^C & \mathbf{C}_{d^i d^i}^C & \mathbf{D}_{d^i \mu} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_{\mu d^i} & \mathbf{0} & \mathbf{M}_{uds} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{ds\mu} & \mathbf{S}_{dsds}^{BB} & \mathbf{S}_{dsds}^{BD} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{dsds}^{DB} & \mathbf{S}_{dsds}^{DD} \end{bmatrix}_k \begin{bmatrix} \Delta u_s \\ \Delta p_s \\ \Delta u_c \\ \Delta p_c \\ \Delta d^i \\ \Delta \mu \\ \Delta d^s_B \\ \Delta d^s_D \end{bmatrix} = \begin{bmatrix} \mathbf{r}_u^S \\ \mathbf{r}_p^S \\ \mathbf{r}_u^C + \mathbf{r}_u^{\bar{\sigma}, C} + \mathbf{r}_u^C \\ \mathbf{r}_p^C \\ \mathbf{r}_{d^i}^\mu + \mathbf{r}_{d^i}^{\bar{\sigma}} + \mathbf{r}_{d^i}^c \\ \mathbf{r}_{\mu}^{d^i} + \mathbf{r}_{\mu}^{d^s} \\ \mathbf{r}_{d^s}^{s,B} + \mathbf{r}_{d^s}^{\mu, B} \\ \mathbf{r}_{d^s}^{s,D} \end{bmatrix}_k \quad (6.17)$$

In contrast to the traction Lagrange multiplier approach, the stress based Lagrange multiplier poses no constraints on the interface discretization. Numerical experience suggests, that any polynomial order and element size can be used for the interface discretization. The interface mesh is therefore chosen to be identical to the structure surface mesh. The advantage of identical structure surface and interface meshes is that the projection  $P^\mu$  in Eq. (6.8) is not required anymore. It follows that

$$\mathbf{D}_\mu = \mathbf{M}_\mu \quad (6.18)$$

and thus the computation of the interface displacement reduces to

$$\Delta \mathbf{d}^i = \mathbf{D}_\mu^{-1} \mathbf{M}_\mu \Delta \mathbf{d}^s = \mathbf{I} \Delta \mathbf{d}^s = \Delta \mathbf{d}^s \quad (6.19)$$

with  $\mathbf{I}$  being the identity matrix. This essentially replaces the weak interface-structure coupling with a matching-nodes coupling.

For the monolithic scheme, the identical surface discretizations remove the need for an extra interface mesh. It can be removed from the global system, which leaves the following system to be solved

$$\begin{bmatrix} \mathbf{F}_{uu} + \mathbf{C}_{uu} & \mathbf{F}_{up} + \mathbf{C}_{up} & \mathbf{C}_{uds} \\ \mathbf{F}_{pu} & \mathbf{F}_{pp} & \mathbf{0} \\ \mathbf{C}_{dsu} & \mathbf{C}_{d^s p} & \mathbf{S}_{dsds} + \mathbf{C}_{dsds} \end{bmatrix}_k \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{p} \\ \Delta \mathbf{d}^s \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_u + \mathbf{r}_u^{\bar{\sigma}} + \mathbf{r}_u^c \\ \mathbf{r}_p \\ \mathbf{r}_{d^s}^s + \mathbf{r}_{d^s}^{s,\bar{\sigma}} + \mathbf{r}_{d^s}^c \end{bmatrix}_k \quad (6.20)$$

with  $\mathbf{r}_{ds}^{s,\bar{\sigma}} = \mathbf{r}_{di}^{\bar{\sigma}}$  and  $\dot{\mathbf{r}}_{ds}^c = \dot{\mathbf{r}}_{di}^c$ . In the implementation, the vectors  $\mathbf{r}_{ds}^{s,\bar{\sigma}}$  and  $\dot{\mathbf{r}}_{ds}^c$  and the corresponding tangent matrices  $\mathbf{C}_{dsu}$ ,  $\mathbf{C}_{dsp}$  and  $\mathbf{C}_{dss}$  are generated by renumbering the DOFs from the interface numbering scheme to the structure surface numbering scheme. Splitting the monolithic system again into interior and surface degrees of freedom, the surface coupling becomes visible:

$$\begin{bmatrix} \mathbf{F}_{uu}^{SS} & \mathbf{F}_{up}^{SS} & \mathbf{F}_{uu}^{SC} & \mathbf{F}_{up}^{SC} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{pu}^{SS} & \mathbf{F}_{pp}^{SS} & \mathbf{F}_{pu}^{SC} & \mathbf{F}_{pp}^{SC} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{uu}^{CS} & \mathbf{F}_{up}^{CS} & \mathbf{F}_{uu}^{CC} + \mathbf{C}_{uu}^{CC} & \mathbf{F}_{up}^{CC} + \mathbf{C}_{up}^{CC} & \mathbf{C}_{uds}^{CB} & \mathbf{0} \\ \mathbf{F}_{pu}^{CS} & \mathbf{F}_{pp}^{CS} & \mathbf{F}_{pu}^{CC} & \mathbf{F}_{pp}^{CC} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{dsu}^{BC} & \mathbf{C}_{dsp}^{BC} & \mathbf{S}_{dsdss}^{BB} + \mathbf{C}_{dsdss}^{BB} & \mathbf{S}_{dsdss}^{BD} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{dsdss}^{DB} & \mathbf{S}_{dsdss}^{DD} \end{bmatrix}_k \begin{bmatrix} \Delta u_S \\ \Delta p_S \\ \Delta u_C \\ \Delta p_C \\ \Delta d_B^s \\ \Delta d_D^s \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_u^s \\ \mathbf{r}_p^s \\ \mathbf{r}_u^c + \mathbf{r}_u^{s,\bar{\sigma},c} + \mathbf{r}_u^c \\ \mathbf{r}_p^c \\ \mathbf{r}_{ds}^{s,B} + \mathbf{r}_{ds}^{s,\bar{\sigma},B} + \mathbf{r}_{ds}^B \\ \mathbf{r}_{ds}^{s,D} \end{bmatrix}_k \quad (6.21)$$

## 6.4 Iterative staggered strong coupling scheme

For the dynamic FSI problems involving incompressible fluid flow and light-weight structures considered here, an iterative staggered scheme based on [Küttler and Wall, 2008] has been implemented. Each field is solved implicitly and an iterative procedure over the fields ensures convergence for the interface conditions at the new time step level  $n + 1$ . It has been shown in [Fürster et al., 2007] that such strong, iteratively coupling schemes are necessary for FSI involving incompressible flow and lightweight structures.

For a partitioned approach, the interface-structure coupling in the lower right part of the matrix in both, Eq. (6.14) and Eq. (6.17) is replaced by an iterative Dirichlet-Neumann scheme. In such scheme, the structural displacement at the surface represents a Dirichlet condition for the interface-fluid system, while the resulting fluid surface traction is applied as Neumann condition on the structure.

For the traction Lagrange multiplier approach, the interface mesh is recreated at every FSI iteration step  $i$  based on the latest structure surface position. Hence, the interface velocity cannot be computed continuously in time, since the interface mesh has different number of nodes at different steps  $i$ . Therefore, the interface velocity  $\mathbf{u}_{FSI}^s$  is computed on the structure surface discretization by using Eq. (5.17). Subsequently, the resulting structure surface

velocity is transferred with the already established mortar matrices as

$$\boldsymbol{u}^i = \boldsymbol{D}_\mu^{-1} \boldsymbol{M}_\mu \boldsymbol{u}_{\text{FSI}}^s \quad (6.22)$$

Note that this does not mix structure and interface time discretization. With the computed interface velocity, the fluid system is solved using Eq. (4.20). The corresponding Neumann condition along  $\Gamma^i$  is derived from the integral  $(\delta \boldsymbol{d}^s, \boldsymbol{\mu})_{\Gamma^i}$ . The fluid interface system is solved as given in Eq. (5.23) with the traction field  $\boldsymbol{\lambda}$  as part of the solution. The interface force vector can be written as

$$\boldsymbol{f}_{II}^{s,\text{FSI}} = \underbrace{\int_{\Gamma^i} P^\mu N_I^s N_J^i dx}_{\boldsymbol{M}_{ds_\mu}} \boldsymbol{\mu}_{Ji} \quad (6.23)$$

or, in matrix notation,

$$\boldsymbol{f}^{s,\text{FSI}} = \boldsymbol{M}_{ds_\mu} \boldsymbol{\mu} = -\boldsymbol{M}_{ds_\mu} \boldsymbol{\lambda} \quad (6.24)$$

Here, the identical size of  $\boldsymbol{\lambda}$  and  $\boldsymbol{\mu}$  as defined in Eq. (6.13) is exploited.

For the stress based Lagrange multiplier approach, interface mesh and structure surface mesh are identical. Furthermore, the interface mesh is unchanged during the FSI simulation. and Eq. (6.22) reduces to

$$\boldsymbol{u}^i = \boldsymbol{u}_{\text{FSI}}^s \quad (6.25)$$

Instead of Eq. (6.25), the structure displacement can directly be copied to the interface mesh as

$$\boldsymbol{d}^i = \boldsymbol{d}^s \quad (6.26)$$

as derived in Eq. (6.19). After solving the fluid system by Eq. (5.34), the interface nodal force due to the applied interface displacement as

$$\delta \boldsymbol{d}^i \boldsymbol{f}_{di}^{i,\text{FSI}} = (\delta \boldsymbol{d}^{i,h}, \bar{\boldsymbol{\sigma}}^h \cdot \boldsymbol{n}^f)_{\Gamma^i} \quad (6.27)$$

based on Eq. (5.24d). For non-matching surface grids, Eq. (6.27) gives

$$\boldsymbol{f}^{s,\text{FSI}} = \boldsymbol{M}_{ds_\mu} \boldsymbol{D}_{\mu d^i}^{-1} \boldsymbol{f}^{i,\text{FSI}} \quad (6.28)$$

Since matching interface and structure surface meshes are used, the discrete nodal force is computed by

$$\boldsymbol{f}^{s,\text{FSI}} = \boldsymbol{f}_{\bar{\boldsymbol{\sigma}}}^{i,\text{FSI}} \quad (6.29)$$

where just DOF renumbering is necessary to transfer data from the interface mesh to the structure surface mesh. This removes the need for storing and

applying the interface-structure Mortar matrices  $D_{\mu d^i}$ ,  $D_{d^i \mu}$ ,  $M_{\mu d^s}$ , and  $M_{d^s \mu}$  in the iterative coupling.

The structure Neumann problem for both, traction and stress Lagrange multiplier, is summarized as

$$\begin{bmatrix} S_{d^s d^s}^{BB} & S_{d^s d^s}^{BD} \\ S_{d^s d^s}^{DB} & S_{d^s d^s}^{DD} \end{bmatrix}_k \begin{bmatrix} \Delta d^{s,B} \\ \Delta d^{s,D} \end{bmatrix} = - \begin{bmatrix} r_{d^s}^{s,B}(d^s) + f^{s,FSI,B} \\ r_{d^s}^{s,D} \end{bmatrix}_k \quad (6.30)$$

The force vector  $f^{s,FSI,B}$  is based on either Eq. (6.24) or Eq. (6.29), respectively.

As proposed in [Mok and Wall, 2001; Wall et al., 1999], the Aitken acceleration scheme for vector sequences [Iron and Tuck, 1969] is applied to relax the computed interface displacement. The relaxed interface displacement for the next FSI step  $i + 1$  is computed by

$$d_{i+1}^s = \omega_i d_{i+1}^{s,n+1} + (1 - \omega_i) d_i^s \quad (6.31)$$

The relaxation parameter  $\omega_i$  is computed as given in [Mok and Wall, 2001; Wall et al., 1999]. The partitioned coupling procedures for both approaches are summarized in Algorithm 6.1 and Algorithm 6.2.

---

**Algorithm 6.1** Iterative FSI for traction Lagrange multipliers

---

```

for each new time step  $n + 1$  do
  for each FSI iteration  $i$  do
    establish interface mesh, compute  $D_\mu, M_\mu$  Eq. (6.8)
    Structure → Fluid: Transfer displacement  $d_{i+1}^i = D_{\mu d^i}^{-1} M_{\mu d^s} d_{i+1}^s$  Eq. (6.22)

    Fluid: Solve for  $u_{i+1}, p_{i+1}$  and  $\lambda_{i+1}$  Eq. (4.20)
    Fluid → Structure: Transfer reaction force  $f_{i+1}^{s,FSI} = -M_\mu^T \lambda_{i+1}$  Eq. (6.24)
    Structure: Solve for  $d_{i+1}^s$  Eq. (6.30)
    if converged then
      stop FSI iteration
    else
      Relax displacements and restart iteration Eq. (6.31)
    end if
  end for
end for

```

---

In the implementation, the fluid-structure coupling information is only exchanged between structure surface and interface mesh. The interface mesh acts as an explicit fluid surface for the FSI algorithms, and the fluid-interface system is algorithmically treated as one block. This allows to us the same coupling implementation as an ALE based FSI scheme, where the structure

**Algorithm 6.2** Iterative FSI for stress Lagrange multipliers

---

```

establish interface mesh as copy of the structure surface mesh
for each new time step  $n + 1$  do
    for each FSI iteration  $i$  do
        Structure → Fluid: Transfer displacement  $d_{i+1}^i = d_{i+1}^s$            Eq. (6.26)
        Fluid: Solve for  $u_{i+1}$ ,  $p_{i+1}$ ,  $f_{i+1}^{i,FSI}$                                 Eq. (5.34)
        Fluid → Structure: Transfer reaction force  $f_{i+1}^{s,FSI} = f_{i+1}^{i,FSI}$    Eq. (6.29)
        Structure: Solve for  $d_{i+1}^s$                                          Eq. (6.30)
        if converged then
            stop FSI iteration
        else
            Relax displacements and restart iteration                         Eq. (6.31)
        end if
    end for
end for

```

---

surface exchanges forces and velocities with the surface of the ALE based fluid mesh.

The combination of stress based approach and the partitioned FSI scheme allows to concentrate entirely on the fixed-grid fluid formulation and the FSI coupling. The strict use of surface coupling allows to use all existing structure models that are implemented in the simulation environment. Although this has not been exploited in the examples shown at the end of this chapter, future applications will certainly benefit from this feature.

## 6.5 Enhancement of flow solutions near the fluid structure interface

For an accurate FSI simulation, it is crucial to properly resolve flow features like boundary layers, flow separation and re-attachment that often occur near the structural surface. An inaccurate resolution may corrupt the entire simulation and the coupled simulation will not be able to predict the physical solution. However, for ‘pure’ fixed-grid methods, an adequate mesh cannot be constructed beforehand, since the position of the fluid-structure interface is usually not known *a priori*. Hence, for pure fixed-grid methods special care is necessary to create an appropriate mesh that allows for reliable simulation of complex problems.

In the following, two different techniques to improve this situation are discussed. The first approach is based on local, adaptive mesh refinement and coarsening combined with error-estimator and heuristics based refinement

indicators. The second approach combines the enriched fixed-grid approach with ALE techniques [Gerstenberger and Wall, 2008a]. It essentially adds a surface layer of deformable fluid elements with an ALE formulation to the structural surface. Such a fluid patch captures the near surface flow with an appropriate fine mesh, which is then coupled to a coarse fixed Eulerian background mesh. For demonstration, the techniques are introduced for two-dimensional problems using the traction Lagrange multiplier approach.

### 6.5.1 Adaptive fixed-grid methods

The current work concentrates on  $h$ -adaptivity. Nevertheless, extensions to  $hp$ -finite element [Babuška and Suri, 1990; Gui and Babuška, 1986] methods seem possible, since the XFEM allows to use higher order approximations consistently [Cheng and Fries, 2010; Laborde et al., 2005; Stazi et al., 2003].

An  $h$ -adaptive scheme essentially consists of two steps: finding an appropriate refinement/coarsening indicator and modifying the computational mesh accordingly.

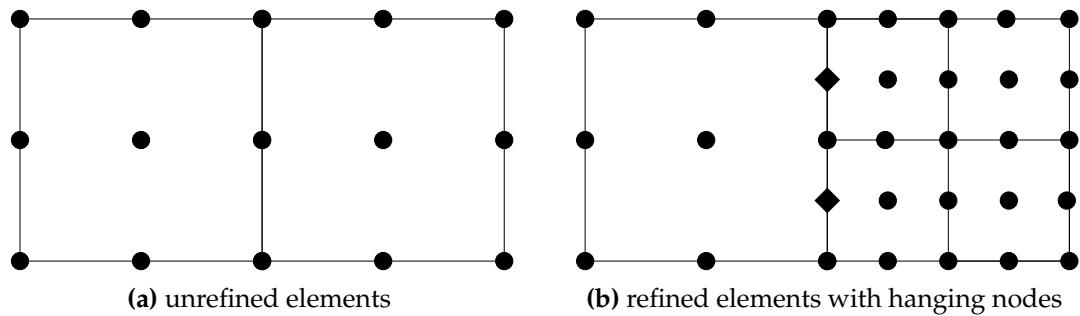
A refinement indicator can be as simple as to refine all element within a given distance from the interface and make a smooth transition of element sizes towards coarser elements. This approach is useful, if for example the size of the boundary layer can be predicted a priori. If flow features develop away from the interface, such heuristics are not sufficient and refinement indicators based on global or local error estimates are more general and preferable. See [Oñate et al., 2006; Sahni et al., 2006; Zienkiewicz and Zhu, 1992a,b] and references therein for a more complete overview on error estimates.

In the presented computations, a combination of refining by the interface distance and a gradient based error indicator is applied. The error indicator  $\varepsilon^e$  for each element  $e$  is computed from the integral of the absolute value of the shear strain rate jump along the element boundaries  $\Gamma^e$  to estimate the smoothness of the computed result

$$\varepsilon^e = \oint_{\Gamma^e} |[u_{x,y}]| dx \quad (6.32)$$

Here,  $u_{x,y}$  is the gradient of the fluid velocity field. For the results given in the example section, this indicator captures boundary layers along explicit and implicit boundaries satisfactorily.

After indicating, which area requires refinement and where coarsening can be allowed, several choices exist for a refinement strategy, see e.g. [Stein et al., 2004] for an overview on refinement techniques. For the presented fixed-grid approach, local element subdivision with hanging nodes has been adopted. Refining an element, but not its neighbors, introduces hanging nodes. To obtain a compatible mesh, one can subdivide neighboring elements



**Figure 6.5:** Refinement of quadratic elements using hanging nodes (indicated by diamonds).

such that no hanging nodes are left [Cougny and Shephard, 1999; Sahni et al., 2006] or one can constrain nodes on the finer side such that a compatible solution is recovered [see e.g. Wohlmuth, 2001]. The latter approach can also be used for higher order hp-refinement and multiple hanging nodes, see e.g. Solin2008.

For testing, a local refinement strategy based on hanging nodes has been implemented into the two-dimensional code. The refinement procedure for a 9-node element is depicted in Figure 6.5. After refinement, there are 3 nodes of a quadratic line element on the coarse side and five nodes of two quadratic lines on the refined side. The two center nodes of the small elements cannot be chosen freely, since that would introduce an incompatible velocity solution. Instead, their values need to be constrained to the coarser solution.

The relation between unknowns on the refined side  $u_J^{\text{fine}}$  and the coarse side  $u_J^{\text{coarse}}$  can be expressed as the following matrix expression

$$\begin{bmatrix} u_1^{\text{fine}} \\ u_2^{\text{fine}} \\ u_3^{\text{fine}} \\ u_4^{\text{fine}} \\ u_5^{\text{fine}} \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.375 & 0.75 & -0.125 \\ 0.0 & 1.0 & 0.0 \\ -0.125 & 0.75 & 0.375 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} u_1^{\text{coarse}} \\ u_2^{\text{coarse}} \\ u_3^{\text{coarse}} \end{bmatrix} \quad (6.33)$$

Where nodes of the coarse mesh coincide with the fine mesh, the matrix contains simply an one-to-one matching, otherwise, it is a sum of several entries. The sum of all entries in one row has to be 1. For linear elements, the connectivity matrix is even simpler

$$\begin{bmatrix} u_1^{\text{fine}} \\ u_2^{\text{fine}} \\ u_3^{\text{fine}} \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 \\ 0.5 & 0.5 \\ 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} u_1^{\text{coarse}} \\ u_2^{\text{coarse}} \end{bmatrix} \quad (6.34)$$

The matrices can be constructed from simple evaluation of line shape functions or by using a local mortar approach along the interface between a coarse

and two small elements  $\Gamma^{\text{adapt}}$ , where the matrix above corresponds to the rectangular matrix

$$0 = \int_{\Gamma^{\text{adapt}}} N_I^{\text{fine}} N_J^{\text{coarse}} dx u_J^{\text{coarse}} - \int_{\Gamma^{\text{adapt}}} N_I^{\text{fine}} N_K^{\text{fine}} dx u_K^{\text{fine}} \quad (6.35)$$

or, written in matrix notation

$$0 = \mathbf{D}^{\text{adapt}} \mathbf{u}^{\text{coarse}} - \mathbf{M}^{\text{adapt}} \mathbf{u}^{\text{fine}} \quad (6.36)$$

Hence, the values on the fine grid are calculated as

$$\mathbf{u}^{\text{fine}} = \mathbf{D}^{\text{adapt},-1} \mathbf{M}^{\text{adapt}} \mathbf{u}^{\text{coarse}} \quad (6.37)$$

For simplicity of implementation, only one coarse element is allowed to be connected to two finer elements.

In practice, hanging nodes do not carry unknowns and their values can be established by interpolation from the coarse side. The integration of elements containing hanging nodes remains unchanged compared to elements without hanging nodes. Only the interpolation and assembly process needs to be adapted. When an element containing a hanging node as one of its nine nodes is processed, it first interpolates the nodal values from the coarser neighbor element. It can then compute residual and tangent stiffness as usual. During the subsequent assembly process, the entries of residual and stiffness matrix belonging to the hanging node are added to the nodes on the coarse element entries using Eqs. (6.33) and (6.34).

In summary,  $h$ -adaptivity based on local element sub-division provides a general applicable way to improve the accuracy of fixed-grid methods. The use of automatic error-estimator based refinement complements the idea of fixed-grid methods, where not much predictions on the structural movement can be made a priori.

For transient problems involving large motions of the structural surface, mesh updates are frequently required to follow the implicit interface. If a too small region around the structure is refined, the mesh update and error estimation has to be performed frequently to follow the motion of the interface. On the other hand, if too large regions are refined to reduce the frequency of adaptive steps, the number of unnecessary elements can be prohibitive high. In such cases it would be advantageous to have a small layer of fluid elements along the structural surface such that a close to optimal boundary layer mesh is kept at all times. Such an approach is presented in the next section.

### 6.5.2 A hybrid fixed-grid / ALE approach

The second approach to improve fixed-grid methods involves the addition of a deformable fluid domain around the structural surface. Surface aligned

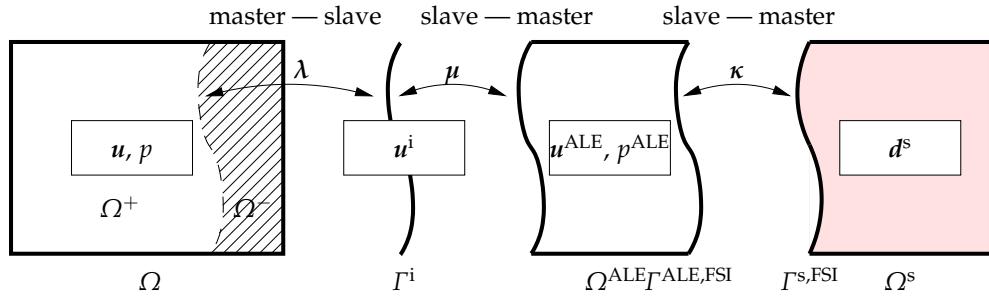
fluid elements around the structure can efficiently resolve boundary layers, because boundary layers have strong gradients usually only normal to the FSI surface. State-of-the-art boundary layer elements may have large aspect ratios ( $1 \times 100$  and more) and there seems to be no straightforward way to get such an optimal boundary layer mesh by using element subdivision as described in the previous section. For three-dimensional computations, this advantage would be even more pronounced. The challenge of this approach lies in the coupling of the flow on the embedded patch and on the background grid.

Hybrid methods have been used before to enable rigid movement of structures in fluids based on the Chimera technique [Houzeaux and Codina, 2003; Meakin and Suhs, 1989; Steger et al., 1983; Wall et al., 2006; Wang and Parthasarathy, 2000]. Recently, a Chimera like method was developed in [Gamnitzer and Wall, 2006; Wall et al., 2008], which also allows thick and thin deforming structures. In these approaches, the physical field on the fixed grid and ALE grid overlap each other and are solved alternating using iterative schemes. Related, non-overlapping approaches are, for instance, the sliding mesh technique proposed in [Behr and Tezduyar, 1999] and the weak coupling scheme presented in [Bazilevs and Hughes, 2008]. The non-overlapping approaches have the additional feature that the coupled fluid-fluid system can be solved monolithically, which eventually may allow monolithic FSI coupling schemes for the entire fixed grid fluid - moving grid fluid - structure system.

In this work, an XFEM based moving grid-fixed grid coupling is established by using the already developed XFEM and Lagrange multiplier techniques from the previous chapters. Here, the submerged patch  $\Omega^{\text{subm}}$  is the union of structural domain and the ALE patch domain  $\Omega^{\text{subm}} = \Omega^{\text{s}} \cup \Omega^{\text{p}}$ . On the fixed grid, no flow is computed in  $\Omega^{\text{subm}}$ . Since the physical fields on the different meshes do not overlap, the coupled fluid fields can be solved monolithically. The primary unknowns on the patch and the fixed mesh are velocity and pressure.

In the following, a partitioned FSI algorithm between structure on one side and the combined moving fluid - fixed fluid system on the other side is assumed. Figure 6.6 shows, how the intermediate ALE mesh is coupled to the structure surface using the Lagrange multiplier field  $\kappa$ . Hereby, the structure displacement is given as Dirichlet boundary condition to the moving fluid in  $\Omega^{\text{p}}$ . The fixed fluid and moving fluid are treated together in a monolithic way and the resulting FSI interface forces along the fluid-structure interface are used as Neumann conditions for the structure as usual. Hence, the monolithic fluid-fluid coupling is hidden inside the fluid formulation and does not affect the overall partitioned FSI scheme.

The fluid-interface coupling in Eq. (4.11c) can be extended such that the



**Figure 6.6:** Hybrid approach: a fixed fluid grid and a deforming ALE fluid grid are coupled using the XFEM/Lagrange multiplier approach. The moving ALE grid is coupled to the structure field.

complete weak form is given as

$$0 = F_M(\delta \mathbf{u}, \mathbf{u}, p) - (\delta \mathbf{u}, \lambda)_{\Gamma^i} \quad (6.38a)$$

$$0 = F_C(\delta p, \mathbf{u}, p) \quad (6.38b)$$

$$0 = -(\delta \lambda, \mathbf{u} - \mathbf{u}^i)_{\Gamma^i} \quad (6.38c)$$

$$0 = (\delta \mathbf{u}^i, \lambda + \mu)_{\Gamma^i} \quad (6.38d)$$

$$0 = -(\delta \mu, \mathbf{u}^p - \mathbf{u}^i)_{\Gamma^i} \quad (6.38e)$$

$$0 = F_M^p(\delta \mathbf{u}^p, \mathbf{u}^p, p^p) - (\delta \mathbf{u}^p, \mu)_{\Gamma^i} \quad (6.38f)$$

$$0 = F_C^p(\delta p^p, \mathbf{u}^p, p^p) \quad (6.38g)$$

The stabilized fluid weak forms  $F_M^p(\delta \mathbf{u}^p, \mathbf{u}^p, p^p)$  and  $F_C^p(\delta p^p, \mathbf{u}^p, p^p)$  are based on the ALE formulation as discussed in [Wall et al., 2008].

For a partitioned FSI scheme, the ALE mesh deformation and the interface position between fixed and moving fluid domain are not part of the fluid solution. Instead, the ALE fluid-structure interface is moved to the new time steps position and the favorite mesh smoothing scheme can be applied to smooth the ALE mesh. The corresponding mesh velocity is then given to the ALE equations as a given mesh velocity field.

The time discretization for ALE fluid patch and Eulerian background fluid are chosen to be the same. The linearization of both fluid fields is performed

as in Section 4.3.1. The resulting matrix equation is then given

$$\begin{bmatrix} F_{uu} & F_{up} & M_{u\lambda} & 0 & 0 & 0 & 0 \\ F_{pu} & F_{pp} & 0 & 0 & 0 & 0 & 0 \\ M_{d^i\lambda} & 0 & 0 & D_{\lambda d^i} & 0 & 0 & 0 \\ 0 & 0 & D_{d^i\lambda} & 0 & D_{d^i\mu} & 0 & 0 \\ 0 & 0 & 0 & D_{\mu d^i} & 0 & M_{\mu u^p} & 0 \\ 0 & 0 & 0 & 0 & M_{u^p \mu} & F_{u^p u^p} & F_{u^p p^p} \\ 0 & 0 & 0 & 0 & 0 & F_{p^p u^p} & F_{p^p p^p} \end{bmatrix}_k \begin{bmatrix} \Delta u \\ \Delta p \\ \Delta \lambda \\ \Delta u^i \\ \Delta \mu \\ \Delta u^p \\ \Delta p^p \end{bmatrix} = - \begin{bmatrix} r_u + r_u^\lambda \\ r_p \\ r_\lambda^u + r_\lambda^{u^i} \\ r_\lambda^u + r_\mu^u \\ r_{u^i}^u + r_{u^i}^{u^p} \\ r_\mu^u + r_\mu^{u^p} \\ r_{u^p} + r_{u^p}^\mu \\ r_{p^p} \end{bmatrix}_k \quad (6.39)$$

This discrete system is similar to Eq. (6.14). The coupling occurs only in the momentum equations. For this approach, an example is given Section 6.6.2.

The stress Lagrange multiplier approach from Section 4.3.2 would generate a simpler matrix structure without saddle points and no additional interface mesh. The construction of the linear system is analogous to the process given in Section 4.3.2 and Section 6.3 and the final, monolithic system before condensation is

$$\begin{bmatrix} F_{uu} & F_{up} & G & 0 & 0 \\ F_{pu} & F_{pp} & 0 & 0 & 0 \\ K_{\bar{\sigma}u} + G_{\bar{\sigma}u} & K_{\bar{\sigma}p} & K_{\bar{\sigma}\bar{\sigma}} & G_{\bar{\sigma}u^p} & 0 \\ 0 & 0 & G_{u^p \bar{\sigma}} & F_{u^p u^p} & F_{u^p p^p} \\ 0 & 0 & 0 & F_{p^p u^p} & F_{p^p p^p} \end{bmatrix}_k \begin{bmatrix} \Delta u \\ \Delta p \\ \Delta \bar{\sigma} \\ \Delta u^p \\ \Delta p^p \end{bmatrix} = - \begin{bmatrix} r_u + r_u^{\bar{\sigma}} \\ r_p \\ r_{\bar{\sigma}} \\ r_{u^p} + r_{u^p}^{\bar{\sigma}} \\ r_{p^p} \end{bmatrix}_k \quad (6.40)$$

After condensation, the monolithic system reduces to

$$\begin{bmatrix} F_{uu} + C_{uu} & F_{up} + C_{up} & C_{uu^p} & 0 \\ F_{pu} & F_{pp} & 0 & 0 \\ C_{u^p u} & C_{u^p p} & F_{u^p u^p} + C_{u^p u^p} & F_{u^p p^p} \\ 0 & 0 & F_{p^p u^p} & F_{p^p p^p} \end{bmatrix}_k \begin{bmatrix} \Delta u \\ \Delta p \\ \Delta u^p \\ \Delta p^p \end{bmatrix} = - \begin{bmatrix} r_u + r_u^{\bar{\sigma}} + \dot{r}_u \\ r_p \\ r_{u^p} + r_{u^p}^{\bar{\sigma}} + \dot{r}_{u^p} \\ r_{p^p} \end{bmatrix}_k \quad (6.41)$$

Preliminary results suggest the correctness of above stress based coupling approach. A more in-depth analysis and numerical testing is still ongoing and is therefore omitted from this work.

If the structural deformation is mainly rotational or translational with only reasonable deformation of the structure, an attached fluid mesh around the structure provides a reliable and potentially more efficient alternative to the more general adaptive procedure presented in Section 6.5.1. For many applications like spinning rotors or large translational motions of objects, the resulting deformation of the boundary layer mesh does not pose a limitation.

## 6.6 Numerical examples

The following examples demonstrate both accuracy and versatility of the fixed-grid approach.

### 6.6.1 Rotating cylinder

This example shows incompressible flows interacting with a rather soft, compressible structure (Poisson ratio  $\nu^s = 0.0$ ) exhibiting large deformation. The setup is a unit square with Dirichlet boundary condition at the top and bottom of the square and zero Neumann conditions along the right and left side generating a simple shear flow. A two-dimensional cylinder representation is located in the center of the domain with the center node fixed. The viscous forces enforce a rotation around its center. Additionally, the structure deforms due to the fluid forces acting on the structural surface. This example also represents a preliminary study to simulation of experiments as given in [Pozrikidis, 2006; Watanabe et al., 2006, 2007; Zhou and Pozrikidis, 1995], where mechanical properties are deducted from the shearing of red blood cells.

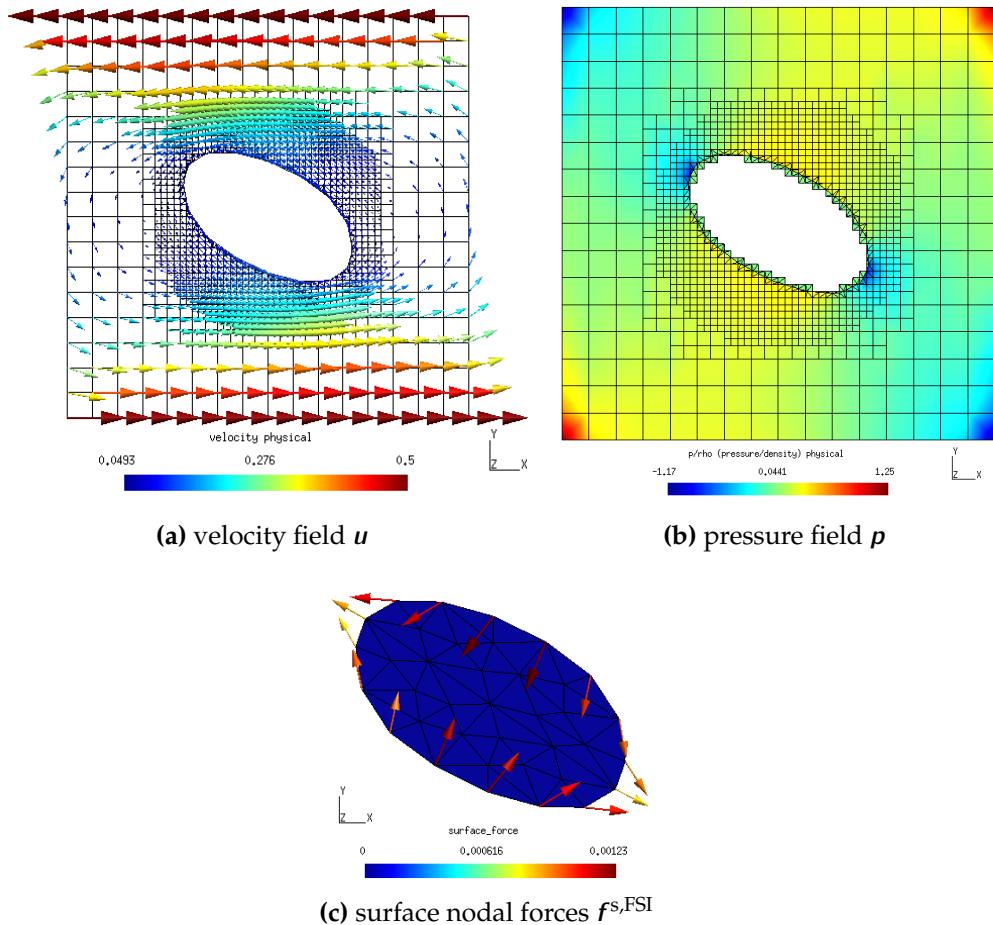
The flow solution and the corresponding structural nodal forces are depicted in Figure 6.7. From the fluid point of view, this problem becomes steady once the structure has reached the final shape as shown in the pictures. However, the structure keeps rotating, making this a transient problem after all. The rotation can go on indefinitely as no accumulating mesh deformation as in ALE-based fluid formulations is present.

### 6.6.2 Resolution of boundary layers

The following example highlight the difference between resolving boundary layers by Cartesian element subdivision and surface adapted boundary layer meshes. As a test case, the stationary flow around a cylinder (benchmark case 2D-1 in [Schäfer and Turek, 1996]) is used again. Although the focus is on fluid solution, the structure is simulated as well, as the approaches have been developed for FSI and have been implemented in a respective environment. Nevertheless, the structure is fixed in both examples such that no structure deformation occurs.

#### *h*-adaptivity

The initial fluid mesh was built from 96 ( $4 \times 24$ ) bi-quadratic elements. The mesh was refined five times; after each refinement the solution of the fluid system was repeated. The final mesh consisted of 2740 fluid elements For the

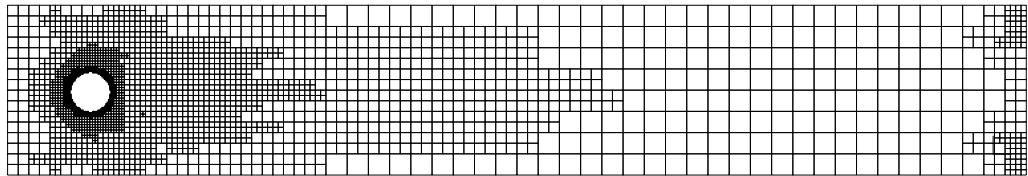


**Figure 6.7:** Flow field around a flexible, rotating cylinder.

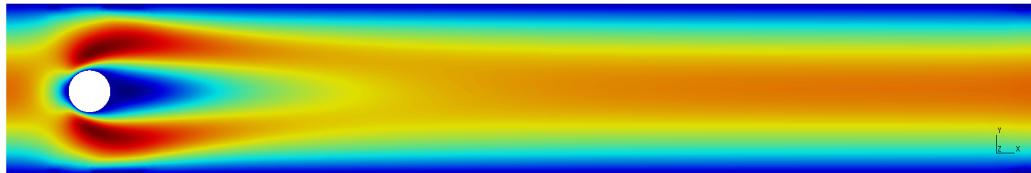
refinement indicator, the strategy from Section 6.5.1 was used. In particular, along the implicit interface, the refinement level was increased by one in each adaptive step. In addition, a number of elements were refined, where high gradients are observed.

The finest mesh and the final velocity solution is depicted in Figure 6.8. The velocity component  $u_x$  of the fluid solution is shown over the whole fluid domain in Figure 6.8b and near the interface in Figure 6.9. Intersected elements are shown using the triangular integration cells. Figure 6.8a shows the finest mesh that was used to study this problem. It consists of 284 linear Lagrange multiplier elements and 2740 quadratic 9-node fluid elements. With this, good agreement with the averaged results given in [Schäfer and Turek, 1996] is obtained.

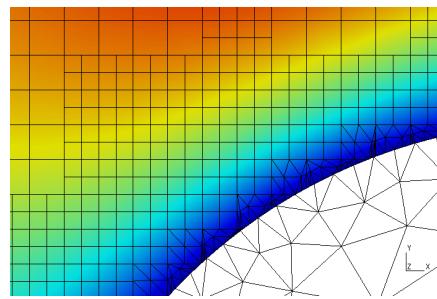
The benchmark also demonstrates, that is possible to compute accurate FSI surfaces using the traction based Lagrange multiplier formulation. To test the correct transfer of interface forces onto the structure surface, a fixed



(a) Maximal refinement used in computation

(b)  $x$ -component of the velocity field

**Figure 6.8:** CFD benchmark: Finest mesh used to calculate lift and drag values on the cylinder. The structure mesh is omitted.

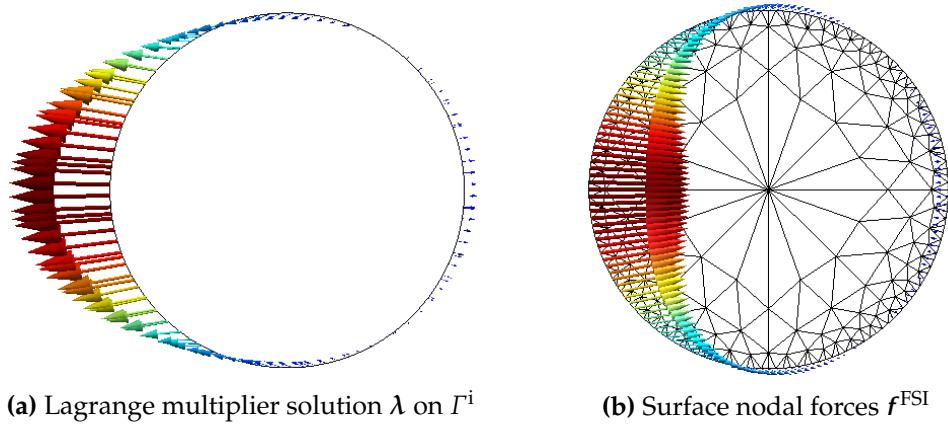


**Figure 6.9:**  $x$ -component of the velocity field near the interface. The triangular structural mesh is displayed in the lower right corner. In the fluid domain, the triangular integration cells are shown.

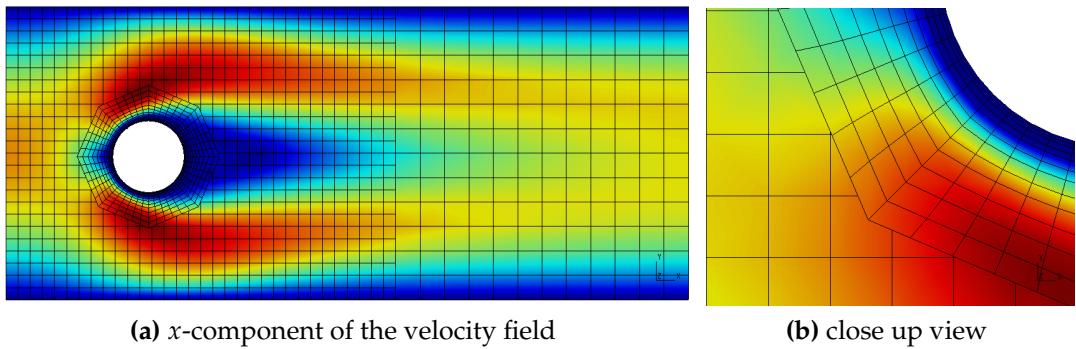
cylinder is simulated using the fully coupled, stationary FSI algorithm from Section 6.4 in conjunction with a sufficiently stiff structure. The structure is fixed at three points away from the FSI surface to prevent any motion. Figure 6.10 visually compares the solution of the Lagrange multiplier field  $\lambda$  with the projected nodal forces  $f^{\text{FSI}}$  on the structural surface nodes. Instead of using directly the fluid forces on the interface mesh, the transferred forces on the structure are used to compute lift and drag values. The calculated values are  $c_{\text{lift}} = 0.0054$  for the lift and  $c_{\text{drag}} = 5.5624$  for the drag, which are in agreement with results given in [Schäfer and Turek, 1996].

### Boundary layer meshes

The simulation with a boundary layer mesh uses the same setup as in the first example. An initial refinement for all elements in the entrance part of the channel is applied. Instead of further refining the mesh near the cylinder



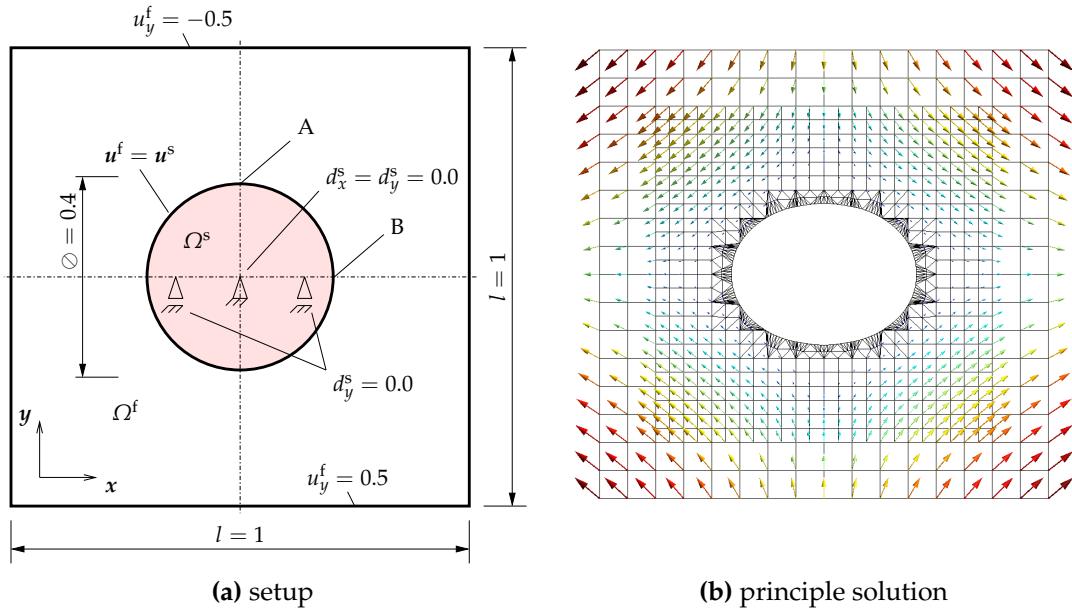
**Figure 6.10:** Lagrange multiplier field on interface discretization and structural surface nodal forces on structural discretization.



**Figure 6.11:** 2D CFD benchmark using the fluid-fluid coupling approach

surface as before, a boundary layer mesh is applied and the proposed fluid-fluid coupling ensures continuity between patch and background mesh. The final solution can be seen in Figure 6.11. The velocity component  $u_x$  of the fluid solution is shown over the whole fluid domain in Figure 6.11a and near the interface in Figure 6.11b.

The surface fitted mesh much closer resembles the boundary layer near the cylinder surface and the boundary layer can be resolved more efficiently than with Cartesian subdivision. For high Reynolds number flows, high aspect ratio elements can be used to resolve the high gradient in surface normal direction, such that overall less elements and unknowns are required. Hence, the boundary layer mesh allows to introduce engineering insight, that can reduce the numerical costs of the simulation. As shown in Figure 6.11a, the hybrid approach can be combined with the aforementioned  $h$ -adaptivity to benefit from both automatic adaptivity and a priori knowledge on boundary layers.



**Figure 6.12:** Setup and principle solution of compressible cylinder example.

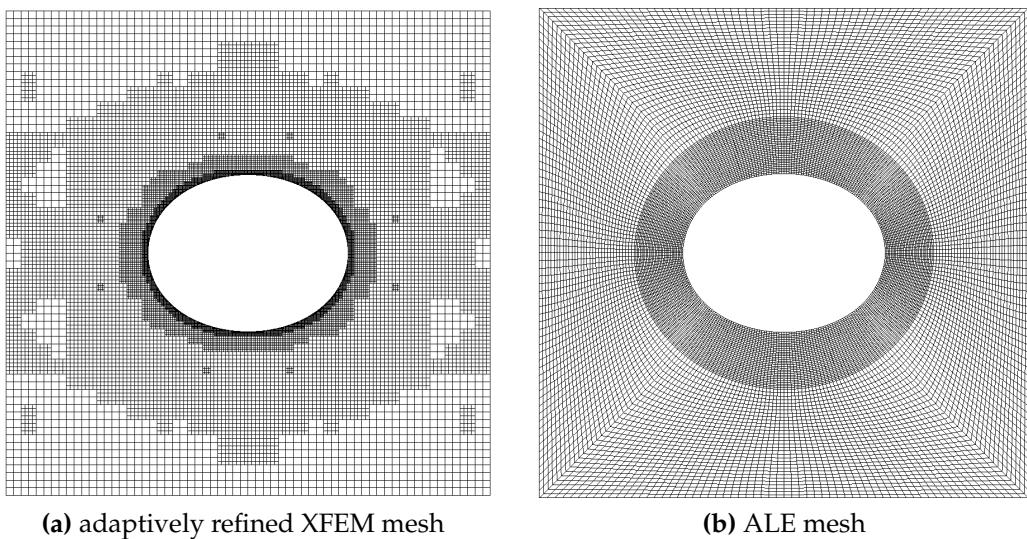
### 6.6.3 FSI including a compressible structure

The next example validates the XFSI approach by quantitatively comparing its results with computations performed by a state-of-the-art partitioned, iterative ALE-based FSI solver as described in [Küttler and Wall, 2008].

The principle setup of the example is depicted in Figure 6.12a. The structure is initially circular and is then deformed by the flow field. The material parameters for a St.-Venant Kirchhoff material law are the Poisson ratio as  $\nu = 0.0$  and the elastic modulus as  $E = 50$ . The fluids kinematic viscosity and density are 1, respectively. For the given symmetric setup, a symmetric flow field and a symmetric structure deformation with respect to the  $x$ - and the  $y$ -axis is expected. The structure was fixed such that the rigid body modes were constrained without constraining the expected symmetric deformation.

The fluid field in both solutions was discretized with equal order, bi-quadratic 9-node elements. In the ALE approach, matching grids were used between fluid and solid surface and the structure was discretized with 2698 quadratic, triangular 6-node elements. In the XFEM computation, about 1294 linear triangular 3-node elements were used.

For the ALE approach, the FSI computation was performed on increasingly fine meshes until convergence to the final structure deformation was reached. Hereby, the meshes were refined in an external pre-processor. For the XFSI approach, a steady state computation was performed, where after

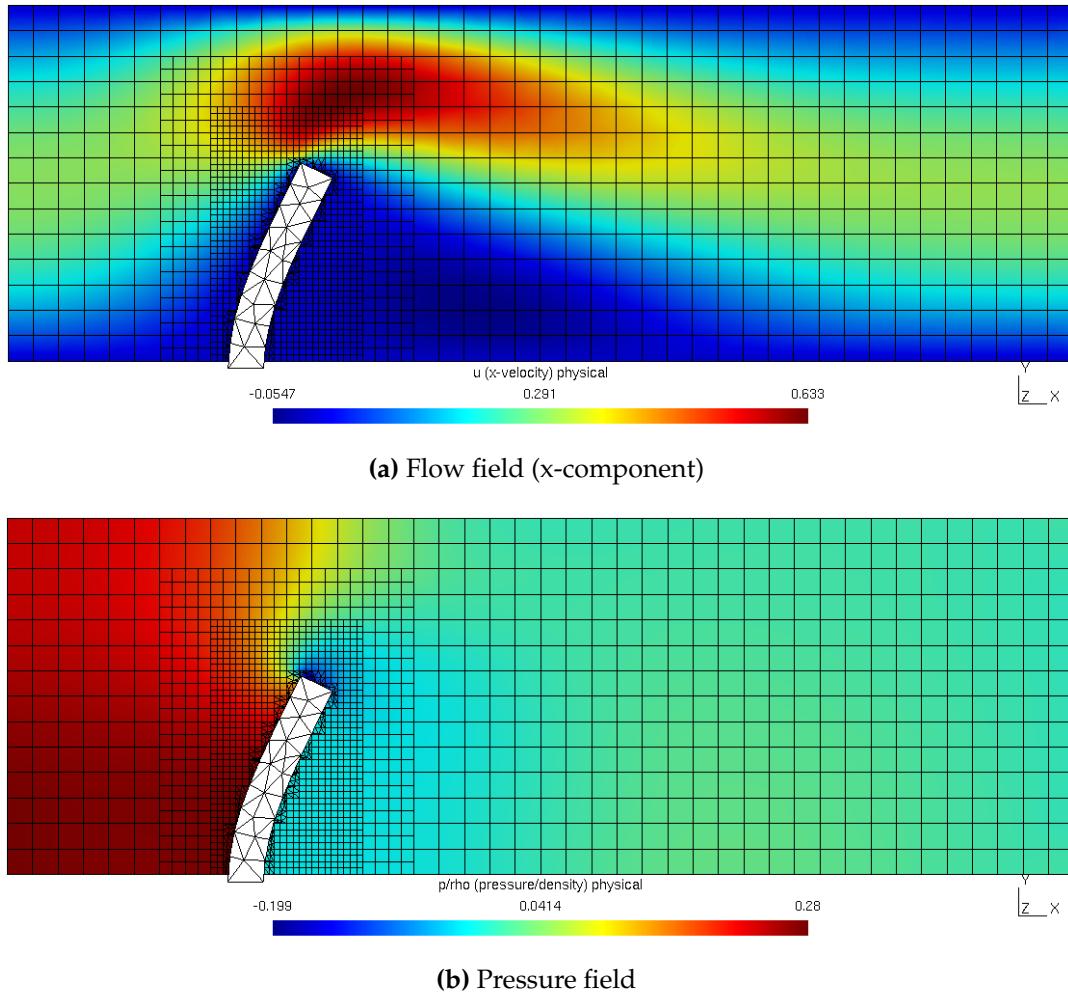


**Figure 6.13:** Final meshes used for comparison.

every second FSI iteration a refinement by subdivision of 20 % of the active fluid elements was performed. The solution was transferred to the newly created nodes and used as the starting field for the next two FSI iterations. Eventually, six refinement steps were performed until no changes in the structural deformation between subsequent refinement levels were observable. The initial fluid mesh consisted of 16 ( $4 \times 4$ ) fluid elements. The final mesh consisted of 17860 active elements counting only elements in the fluid domain or intersected elements. The final number of unknowns — fluid velocity ( $\mathbf{u}$ ), pressure ( $p$ ) and Lagrange multipliers ( $\lambda$ ) combined — is about 23000.

The final meshes for the XFEM and ALE comparison are shown in Figure 6.13. The maximal displacement was observed at the top and the bottom (Point A) of the structure at  $x = 0.5$ . Point A moved by  $d_y^A = -0.0379$ ; and Point B had a displacement of  $d_x^B = 0.00663$ . ALE and XFEM computations differed by less than 0.01 % in the maximal structure displacement. The area occupied by the compressible structure changed by approximately 16.3 % when compared with the undeformed area.

This example highlights the advantage of using surface coupling techniques and a sharp separation of the physical flow from the fictitious domain. The structure deformation may be modeled by any compressible or incompressible material model without interfering with the fluid incompressibility constraint or viscosity. In contrast, volume coupling the DLM/FD [Yu, 2005] or the IFEM [Zhang et al., 2004] have no ability to turn off the fictitious flow field and the schemes are therefore constrained to incompressible structures. The use of surface coupling makes the proposed fixed-grid method as flexible as ALE-based methods with respect to structure material choices.

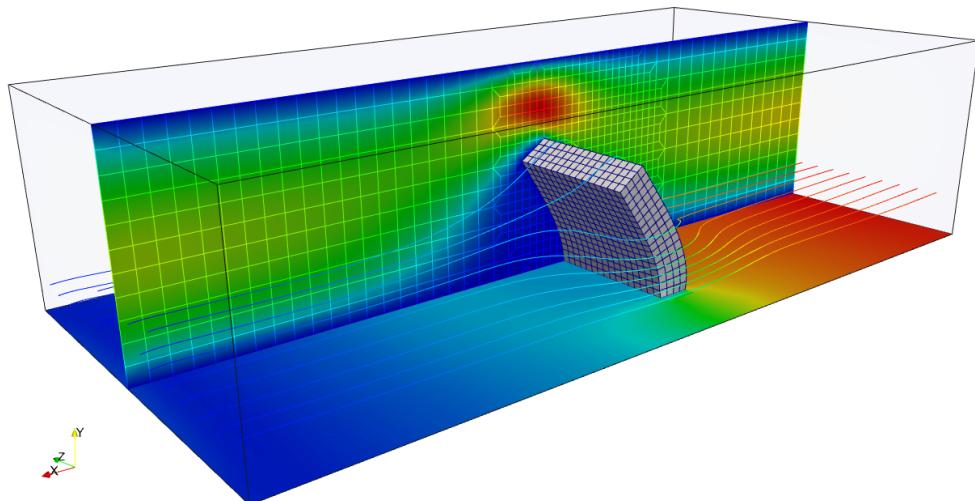


**Figure 6.14:** Stationary flow field through a channel with a flexible structure.

#### 6.6.4 Channel flow over a bending beam

In this example, a flexible structure (Poisson ratio  $\nu^s = 0.0$ , Young's modulus  $E^s = 0.5$ ) is deformed due to a flow through a channel. It is computed until steady-state using fully coupled FSI equations and XFEM. Top and bottom channel walls have zero (no-slip) velocity prescribed, the inflow from the left is prescribed by a parabolic velocity condition. The outflow boundary (right) is of Neumann (zero traction) type. The inflow and the wall boundary conditions are standard Dirichlet boundary conditions.

The Reynolds number based on the mean inflow velocity and the channel height is 16. The stationary equilibrium solution is shown in Figure 6.14. It is possible to overlap standard Dirichlet boundary conditions and Lagrange multiplier conditions along the interface as long as the enforced velocity is the same. This can be seen at the bottom side of the channel, where the structural surface connects to the channel bottom.



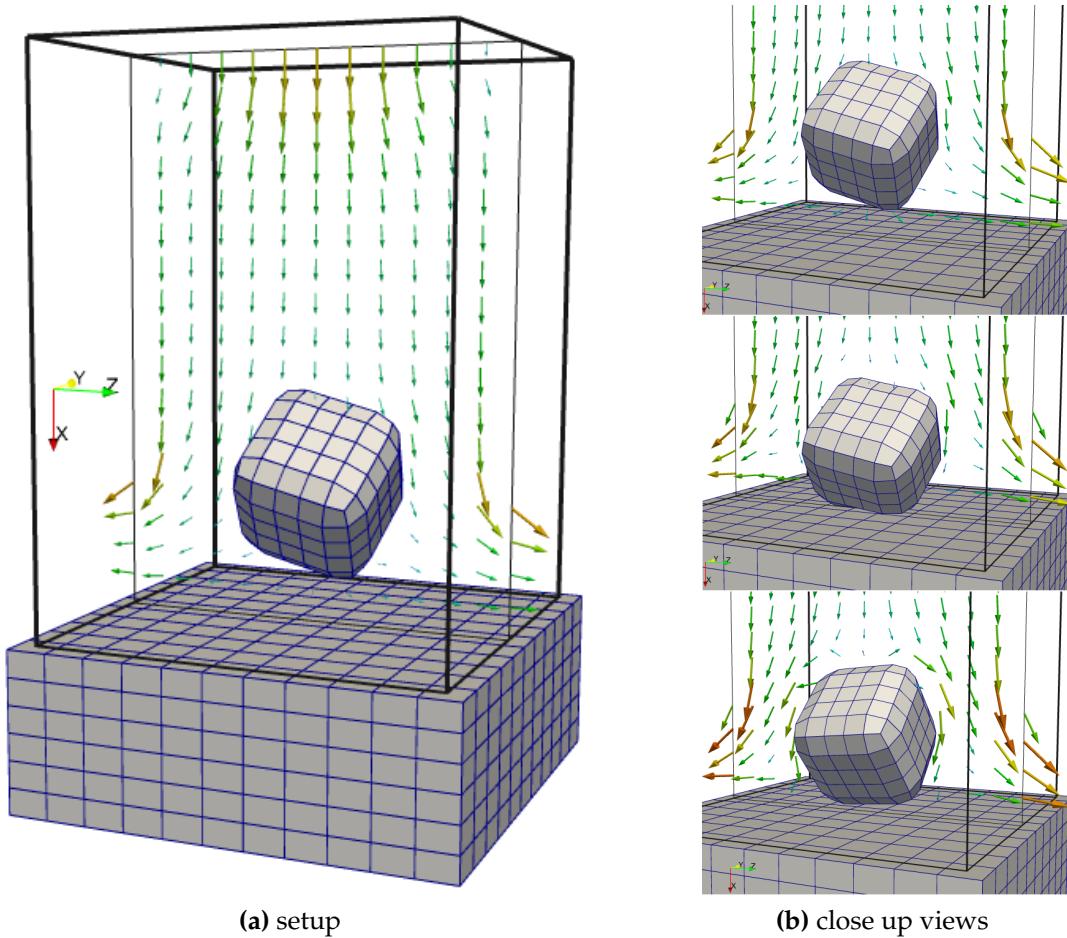
**Figure 6.15:** Stationary channel flow interacting with a flexible structure.

### 6.6.5 Channel flow over a deforming thick plate

In a similar but three-dimensional setup, a flexible structure (Poisson ratio  $\nu^s = 0.48$ , Young's modulus  $E^s = 90 \text{ N/m}^2$ ) interacts with flow in a channel. The channel has the dimensions  $0.5 \times 1.0 \times 3.0\text{m}$ . The FSI simulation is performed until steady-state is reached. Top and bottom channel walls have zero (no-slip) velocity prescribed, the inflow from the left is prescribed by a parabolic velocity condition. The Reynolds number based on the mean inflow velocity and the channel height is 16. The stationary equilibrium solution is shown in Figure 6.15.

### 6.6.6 Contact of submerged structures

The last example deals with contact of submerged structures and is an outlook towards future applications of the fixed grid method rather than a fully developed FSI contact formulation. Contact of submerged structures can be observed in many FSI problems. For instance, in bio-physical applications, closing valves, collapsing veins, interaction of blood cells are a few examples, where numerical tools could provide valuable insight. With ALE-methods, the deforming mesh essentially does not allow two objects to come into full contact (with no fluid between the structures); a minimal distance is required to keep a minimal quality of the fluid mesh, see e.g. [Sathe and Tezduyar, 2008; Tezduyar and Sathe, 2007]. Fixed-grid methods, on the other hand, have no deforming mesh between such objects and potentially open up an entire new field that can be treated by numerical tools. Recent examples for fixed-grid FSI simulations including contact of submerged structures can be found in [Astorino et al., 2009; Diniz dos Santos et al., 2008; van Loon et al.,



**Figure 6.16:** Setup of contact example and three snapshots of the first contact between a soft body and a relatively stiff block.

2006].

In [Mayer et al., 2010], the presented XFSI approach is combined with a contact formulation based on dual-Lagrange multipliers and a primal-dual active set strategy for contact constraint enforcement as described in [Popp et al., 2009]. From an algorithmic point of view, the key element of the contact formulation is the complete condensation of the contact Lagrange multipliers within the structural formulation. Consequently, the structure block can be extended with a contact formulation without interfering with the FSI formulation.

An preliminary example of such FSI contact is shown in Figure 6.16. A body force pulls a little, soft block towards a large stiff block, that has a fixed bottom surface. At the same time, a growing inflow at the top of the fluid domain pushes the little block additionally in  $x$ -direction towards the wall. Two exits at the lower end of the channel allow the flow to exit the domain.

The three small pictures show: how the fluid is pushed out of the closing gap between the little and the large block, the moment of largest deformation of the little block at first contact, and the bouncing back of the small block. The same simulation without fluid flow (not shown here) allows the block to bounce back many times to approximately the same height from where it started. Including the FSI simulation, the movement is damped by the fluid viscosity, such that it comes to rest on the lower block quickly. Since no frictional contact between both objects is employed, eventually, the little block is pushed sideways out of the fluid domain. Note that no special wet or dry contact physics is employed and that this simulation is not properly resolved to get quantitatively accurate results. Nevertheless, the qualitative behavior demonstrates the increased flexibility that comes with fixed grid methods for FSI.

# CHAPTER 7

## Summary and conclusion

### 7.1 Summary

The focus of the presented work was the development and implementation of a finite element based fixed grid method for the interaction of incompressible viscous flow with structures undergoing large deformations. The primary motivation was to circumvent the main bottleneck of ALE-based methods, namely the fluid mesh distortion and the associated need for remeshing. Since FSI is a complex physical phenomenon, hence, complex (but not complicated) algorithms can be expected. The theme for this work was to derive a general method without trading algorithmic simplicity with theoretical soundness.

The first step was to add a separate interface discretization between fluid and structure and from there on describe the coupling of each field towards this interface. For the fluid description this interface separates the computational fluid domain into a physical and a fictitious part. In the fictitious domain, the structure is modeled using a Lagrangian formulation and a separate mesh. Hence, no fluid computations are necessary and therefore should be avoided. The XFEM has been used to model the jump between physical flow and the void by adding strong discontinuities into the FE approximation of velocity and pressure. The result is a sharp interface representation independent of any fluid element boundary. In addition, no degrees of freedom are required in the fictitious fluid domain.

A new method has been developed to couple the physical flow with the interface discretization. In contrast to classical traction based Lagrange multiplier, an element stress field is introduced in intersected elements that are used to enforce the interface conditions. The new method (i) requires no separate interface mesh for the Lagrange multiplier; (ii) allows for a complete La-

grange multiplier condensation on the element level; (iii) is applicable to any linear or higher order element shape; (iv) leaves the stabilized, equal-order velocity-pressure formulation for the Navier-Stokes equations unchanged; (v) produces saddle point free discrete systems; and (vi) is stable without user-defined and potentially problem specific stabilization parameters. The combination of these facts provides a sound variational foundation for a parallel, three-dimensional implementation and for parallel solution techniques including iterative solvers. Since such duality between velocity and stress variables can be found in many physical problems, this Lagrange multiplier approach may be useful for other physical problems as well. An application to heat conduction and linear elasticity was shown.

For moving interfaces, initial considerations showed that a pure Eulerian approach is formally not appropriate for moving interfaces. The separate time and space discretization leads to apparent choices on how the fluid field should be enriched: For classical finite difference treatment of time derivatives, one can conclude that either several discontinuities exist at the same time (spatial discretization first) or one enrichment exists at the new time step (time discretization first). The latter approach requires to guess old time step values near the interface, but has been shown to converge optimally for decreasing time step sizes. In contrast, the former approach requires no guess, but leads to wrong FSI interface forces and does not converge for decreasing time step sizes. It can be concluded, that time discretization has to be performed first. Based on this finding, the remaining task is to study in more detail the required extrapolation process for complex three-dimensional fluid-structure surfaces.

The clear separation of fluid and structure concerns eventually allowed a straightforward fluid-structure coupling based on established monolithic and partitioned FSI algorithms. The structure implementation is based on the commonly used Lagrangian formulation and any compressible or incompressible material can be used without affecting the fluid implementation. No mesh size dependency exists between fluid and structure mesh. Both fields can be arbitrarily distributed on parallel processors. For partitioned FSI approaches, only the interface discretization is kept redundant on all processors, which allows subsequently an almost completely parallel evaluation of element matrices. The Lagrangian structure surface identifies the fluid-structure interface position at all times. Processor-local search trees provide means to efficiently treat the three-dimensional interface localization.

Beyond the use in FSI algorithms, the proposed XFEM/Lagrange multiplier approach can be used for a general non-overlapping domain decomposition approach. This has been demonstrated for flow problems using the classical traction Lagrange multiplier in two dimensions. Initial results show that the new stress based Lagrange multiplier improves this technique and

allows a monolithic, stabilization- and saddle-point-free domain decomposition technique. For the flow problems, boundary layer meshes with potentially high element aspect ratios can be added around structures, which provides an efficient way to improve the flow resolution near the fluid structure interface. Alternatively, or in combination with boundary meshes, spatial adaptivity based on hanging nodes allows for a fluid mesh refinement without deterioration of the fluid mesh quality. Here, both heuristic and/or error based refinement indicators have been used.

## 7.2 Outlook

Several open tasks remain to make the approach both more robust and more efficient as required for real world applications.

The results indicate that the use of integration cell techniques and full integration provides an accurate, stable and optimally convergent method. However, the generation of integration cells is a geometrically complex problem. A non-robust implementation jeopardizes large scale simulations, since any geometrically complicated intersection case will occur with increasing likelihood, if the number of elements increases and the time-step size decreases. Here, the future effort should be directed to harden the cell generation algorithms or to find alternative ways of integrating the weak form.

The relative high cost of numerical integration in intersected fluid elements becomes an issue, if the fluid-structure interface intersects fluid elements on only some processors in a parallel computation. Here, automatic parallel rebalancing strategies are required that respect the geometric location of intersected elements. This avoids that many processors wait for few processors, where the few processors manage intersections and expensive numerical integration. Likewise, processors that contain only nodes without any degrees of freedom should to be avoided. Potentially, different parallel layouts are required for element integration and solution of the linear system.

Another obstacle that hampers large scale computations is that iterative solution techniques suffer from the large condition number of the system matrix. The moving interface inevitably generates tiny element fractions with physical fluid, which lead to arbitrarily small entries in the linearized system. The heuristics to switch off the enrichment, if too small portions of an enriched shape functions are non-zero, needs to be explored further. Preconditioning techniques for the linear system, that are tailored to specific enrichments may further improve the performance of iterative solution techniques.

The thoughts on the time integration show that the Eulerian formulation can successfully be applied to moving, implicit interfaces and optimal temporal convergence can be achieved, if a proper estimate for old time step values

is applied. Nevertheless, the extrapolation process requires a more in-depth analysis with respect to conservation properties than the one given in this work.

With such improvements on robustness and efficiency, the proposed XFSI approach has all necessary features to be a useful and competitive tool for solving complex FSI problems.

# APPENDIX A

## Stress Lagrange multiplier for linear-elastic continua

In the following, the steady, linear momentum equation for a linear elastic material (small displacements and small strains) shall serve to highlight again the core features of the proposed method to weakly enforce Dirichlet conditions. The main differences compared to incompressible NS-equations are the lack of an extra pressure unknown and that for this linear problem no incremental formulation and Newton-Raphson iteration is required. The following discussion can be found with slightly different notation in [Gerstenberger and Wall, 2010].

### A.1 Problem definition

The steady state conservation of momentum in  $\Omega^s$  is

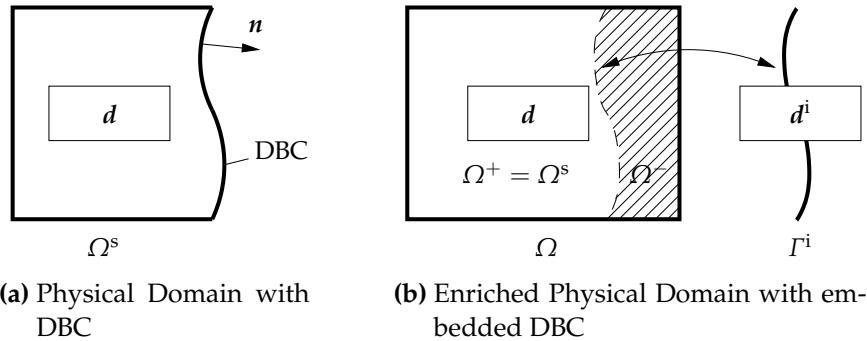
$$-\nabla \cdot \underline{\sigma} = \mathbf{0} \quad \text{in } \Omega^s \quad (\text{A.1})$$

Here,  $\underline{\sigma}$  denotes the Cauchy stress tensor. The displacement field is denoted as  $\mathbf{d}$ . The Cauchy stress  $\underline{\sigma}$  is related to the strain  $\underline{\varepsilon}$  via the linear, fourth order elasticity tensor  $\underline{\underline{C}}$  as

$$\underline{\sigma} = \underline{\underline{C}} : \underline{\varepsilon} \quad (\text{A.2})$$

$$\underline{\varepsilon} = \frac{1}{2}(\nabla \mathbf{d} + (\nabla \mathbf{d})^T) \quad (\text{A.3})$$

For brevity, volumetric forces have been omitted. The superscript for structure variables is omitted in this section, since no other fields are present. The interface variables are denoted as usual with superscript  $\cdot^i$ .



**Figure A.1:** Figure A.1a shows the physical domain with Dirichlet Boundary Conditions (DBC), while Figure A.1b illustrates how the DBC problem translates to the embedded Dirichlet problem. Physical field  $\Omega^s$  and interface  $\Gamma^i$  along with respective domain normals and variables are shown.

The domain setup including an one-sided discontinuity, where the computational domain is split into a physical part and a fictitious part is identical to the fluid description, but is repeated here for convenience. Assume a Dirichlet Boundary Condition (DBC) on parts of a boundary of a physical domain  $\Omega^s$  as depicted in Figure A.1a.

The jump in the displacements  $\llbracket \mathbf{d} \rrbracket$  between the physical values  $\mathbf{d}^+$  and the void ( $\mathbf{d}^-$ ) can be expressed as

$$\llbracket \mathbf{d} \rrbracket = \mathbf{d}^+ - \underbrace{\mathbf{d}^-}_{=0} \quad \text{in } \Gamma^i \quad (\text{A.4})$$

The jump height  $\llbracket \mathbf{d} \rrbracket$  equals the value of  $\mathbf{d}^+$  at  $\Gamma^i$  and one can pose a kinematic constraint for  $\mathbf{d}^+$  along  $\Gamma^i$  as

$$\mathbf{d}^+ - \hat{\mathbf{d}}^i = \mathbf{0} \quad \text{in } \Gamma^i \quad (\text{A.5})$$

where the interface displacement  $\hat{\mathbf{d}}^i$  is given. Likewise, a jump discontinuity exists in the stress field, which results in a surface traction vector field  $\lambda$  as

$$\llbracket \boldsymbol{\sigma} \cdot \mathbf{n} \rrbracket = \boldsymbol{\sigma}^+ \cdot \mathbf{n} - \underbrace{\boldsymbol{\sigma}^- \cdot \mathbf{n}}_{=0} = \lambda \quad \text{in } \Gamma^i \quad (\text{A.6})$$

In case of modeling a void,  $\boldsymbol{\sigma}^+ \cdot \mathbf{n} = \lambda$  is the surface traction—the ‘reaction force’—due to the displacement constraint along the interface.

The jump discontinuities are modeled by a void enrichment [Daux et al., 2000; Sukumar et al., 2001] using the XFEM [Belytschko and Black, 1999; Moës et al., 1999]. Nodes and elements are generated independent of the interface position and then an enriched approximation of intersected elements is applied using the void enrichment Eq. (3.13). The strong form of this one-sided

problem including the interface condition at  $\Gamma^i$  reads as

$$-\nabla \cdot \underline{\sigma} = \mathbf{0} \quad \text{in } \Omega^s \quad (\text{A.7a})$$

$$\underline{d} - \hat{\underline{d}}_D = \mathbf{0} \quad \text{in } \Gamma_D^s \quad (\text{A.7b})$$

$$\underline{\sigma} \cdot \underline{n} - \hat{h} = \mathbf{0} \quad \text{in } \Gamma_N^s \quad (\text{A.7c})$$

$$\underline{d} - \hat{\underline{d}}^i = \mathbf{0} \quad \text{in } \Gamma^i \quad (\text{A.7d})$$

In the one-sided problem, the domain  $\Omega^-$  is of no interest. Hence, the notation for the structure domain, namely  $\Omega^s$  instead of  $\Omega^+$  is used in the following.

## A.2 Weakly enforced Dirichlet conditions

### A.2.1 Traction Lagrange multiplier

The ‘classical’ Lagrange multiplier method for conditions along surfaces has been abandoned for the fixed grid fluid implementation. Nevertheless, it is repeated here to contrast it with the newly developed stress Lagrange multiplier method. In the Lagrange multiplier method, a Lagrange multiplier field  $\lambda(x)$  is introduced along the boundary to weakly enforce the constraint Eq. (A.7d). The resulting weighted residual equation is

$$-(\delta \underline{d}, \nabla \cdot \underline{\sigma})_{\Omega^s} + (\delta \underline{d}, \underline{\sigma} \cdot \underline{n} - \hat{h})_{\Gamma_N^s} + (\delta \underline{d}, \underline{\sigma} \cdot \underline{n} - \lambda)_{\Gamma_i} - (\delta \lambda, \underline{d} - \hat{\underline{d}}^i)_{\Gamma^i} = 0 \quad (\text{A.8})$$

where  $\delta \underline{d}$  and  $\delta \lambda$  denote the displacement and the Lagrange multiplier test functions, respectively. Integration by parts of the stress term yields

$$-(\delta \underline{d}, \nabla \cdot \underline{\sigma})_{\Omega^s} = -(\delta \underline{d}, \underline{\sigma} \cdot \underline{n})_{\Gamma} + (\nabla \delta \underline{d}, \underline{\sigma})_{\Omega^s} \quad (\text{A.9})$$

Dirichlet and Neumann conditions away from the embedded interface are treated as usual. The resulting weak form is then

$$(\nabla \delta \underline{d}, \underline{\sigma})_{\Omega^s} - (\delta \underline{d}, \hat{h})_{\Gamma_N^s} - (\delta \underline{d}, \lambda)_{\Gamma_i} - (\delta \lambda, \underline{d} - \hat{\underline{d}}^i)_{\Gamma^i} = 0 \quad (\text{A.10})$$

For completeness, the corresponding strong form is given as

$$-\nabla \cdot \underline{\sigma}^s = 0 \quad \text{in } \Omega^s \quad (\text{A.11a})$$

$$\underline{d} - \hat{\underline{d}}_D = \mathbf{0} \quad \text{in } \Gamma_D \quad (\text{A.11b})$$

$$\underline{\sigma} \cdot \underline{n} = \hat{h} \quad \text{in } \Gamma_N \quad (\text{A.11c})$$

$$\underline{d} = \hat{\underline{d}} \quad \text{in } \Gamma^i \quad (\text{A.11d})$$

$$\underline{\sigma} \cdot \underline{n} = \lambda \quad \text{in } \Gamma^i \quad (\text{A.11e})$$

The primary difficulty for this formulation is to find an appropriate Lagrange multiplier space on the interface for three-dimensional problems. Resulting problems are the saddle point structure of the discrete system and the additional number of unknowns that have to be solved.

### A.2.2 Cauchy stress based Lagrange multiplier

The key point of the stress based Lagrange multiplier is that instead a vector traction field  $\lambda$  along  $\Gamma^i$  an additional primary stress field  $\bar{\sigma}$  is introduced in intersected background elements. Hence,  $\delta\bar{\sigma} \cdot \mathbf{n}$  instead of  $\delta\lambda$  is used along  $\Gamma^i$  to weakly enforce the interface constraint Eq. (A.7d). The stress field is embedded into the intersected background element as explained in the following.

The two primary unknowns are the displacement  $\mathbf{d}$  and the Cauchy stress  $\bar{\sigma}$ . Their corresponding test functions are  $\delta\mathbf{d}$  and  $\delta\bar{\sigma}$ , respectively. Along the interface, the stress field is used to constrain the displacement field as

$$(\delta\bar{\sigma} \cdot \mathbf{n}, \mathbf{d} - \hat{\mathbf{d}}^i)_{\Gamma^i} \quad (\text{A.12})$$

where  $\delta\bar{\sigma} \cdot \mathbf{n}$  is the virtual traction along the interface.

Eq. (A.12) provides only three equations for six unknowns of the symmetric stress tensor  $\bar{\sigma}$ . Having two primary unknowns in  $\Omega$ , an additional equation closes the set of equations in  $\Omega$  by matching the strains computed from both unknowns

$$\bar{\boldsymbol{\varepsilon}} - \boldsymbol{\varepsilon} = \mathbf{0} \quad \text{in } \Omega \quad (\text{A.13})$$

The two strains are defined in terms of the primary unknowns  $\mathbf{d}$  and  $\bar{\sigma}$  as

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\mathbf{d} + (\nabla\mathbf{d})^T) \quad (\text{A.14})$$

$$\bar{\boldsymbol{\varepsilon}} = \tilde{\mathcal{C}}^{-1} : \bar{\sigma} = \tilde{\mathcal{S}} : \bar{\sigma} \quad (\text{A.15})$$

The fourth order tensors  $\tilde{\mathcal{C}}$  and  $\tilde{\mathcal{S}}$  represent the material and compliance tensor, respectively. The new task is then: find  $\mathbf{d}$  and  $\bar{\sigma}$  such that

$$\nabla \cdot \sigma = \mathbf{0} \quad \text{in } \Omega \quad (\text{A.16a})$$

$$\bar{\boldsymbol{\varepsilon}} - \boldsymbol{\varepsilon} = \mathbf{0} \quad \text{in } \Omega \quad (\text{A.16b})$$

$$\mathbf{d} - \hat{\mathbf{d}}_D = \mathbf{0} \quad \text{in } \Gamma_D \quad (\text{A.16c})$$

$$\sigma \cdot \mathbf{n} - \hat{h} = \mathbf{0} \quad \text{in } \Gamma_N \quad (\text{A.16d})$$

$$\mathbf{d} - \hat{\mathbf{d}}^i = \mathbf{0} \quad \text{in } \Gamma^i \quad (\text{A.16e})$$

$$\sigma \cdot \mathbf{n} - \bar{\sigma} \cdot \mathbf{n} = \mathbf{0} \quad \text{in } \Gamma^i \quad (\text{A.16f})$$

The Cauchy stress  $\sigma$  is computed from the displacement unknowns via  $\boldsymbol{\varepsilon}$  using Eq. (A.2). The resulting weighted residual equation is

$$\begin{aligned} & -(\delta\mathbf{d}, \nabla \cdot \sigma)_{\Omega^s} - (\delta\bar{\sigma}, \bar{\boldsymbol{\varepsilon}} - \boldsymbol{\varepsilon})_{\Omega^s} \\ & + (\delta\mathbf{d}, \sigma \cdot \mathbf{n} - \hat{h})_{\Gamma_N^s} + (\delta\mathbf{d}, \sigma \cdot \mathbf{n} - \bar{\sigma} \cdot \mathbf{n})_{\Gamma^i} - (\delta\bar{\sigma} \cdot \mathbf{n}, \mathbf{d} - \hat{\mathbf{d}}^i)_{\Gamma^i} = 0 \end{aligned} \quad (\text{A.17})$$

Integrating by parts using Eq. (A.9), the final weak form is given by

$$(\nabla \delta \mathbf{d}, \boldsymbol{\sigma})_{\Omega^s} - (\delta \tilde{\boldsymbol{\sigma}}, \bar{\boldsymbol{\varepsilon}} - \boldsymbol{\varepsilon})_{\Omega^s} - (\delta \mathbf{d}, \hat{\mathbf{h}})_{\Gamma_N^s} - (\delta \mathbf{d}, \tilde{\boldsymbol{\sigma}} \cdot \mathbf{n})_{\Gamma^i} - (\delta \tilde{\boldsymbol{\sigma}} \cdot \mathbf{n}, \mathbf{d} - \hat{\mathbf{d}}^i)_{\Gamma^i} = 0 \quad (\text{A.18})$$

Along  $\Gamma^i$ , both displacement and stress field are discontinuous and enriched with  $\bar{\psi}(\mathbf{x})$ . The complete discretizations for trial and test functions are given as

$$\mathbf{d}^h(\xi^\Omega) = \sum_L \phi_L^d(\xi^\Omega) \mathbf{d}_L \quad (\text{A.19})$$

$$\delta \mathbf{d}^h(\xi^\Omega) = \sum_L \phi_L^d(\xi^\Omega) \delta \mathbf{d}_L \quad (\text{A.20})$$

$$\tilde{\boldsymbol{\sigma}}^h(\xi^\Omega) = \sum_L \phi_L^{\tilde{\boldsymbol{\sigma}}}(\xi^\Omega) \tilde{\boldsymbol{\sigma}}_L \quad (\text{A.21})$$

$$\delta \tilde{\boldsymbol{\sigma}}^h(\xi^\Omega) = \sum_L \phi_L^{\tilde{\boldsymbol{\sigma}}}(\xi^\Omega) \delta \tilde{\boldsymbol{\sigma}}_L \quad (\text{A.22})$$

The approximation functions are the usual combination of shape functions and appropriate enrichment functions

$$\phi_L(\xi^\Omega) = N_L(\xi^\Omega) \psi(\xi^\Omega) \quad (\text{A.23})$$

The displacement shape functions  $N_I^d(\mathbf{x})$  are chosen as piecewise continuous polynomials, while being  $C^0$ -continuous at inter-element boundaries. The shape functions for the stress unknowns  $N_K^{\tilde{\boldsymbol{\sigma}}}(\mathbf{x})$  shall also be polynomial functions inside each element, however, they shall be  $C^{-1}$  discontinuous at inter-element boundaries. Hence, stress unknowns are element DOFs and the discontinuous stress approximations fit the elementwise discontinuous strains based on derivatives of the  $C^0$  continuous displacement shape functions.

The global system is assembled from element stiffness matrices based on Eq. (A.18) as

$$\mathbf{A}_e \begin{bmatrix} \mathbf{K}_{dd} & \mathbf{G}_{d\tilde{\boldsymbol{\sigma}}} \\ \mathbf{K}_{\tilde{\boldsymbol{\sigma}}d} + \mathbf{G}_{\tilde{\boldsymbol{\sigma}}d} & \mathbf{K}_{\tilde{\boldsymbol{\sigma}}\tilde{\boldsymbol{\sigma}}} \end{bmatrix}_e \mathbf{A}_e \begin{bmatrix} \mathbf{d} \\ \tilde{\boldsymbol{\sigma}} \end{bmatrix}_e = \mathbf{A}_e \begin{bmatrix} \mathbf{f}_N \\ \mathbf{G}_{\tilde{\boldsymbol{\sigma}}d} \hat{\mathbf{d}}^i \end{bmatrix} \quad (\text{A.24})$$

The nodal displacements for this element are denoted as  $\mathbf{d}$ , element stress unknowns as  $\tilde{\boldsymbol{\sigma}}$  and discrete forces resulting from Neumann conditions as  $\mathbf{f}_N$ . If the interface is discretized, the prescribed interface displacement is denoted with  $\hat{\mathbf{d}}^i$ . The element matrices  $\mathbf{K}$  correspond to operators evaluated on the element domain  $\Omega^e$ , while all  $\mathbf{G}$  matrices are related to boundary integrals over that part of the interface, that crosses the element domain, namely  $\Gamma^{e+} = \Gamma^i \cap \Omega^e$ . The element matrices relate to the weak form integrals as

$$\mathbf{K}_{dd} : -(\nabla \delta \mathbf{d}, \boldsymbol{\sigma})_{\Omega^e} \quad (\text{A.25})$$

$$\mathbf{K}_{\tilde{\boldsymbol{\sigma}}d} : +(\delta \tilde{\boldsymbol{\sigma}}, \boldsymbol{\varepsilon})_{\Omega^e} \quad (\text{A.26})$$

$$\mathbf{K}_{\tilde{\boldsymbol{\sigma}}\tilde{\boldsymbol{\sigma}}} : -(\delta \tilde{\boldsymbol{\sigma}}, \bar{\boldsymbol{\varepsilon}} - \boldsymbol{\varepsilon})_{\Omega^e} \quad (\text{A.27})$$

and

$$\mathbf{G}_{d\bar{\sigma}} : -(\delta d, \bar{\sigma} \cdot \mathbf{n})_{\Gamma^{e+}} \quad (\text{A.28})$$

$$\mathbf{G}_{\bar{\sigma}d} : -(\delta \bar{\sigma} \cdot \mathbf{n}, d)_{\Gamma^{e+}} \quad (\text{A.29})$$

$$\mathbf{G}_{\bar{\sigma}d^i} : -(\delta \bar{\sigma} \cdot \mathbf{n}, \hat{d}^i)_{\Gamma^{e+}} \quad (\text{A.30})$$

One can now distinguish between element tangent stiffness matrices for intersected and not intersected elements. For intersected elements, due to the elementwise discontinuous flux approximation the element flux unknowns can be expressed as

$$\bar{\sigma} = \mathbf{K}_{\bar{\sigma}\bar{\sigma}}^{-1}(-(\mathbf{K}_{\bar{\sigma}d} + \mathbf{G}_{\bar{\sigma}d})\mathbf{d} + \mathbf{G}_{\bar{\sigma}d^i}\hat{d}^i) \quad (\text{A.31})$$

This allows the condensation of the stress unknowns at the element level and leaves only displacement unknowns as

$$[\mathbf{K}_{dd} - \mathbf{G}_{d\bar{\sigma}}\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^{-1}(\mathbf{K}_{\bar{\sigma}d} + \mathbf{G}_{\bar{\sigma}d})] [\mathbf{d}]_e \text{ and } [\mathbf{f}_N - \mathbf{G}_{d\bar{\sigma}}\mathbf{K}_{\bar{\sigma}\bar{\sigma}}^{-1}\mathbf{G}_{\bar{\sigma}d^i}\hat{d}^i]_e \quad (\text{A.32})$$

For elements that are not intersected by  $\Gamma^i$ , no boundary integrals exist and element tangent stiffness and right-hand side vector reduce to

$$\begin{bmatrix} \mathbf{K}_{dd} & \mathbf{0} \\ \mathbf{K}_{\bar{\sigma}d} & \mathbf{K}_{\bar{\sigma}\bar{\sigma}} \end{bmatrix}_e \text{ and } \begin{bmatrix} \mathbf{f}_N \\ \mathbf{0} \end{bmatrix}_e \quad (\text{A.33})$$

It follows that the solution of  $\mathbf{d}$  is independent of the stress solution in such an element and only nodal displacement are left as

$$[\mathbf{K}_{dd}]_e \text{ and } [\mathbf{f}_N]_e \quad (\text{A.34})$$

Hence, all stress unknowns in non-intersected elements can be omitted without effect on the global displacement solution.

The model problem in this section has not been implemented, hence it is only possible to speculate on the proper shape function order for the stress field. Based on the experience with the Poisson equation and the incompressible NS equations, the stress shape function should be of the same order as the displacement derivatives or higher. Future mathematical analysis may clarify the optimal choice of shape functions.

In summary, the constraining Lagrange multiplier field is shifted from the interface to the background domain by replacing  $\lambda$  and  $\delta\lambda$  with  $\bar{\sigma} \cdot \mathbf{n}$  and  $\delta\bar{\sigma} \cdot \mathbf{n}$ , respectively. No interface mesh is required to discretize a Lagrange multiplier field. The two primary fields in  $\Omega^s$ , namely displacement and Cauchy stress, are weakly coupled by matching the strain tensors computed from each of the two fields. Hence, the stress is defined by its relation to the displacement even without an interface integral present in a particular element.

### A.3 Relation to the HR variational principle and Nitsche's method

As already discussed in Section 3.3, the variational equations have many similarities with hybrid/mixed types of element technology, for example based on the Hellinger-Reissner principle. These methods introduce additional unknowns to improve the stress and/or strain approximations, for example to fight locking phenomena [Pian and Wu, 2006]. In the following, the relation of the proposed weak form to the HR variational principle and Nitsche's method is discussed to point out some differences and commonality between these approaches.

The hybrid/mixed weak form from Eq. (A.18) was given as

$$(\nabla \delta \mathbf{d}, \boldsymbol{\sigma})_{\Omega^s} - (\delta \bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\xi}} - \boldsymbol{\xi})_{\Omega^s} - (\delta \mathbf{d}, \hat{\mathbf{h}})_{\Gamma_N^s} - (\delta \mathbf{d}, \boldsymbol{\sigma} \cdot \mathbf{n})_{\Gamma^i} - (\delta \bar{\boldsymbol{\sigma}} \cdot \mathbf{n}, \mathbf{d} - \hat{\mathbf{d}}^i)_{\Gamma^i} = 0 \quad (\text{A.35})$$

Replacing  $\boldsymbol{\sigma}(\mathbf{d})$  with the primary stress unknown  $\bar{\boldsymbol{\sigma}}$  *only* in the first integral, the following variational form

$$(\nabla \delta \mathbf{d}, \bar{\boldsymbol{\sigma}})_{\Omega^s} - (\delta \bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\xi}} - \boldsymbol{\xi})_{\Omega^s} - (\delta \mathbf{d}, \hat{\mathbf{h}})_{\Gamma_N^s} - (\delta \mathbf{d}, \bar{\boldsymbol{\sigma}} \cdot \mathbf{n})_{\Gamma^i} - (\delta \bar{\boldsymbol{\sigma}} \cdot \mathbf{n}, \mathbf{d} - \hat{\mathbf{d}}^i)_{\Gamma^i} = 0 \quad (\text{A.36})$$

is obtained. This symmetric weak form corresponds to the variation of the HR functional for linear elasticity. For an overview on such multi-variable formulations see e.g. [Pian and Wu, 2006, chap 1.3]. The corresponding matrix structure is

$$\mathbf{A}_e \begin{bmatrix} 0 & K_{d\bar{\sigma}} + G_{d\bar{\sigma}} \\ K_{\bar{\sigma}d} + G_{\bar{\sigma}d} & K_{\bar{\sigma}\bar{\sigma}} \end{bmatrix}_e \mathbf{A}_e \begin{bmatrix} \mathbf{d} \\ \bar{\boldsymbol{\sigma}} \end{bmatrix}_e = \mathbf{A}_e \begin{bmatrix} f_N \\ G_{\bar{\sigma}d^i} \hat{\mathbf{d}}^i \end{bmatrix} \quad (\text{A.37})$$

and in condensed form

$$[-(K_{d\bar{\sigma}} + G_{d\bar{\sigma}})K_{\bar{\sigma}\bar{\sigma}}^{-1}(K_{\bar{\sigma}d} + G_{\bar{\sigma}d})] [\mathbf{d}]_e \text{ and } [f_N - G_{d\bar{\sigma}}K_{\bar{\sigma}\bar{\sigma}}^{-1}G_{\bar{\sigma}d^i}\hat{\mathbf{d}}^i]_e \quad (\text{A.38})$$

Both the condensed and uncondensed system are symmetric. For details on the HR formulation, see for example [Pian and Wu, 2006].

Nitsche's method [Nitsche, 1971] does not introduce additional primary variables. Writing Nitsche's method in a form similar to Eqs. (A.35) and (A.36) gives

$$(\nabla \delta \mathbf{d}, \boldsymbol{\sigma})_{\Omega^s} - (\delta \mathbf{d}, \hat{\mathbf{h}})_{\Gamma_N^s} - (\delta \mathbf{d}, \boldsymbol{\sigma} \cdot \mathbf{n})_{\Gamma^i} - (\delta \boldsymbol{\sigma} \cdot \mathbf{n}, \mathbf{d} - \hat{\mathbf{d}}^i)_{\Gamma^i} - \alpha(\delta \mathbf{d}, \mathbf{d} - \hat{\mathbf{d}}^i)_{\Gamma^i} = 0 \quad (\text{A.39})$$

Here, *all* stress terms are based on nodal displacements. The method is reported to be stable only if an additional stabilization or penalty term along

the boundary is introduced. For choices of the user-defined parameter  $\alpha$ , see e.g. [Dolbow and Harari, 2009], for an application of Nitsche's method to elasticity problems, see e.g. [Fernández-Méndez and Huerta, 2004].

Comparing these three approaches, all of them result in element stiffness matrices with only displacement unknowns. No interface mesh and interface unknowns are present, which completely removes the need for an interface discretization. However, only the HR-approach and Nitsche's method result in symmetric formulations — provided the physical problem itself is symmetric. For elastic problems or Poisson's equation as used in heat conduction problems, switching to asymmetric solvers might be regarded as a suboptimal choice. For asymmetric problems like solving the NS equations, this asymmetry does not present additional difficulties.

Furthermore, the HR approach and the proposed mixed/hybrid form is free of user-defined parameter, however, more computational work for intersected elements is required to condense the stress unknowns in intersected elements. Without an in-depth mathematical analysis, one can only speculate on the stability properties of the proposed approach. Here, as well as for the HR formulation, it seems that a proper pair of function spaces leads to a stable formulation, whereas Nitsche's formulation is reported to be stable only with proper choices of parameter  $\alpha$ . For the approximations spaces as given in Section 3.3.4 and Section 4.3.2, no stability problems have been observed in any simulations.

Finally, Nitsche's method and the stress Lagrange multiplier approach leave the virtual elastic energy  $(\nabla \delta \mathbf{d}, \boldsymbol{\sigma})_{\Omega^s}$  untouched. Hence, the boundary constraints could be added to other advanced existing element formulation. When choosing an appropriate method for applying weak Dirichlet conditions to the incompressible NS equation, both approaches leave the fine-tuned equal-order velocity-pressure formulation intact, whereas an equivalent to HR would require at least a partial rewrite and modification of the element formulation and implementation.

# APPENDIX B

## Reference solution for the Jeffery-Hamel flow

The function  $u_r(\phi)$  as solution for the differential equation Eq. (4.68) can not be computed explicitly. Hence, before this particular flow field can be used for a convergence study, an explicit approximation of the solution is required. As a remedy, a numerical solution was sought using the commercial computer algebra system (CAS) Mathematica 6.0 [Inc., 2007]. Note the helpful discussion on specific properties of the ODE presented in [Corless and Assefa, 2007]. The numerical solution has subsequently been fitted to a polynomial, which has then been used as reference solution in the C++ code. The required commands for Mathematica are summarized below. The angle  $\phi$  is replaced by variable  $x$  to simplify the notation. Removing any semicolon at the end of a line produces intermediate output that is otherwise suppressed.

First, the numerical solution is sought by specifying the differential equation, three boundary and symmetry conditions, the solution variable  $u[x]$ , the range where the solution is sought and the solution algorithm

```
us = Flatten[
 NDSolve[
 {
 u'''[x] + 4 u'[x] + 2 u'[x] u[x] == 0,
 u'[Pi/8] == 0,
 u[0] == 0,
 u[Pi/8] == -170
 },
 u,
 {x, 0, Pi/4},
```

```

    Method -> "Adams"
]
]
```

The resulting solution for  $u[x]$  can not directly be used in C++ code, hence a polynomial fit has to be generated first. The next commands generate a polynomial that is symmetric with respect to  $\pi/8$  and has polynomial order 32.

```

indmax = 16;
ufittemp[x_] =
Sum[
  k[i]*(x - Pi/8)^(2 i), {i, 0, indmax}
];
```

The general polynomial `ufittemp` is fitted to the numerical solution `us` using 1001 sample points by

```

var = k /@ (Range[indmax + 1] - 1);
numsp = 1000
sp = Pi/4/numsp*(Range[numsp+1] - 1);
fitsol =
  FindFit[
    Thread[{sp, u[sp] /. us}],
    {ufittemp[x]},
    var,
    x
];
```

and the resulting `k[i]` in the array `fitsol` is inserted into the general polynomial `ufittemp` as

```
ufit[x_] = ufittemp[x] /. fitsol;
```

Plotting the fitted polynomial is done by the following command

```

Plot[
  ufit[x], {x, 0, Pi/4}, AxesOrigin -> {0, 0},
  AxesLabel -> {Phi, Subscript[u, r]},
  Ticks -> {{0, Pi/16, Pi/8, (3/16) Pi, Pi/4}},
  PlotStyle -> {Thick},
  LabelStyle -> {FontSize -> 14}
]
```

Following [Corless and Assefa, 2007], the Reynolds number for this setup is about 85 and is computed by

```
Reynoldsnumber =
-3.0/4.0 NIntegrate[ufit[x], {x, 0, Pi/4}]
```

An syntactically almost correct C code is generated by using

```
ufit[x] // CForm
```

The fitted function is used as reference solution and boundary condition for inflow and outflow in Section 4.5.3. Below, the function to compute the radial velocity depending on the angle  $\alpha$  is given as C++ code

```
double JefHamRadialVelocity(
    const double& theta
)
{
if ( -1.e-8 < theta or theta < ((M_PI/4.0) + 1.e-8) )
{
    cout << "Angle out of range! ";
    cout << "Must be between 0 and PI/4" " << endl;
    exit(1);
}

const double alpha = theta - (M_PI/8.0);

// generated by Mathematica 6
return -169.99995631379676
    + 51.72093368870616* pow(alpha, 2)
    + 1453.2480022627062* pow(alpha, 4)
    + 15469.249826734282* pow(alpha, 6)
    + 153515.23050166125* pow(alpha, 8)
    - 3.1393239563288596e6* pow(alpha, 10)
    + 1.3791074955303007e8* pow(alpha, 12)
    - 3.9054873809559045e9* pow(alpha, 14)
    + 8.064960114312076e10* pow(alpha, 16)
    - 1.2314175442399622e12*pow(alpha, 18)
    + 1.3952367263056582e13*pow(alpha, 20)
    - 1.1694967181678298e14*pow(alpha, 22)
    + 7.149819830278836e14* pow(alpha, 24)
    - 3.0970644442180215e15*pow(alpha, 26)
```

```

+ 9.001387683687223e15* pow(alpha,28)
- 1.5737446665792184e16*pow(alpha,30)
+ 1.250536015803445e16* pow(alpha,32);
}

```

Here,  $\theta$  is the angle starting from the  $x$ -axis. The polynomial is given as symmetric function such that  $\theta$  is first transformed to  $\alpha$  by subtracting  $\pi/8$ . The resulting profile at radius  $r = 1$  over  $\theta$  is plotted in Figure B.1.

The  $x$  and  $y$  velocity components based on the radial velocity are then computed by

```

const double x = physpos(0);
const double y = physpos(1);

const double theta = atan(y/x);
const double u_theta = JefHamRadialVelocity(theta);

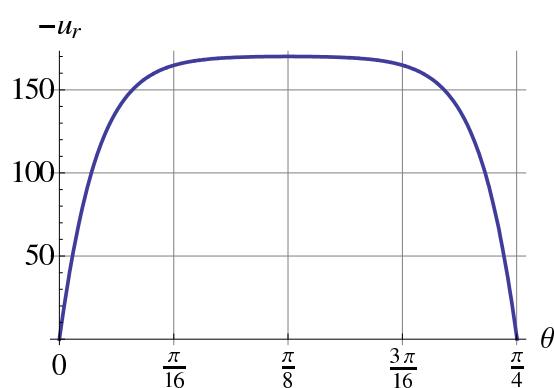
const double u_exact_x = (u_theta/(x*x+y*y))*x;
const double u_exact_y = (u_theta/(x*x+y*y))*y;

if (1.0 < x and x < 2.0 and 0.0 < y and y < x)
{
    const double eps_x = (gpvelnp(0) - u_exact_x);
    const double eps_y = (gpvelnp(1) - u_exact_y);

    L2squared += (eps_x*eps_x + eps_y*eps_y)*fac;
}

```

Here, `physpos` contains the coordinates of the integration point in physical coordinates  $x$  and `fac` represents the product of integration weights and Jacobi determinants. The interpolated velocity  $\mathbf{u}^{f,h}$  at the integration point is stored in `gpvelnp`. After a loop over all integration points in all XFEM integration cells and all elements, `L2squared` contains the squared  $L_2$  norm as defined in Eq. (4.69).



**Figure B.1:** Jeffery-Hamel flow: Profile of radial velocity  $u_r(\theta)$  at  $r = 1$ .



# Bibliography

- J. Ahrens, B. Geveci, and C. Law. *Visualization Handbook*, chapter ParaView: An End-User Tool for Large Data Visualization. Elsevier, 2005.
- M. Arienti, P. Hung, E. Morano, and J. E. Shepherd. A level set approach to Eulerian-Lagrangian coupling. *Journal of Computational Physics*, 185(1): 213–251, 2003.
- M. Astorino, J.-F. Gerbeau, O. Pantz, and K.-F. Traoré. Fluid-structure interaction and multi-body contact: Application to aortic valves. *Computer Methods in Applied Mechanics and Engineering*, 198(45-46):3603–3612, 2009. ISSN 0045-7825.
- F. P. T. Baaijens. A fictitious domain/mortar element method for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 35(7):743–761, 2001.
- I. Babuška and M. Suri. The p- and h-p version of the finite element method, an overview. In *ICOSAHOM '89: Proceedings of the conference on Spectral and high order methods for partial differential equations*, pages 5–26, Amsterdam, The Netherlands, 1990. North-Holland Publishing Co.
- I. M. Babuška and J. M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40(4):727–758, 1997.
- H. J. C. Barbosa and T. J. R. Hughes. Circumventing the Babuška-Brezzi condition in mixed finite element approximations of elliptic variational inequalities. *Computer Methods in Applied Mechanics and Engineering*, 97(2):193–210, 1992. ISSN 0045-7825.
- G. R. Barrenechea and F. Valentin. An unusual stabilized finite element method for a generalized Stokes problem. *Numerische Mathematik*, 92(4): 653–677, 2002.

- Y. Bazilevs and T. Hughes. Nurbs-based isogeometric analysis for the computation of flows about rotating components. *Computational Mechanics*, 43(1):143–150, 2008.
- R. Becker, P. Hansbo, and R. Stenberg. A finite element method for domain decomposition with non-matching grids. *Mathematical Modelling and Numerical Analysis*, 37(2):209–225, 2003.
- M. Behr. Simplex space-time meshes in finite element simulations. *International Journal for Numerical Methods in Fluids*, 57(9):1421–1434, 2008.
- M. Behr and T. Tezduyar. Shear-slip mesh update in 3d computation of complex flow problems with rotating mechanical components. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3189–3200, 2001. ISSN 0045-7825.
- M. Behr and T. E. Tezduyar. The shear-slip mesh update method. *Computer Methods in Applied Mechanics and Engineering*, 174:261–274, 1999.
- M. A. Behr, L. P. Franca, and T. E. Tezduyar. Stabilized finite element methods for the velocity-pressure-stress formulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 104(1):31–48, 1993.
- T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620, 1999.
- T. Belytschko and J. M. Kennedy. Computer models for subassembly simulation. *Journal of Nuclear Engineering and Design*, 49:17–38, 1978.
- T. Belytschko, J. M. Kennedy, and D. Schoeberle. Quasi-Eulerian finite element formulation for fluid structure interaction. *Journal of Pressure Vessel Technology*, 102:62–69, 1980.
- T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons, Inc., 2000. ISBN 015-Belytschko/2000-1.
- T. Belytschko, N. Moës, S. Usui, and C. Parimi. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering*, 50(4):993–1013, 2001.
- C. Bernardi, Y. Maday, and A. T. Patera. *Asymptotic and numerical methods for partial differential equations with critical parameters*, volume 384, chapter Domain decomposition by the mortar element method, pages 269–286. Reidel, Dordrecht, 1993.

- C. Bernardi, Y. Maday, and A. T. Patera. A new nonconforming approach to domain decomposition: the mortar element method. *Nonlinear partial differential equations and their applications*, 299:13–51, 1994.
- M. Bischoff, W. A. Wall, K.-U. Bletzinger, and E. Ramm. Models and finite elements for thin-walled structures. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 2, pages 59–137. John Wiley & Sons, Ltd., 2004.
- J. Burkardt. Quadrature rules for tetrahedrons. online, 2010. URL [http://people.scs.fsu.edu/~burkardt/datasets/quadrature\\_rules\\_tet/quadrature\\_rules\\_tet.html](http://people.scs.fsu.edu/~burkardt/datasets/quadrature_rules_tet/quadrature_rules_tet.html).
- E. Béchet, H. Minnebo, N. Moës, and B. Burgardt. Improved implementation and robustness study of the X-FEM for stress analysis around cracks. *International Journal for Numerical Methods in Engineering*, 64(8):1033–1056, 2005.
- K. W. Cheng and T.-P. Fries. Higher-order XFEM for curved strong and weak discontinuities. *International Journal for Numerical Methods in Engineering*, 82(5):564–590, 2010.
- J. Chessa and T. Belytschko. An extended finite element method for two-phase fluids. *Journal of Applied Mechanics, Transactions ASME*, 70(1):10–17, 2003.
- J. Chessa and T. Belytschko. Arbitrary discontinuities in space-time finite elements by level sets and X-FEM. *International Journal for Numerical Methods in Engineering*, 61(15):2595–2614, 2004.
- J. Chessa, P. Smolinski, and T. Belytschko. The extended finite element method (XFEM) for solidification problems. *International Journal for Numerical Methods in Engineering*, 53(8):1959–1977, 2002.
- F. Cirak and R. Radovitzky. A Lagrangian-Eulerian shell-fluid coupling algorithm based on level sets. *Computers & Structures*, 83(6-7):491–498, 2005.
- R. Codina and J. Baiges. Approximate imposition of boundary conditions in immersed boundary methods. *International Journal for Numerical Methods in Engineering*, 80(11):1379–1405, 2009.
- R. Codina and J. Blasco. Analysis of a stabilized finite element approximation of the transient convection-diffusion-reaction equation using orthogonal subscales. *Computing and Visualization in Science*, 4(3):167–174, 2002.

- R. Codina, G. Houzeaux, H. Coppola-Owen, and J. Baiges. The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of Computational Physics*, 228(5):1591–1611, 2009. ISSN 0021-9991.
- R. M. Corless and D. Assefa. Jeffery-Hamel flow with maple: a case study of integration of elliptic functions in a CAS. In *ISSAC '07: Proceedings of the 2007 international symposium on Symbolic and algebraic computation*, pages 108–115, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-743-8.
- H. L. D. Cougny and M. S. Shephard. Parallel refinement and coarsening of tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, 46(7):1101–1125, 1999.
- C. Daux, N. Moës, J. Dolbow, N. Sukumar, and T. Belytschko. Arbitrary cracks and holes with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 48(12):1741–1760, 2000.
- J. De Hart, G. W. Peters, P. J. Schreurs, and F. P. Baaijens. A two-dimensional fluid-structure interaction model of the aortic valve. *Journal of Biomechanics*, 33(9):1079–1088, 2000.
- N. Diniz dos Santos, J.-F. Gerbeau, and J.-F. Bourgat. A partitioned fluid-structure algorithm for elastic thin valves with contact. *Computer Methods in Applied Mechanics and Engineering*, 197(19-20):1750–1761, 2008. ISSN 0045-7825.
- J. Dolbow and L. Franca. Residual-free bubbles for embedded Dirichlet problems. *Computer Methods in Applied Mechanics and Engineering*, 197:3751–3759, 2008.
- J. Dolbow and I. Harari. An efficient finite element method for embedded interface problems. *International Journal for Numerical Methods in Engineering*, 78(2):229–252, 2009.
- J. Dolbow, N. Moës, and T. Belytschko. Discontinuous enrichment in finite elements with a partition of unity method. *Finite Elements in Analysis and Design*, 36(3-4):235–260, 2000. ISSN 0168-874X.
- J. Donéa and A. Huerta. *Finite Element Methods for Flow Problems*. Wiley, first edition, 2003.
- J. Donéa, P. Fasoli-Stella, and S. Giuliani. Lagrangian and Eulerian finite element techniques for transient fluid-structure interaction problems. In *Trans. 4th Int. Conf. on Structural Mechanics in Reactor Technology*, 1977.

- H. Engels. *Numerical quadrature and cubature*. Academic Press, London, New York, 1980. ISBN 012238850 012238850.
- D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183:83–116, 2002.
- R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457–492, 1999.
- S. Fernández-Méndez and A. Huerta. Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 193(12-14):1257–1275, 2004. ISSN 0045-7825.
- B. Flemisch and B. I. Wohlmuth. Stable Lagrange multipliers for quadrilateral meshes of curved interfaces in 3d. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1589–1602, 2007.
- L. P. Franca and F. Valentin. On an improved unusual stabilized finite element method for the advective-reactive-diffusive equation. *Computer Methods in Applied Mechanics and Engineering*, 190(13-14):1785–1800, 2000. ISSN 0045-7825.
- T.-P. Fries and A. Zilian. On time integration in the XFEM. *International Journal for Numerical Methods in Engineering*, 79(1):69–93, 2009.
- C. Förster. *Robust methods for fluid-structure interaction with stabilised finite elements*. PhD thesis, Universität Stuttgart, Institut für Baustatik, 2007.
- C. Förster, W. A. Wall, and E. Ramm. On the geometric conservation law in transient flow calculations on deforming domains. *International Journal for Numerical Methods in Fluids*, 50(12):1369–1379, 2006.
- C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196 (7):1278–1293, 2007.
- P. Gamnitzer and W. A. Wall. An ALE-Chimera method for large deformation fluid structure interaction. In P. Wesseling, E. Oñate, and J. Périoux, editors, *European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2006*. TU Delft, 2006.
- D. K. Gartling. Multipoint constraint methods for moving body and non-contiguous mesh simulations. *International Journal for Numerical Methods in Fluids*, 47(6-7):471–489, 2005.

- A. Gerstenberger and W. A. Wall. Enhancement of fixed-grid methods towards complex fluid-structure interaction applications. *International Journal for Numerical Methods in Fluids*, 57(9):1227–1248, 2008a.
- A. Gerstenberger and W. A. Wall. An extended finite element method / Lagrange multiplier based approach for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 197(19-20):1699–1714, 2008b.
- A. Gerstenberger and W. A. Wall. An embedded Dirichlet formulation for 3d continua. *International Journal for Numerical Methods in Engineering*, 82(5): 537–563, 2010.
- A. Gerstenberger, U. M. Mayer, and W. A. Wall. A 3d XFEM/Lagrange multiplier based approach for fluid-structure interaction. 8th World Congress on Computational Mechanics / 5th European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2008, Venice, 2008.
- C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Royal Astronomical Society, Monthly Notices*, 181:375–389, 1977.
- R. Glowinski, T.-W. Pan, and J. Périaux. A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 112(1-4):133–148, 1994.
- R. Glowinski, T.-W. Pan, T. I. Hesla, and D. D. Joseph. A distributed Lagrange multiplier / fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 25(5):755–794, 1999.
- P. Gresho and R. Sani. *Incompressible Flow and the Finite Element Method: Volume 2, Isothermal Laminar Flow*, volume 2. John Wiley & Sons, LTD, 2000.
- W. Gui and I. Babuška. The h, p and h-p versions of the finite element methods in 1 dimension . part iii. the adaptive h-p version. *Numer. Math.*, 49(6): 659–683, 1986. ISSN 0029-599X.
- G. Hamel. Spiralförmige Bewegungen zäher Flüssigkeiten. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 25:34–60, 1916.

- A. Hansbo and P. Hansbo. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 191(47-48):5537–5552, 2002. ISSN 0045-7825.
- A. Hansbo and P. Hansbo. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(33-35):3523–3540, 2004.
- P. Hansbo, J. Hermansson, and T. Svedberg. Nitsche's method combined with space-time finite elements for ale fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41):4195–4206, 2004. ISSN 0045-7825.
- E. Hellinger. Der Allgemeine Ansatz der Mechanik der Kontinua. *Encycl. Math. Wiss.*, 4:602, 1914.
- P. Hemker, W. Hoffmann, and M. van Raalte. Discontinuous Galerkin discretization with embedded boundary conditions. *Computational Methods in Applied Mathematics*, 3:135–158, 2003.
- A. Henderson. *ParaView Guide, A Parallel Visualization Application*. Kitware Inc., 2007.
- M. A. Heroux and J. M. Willenbring. Trilinos Users Guide. Technical Report SAND2003-2952, Sandia National Laboratories, 2003.
- M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the Trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005. ISSN 0098-3500.
- C. Hirth, A. A. Amsden, and J. Cook. An Arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14:227–253, 1974.
- J.-M. Hong, T. Shinar, M. Kang, and R. Fedkiw. On boundary condition capturing for multiphase interfaces. *Journal of Scientific Computing*, 31(1):99–125, 2007.
- G. Houzeaux and R. Codina. Transmission conditions with constraints in finite element domain decomposition methods for flow problems. *Communications in Numerical Methods in Engineering*, 17(3):179–190, 2001.

- G. Houzeaux and R. Codina. A Chimera method based on a Dirichlet/Neumann (Robin) coupling for the Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 192(31-32):3343–3377, 2003.
- T. Hughes. *The Finite Element Method – Linear Statics and Dynamic Finite Element Analysis*. Prentice Hall, 1987.
- T. Hughes, G. Scovazzi, and L. Franca. Multiscale and stabilized methods. In E. Stein, R. de Borst, and T. Hughes, editors, *Encyclopedia of Computational Mechanics*, pages 5–59. John Wiley & Sons, Chichester, 2004.
- T. J. Hughes and W. K. Liu. Implicit - explicit finite elements in transient analysis. *Journal of Applied Mechanics*, 45:371–378, 1978.
- T. J. Hughes, W. K. Liu, and T. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29:329–349, 1981.
- T. J. R. Hughes and G. M. Hulbert. Space-time finite element methods for elastodynamics: Formulations and error estimates. *Computer Methods in Applied Mechanics and Engineering*, 66(3):339–363, 1988. ISSN 0045-7825.
- S. Idelsohn, E. Oñate, and F. D. Pin. The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal for Numerical Methods in Engineering*, 61(7):964–989, 2004.
- S. Idelsohn, E. Onate, F. D. Pin, and N. Calvo. Fluid-structure interaction using the particle finite element method. *Computer Methods in Applied Mechanics and Engineering*, 195(17-18):2100–2123, 2006.
- S. R. Idelsohn, E. Onate, and F. Del Pin. A Lagrangian meshless finite element method applied to fluid-structure interaction problems. *Computers & Structures*, 81(8-11):655–671, 2003.
- W. R. Inc. Mathematica Edition: Version 6.0. Wolfram Research, Inc. Place of publication: Champaign, IL, 2007.
- D. M. Ingram, D. M. Causon, and C. G. Mingham. Developments in cartesian cut cell methods. *Mathematics and Computers in Simulation*, 61(3-6):561–572, 2003.
- B. M. Irons and R. C. Tuck. A version of the Aitken accelerator for computer iteration. *International Journal for Numerical Methods in Engineering*, 1(3):275–277, 1969.

- G. B. Jeffery. The two-dimensional steady motion of a viscous fluid. *Philosophical magazine*, 29:455–465, 1915.
- H. Ji and J. E. Dolbow. On strategies for enforcing interfacial constraints and evaluating jump conditions with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 61(14):2508–2535, 2004.
- M. Kang, R. P. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3):323–360, 2000.
- U. K  ttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1):61–72, 2008.
- U. K  ttler and W. A. Wall. Vector extrapolation for strong coupling fluid-structure interaction solvers. *Journal of Applied Mechanics*, 76(2):1–7, 2009.
- P. Laborde, J. Pommier, Y. Renard, and M. Sala  n. High-order extended finite element method for cracked domains. *International Journal for Numerical Methods in Engineering*, 64(3):354–381, 2005.
- L. Lee and R. J. LeVeque. An immersed interface method for incompressible Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 25(3):832–856, 2003.
- A. Legay, J. Chessa, and T. Belytschko. An Eulerian-Lagrangian method for fluid-structure interaction based on level sets. *Computer Methods in Applied Mechanics and Engineering*, 195(17-18):2070–2087, 2006.
- G. Legrain, N. Mo  s, and A. Huerta. Stability of incompressible formulations enriched with X-FEM. *Computer Methods in Applied Mechanics and Engineering*, 197(21-24):1835–1849, 2008.
- R. J. LeVeque and D. Calhoun. Cartesian grid methods for fluid flow in complex geometries. In L. J. Fauci and S. Gueron, editors, *Computational Modeling in Biological Fluid Dynamics*, volume 124, pages 117–143. IMA Volumes in Mathematics and its Applications, Springer-Verlag, 2001.
- A. Lew, J. E. Marsden, M. Ortiz, and M. West. An overview of variational integrators. In L. P. Franca, T. E. Tezduyar, and A. Masud, editors, *Finite Element Methods: 1970's and beyond*. CIMNE, Barcelona, Spain, 2003.
- A. J. Lew and G. C. Buscaglia. A discontinuous-Galerkin-based immersed boundary method. *International Journal for Numerical Methods in Engineering*, 76(4):427–454, 2008.

- L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astron. J.*, 82:1013–1024, 1977.
- R. Löhner, J. D. Baum, E. Mestreau, D. Sharov, C. Charman, and D. Pelessone. Adaptive embedded unstructured grid methods. *International Journal for Numerical Methods in Engineering*, 60(3):641–660, 2004.
- R. Löhner, J. R. Cebral, F. E. Camelli, S. Appanaboyina, J. D. Baum, E. L. Mestreau, and O. A. Soto. Adaptive embedded and immersed unstructured grid techniques. *Computer Methods in Applied Mechanics and Engineering*, 197 (25-28):2173–2197, 2008. ISSN 0045-7825.
- U. Mayer and W. Wall. A finite element approach to 3d intermolecular & surface interaction of multiple flexible mesoscopic structures in fluid flow. in preparation, 2010.
- U. Mayer, A. Popp, A. Gerstenberger, and W. Wall. 3d fluid–structure-contact interaction based on a combined XFEM FSI and dual mortar contact approach. *Computational Mechanics*, 46(1):53–67, 2010.
- U. M. Mayer, A. Gerstenberger, and W. A. Wall. Interface handling for three-dimensional higher-order XFEM computations in fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 47(7):846–869, 2009.
- R. L. Meakin and N. E. Suhs. Unsteady aerodynamic simulations of multiple bodies in relative motion. *AIAA Paper*, 89-1996-CP, 1989.
- J. Mergheim, E. Kuhl, and P. Steinmann. A hybrid discontinuous Galerkin/interface method for the computational modelling of failure. *Communications in Numerical Methods in Engineering*, 20(7):511–519, 2004.
- R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, 2005.
- D. P. Mok and W. A. Wall. Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In W. Wall, K.-U. Bletzinger, and K. Schweizerhof, editors, *Trends in Computational Structural Mechanics*, pages 689–698. CIMNE: Barcelona, 2001.
- J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574, 1992.
- J. J. Monaghan. Simulating free surface flows with sph. *Journal of Computational Physics*, 110(2):399–406, 1994. ISSN 0021-9991.

- H. M. Mourad, J. Dolbow, and I. Harari. A bubble-stabilized finite element method for Dirichlet constraints on embedded interfaces. *International Journal for Numerical Methods in Engineering*, 69(4):772–793, 2007.
- S. E. Mousavi, H. Xiao, and N. Sukumar. Generalized gaussian quadrature rules on arbitrary polygons. *International Journal for Numerical Methods in Engineering*, 82(1):99–113, 2010.
- N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.
- N. Moës, E. Béchet, and M. Tourbier. Imposing Dirichlet boundary conditions in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 67(12):1641–1669, 2006.
- S. Natarajan, S. Bordas, and D. R. Mahapatra. Numerical integration over arbitrary polygonal domains based on schwarz-christoffel conformal mapping. *International Journal for Numerical Methods in Engineering*, 80(1):103–134, 2009.
- J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Sem. Univ. Hamburg*, 36:9–15, 1971.
- W. Noh. CEL: A time-dependent two-space-dimensional coupled Eulerian-Lagrangian code. In B. Alder, S. Fernbach, and M. Rotenberg, editors, *Methods in Computational Physics*, volume 3, pages 117–179. Academic Press: New York, 1964.
- S. J. Osher and R. P. Fedkiw. *Level set methods and dynamic implicit surfaces - applied mathematical sciences*, volume 153. Springer, 2003. ISBN 0-387-95482-1.
- E. Oñate, J. Arteaga, J. García, and R. Flores. Error estimation and mesh adaptivity in incompressible viscous flows using a residual power approach. *Computer Methods in Applied Mechanics and Engineering*, 195(4-6):339–362, 2006.
- K. C. Park, C. A. Felippa, and R. Ohayon. Partitioned formulation of internal fluid-structure interaction problems by localized Lagrange multipliers. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):2989–3007, 2001.

- A. Peano. Gauss-Lobatto integration of high precision tetrahedral elements. *International Journal for Numerical Methods in Engineering*, 18(2):311–313, 1982.
- C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972. ISSN 0021-9991.
- C. S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977.
- C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11(1):479–517, 2002.
- T. H. H. Pian and C.-C. Wu. *Hybrid and incompatible finite element methods*. CRC Series: Modern Mechanics and Mathematics. Chapman & Hall/CRC, 2006.
- A. Popp, M. W. Gee, and W. A. Wall. A finite deformation mortar contact formulation using a primal-dual active set strategy. *International Journal for Numerical Methods in Engineering*, 79(11):1354–1391, 2009.
- C. Pozrikidis. Finite deformation of liquid capsules enclosed by elastic membranes in simple shear flow. *Journal of Fluid Mechanics Digital Archive*, 297 (-1):123–152, 2006.
- M. A. Puso. A 3D mortar method for solid mechanics. *International Journal for Numerical Methods in Engineering*, 59(3):315–336, 2004.
- M. A. Puso, T. Laursen, and J. Solberg. A segment-to-segment mortar contact method for quadratic elements and large deformations. *Computer Methods in Applied Mechanics and Engineering*, 197(6-8):555–566, 2008.
- E. Reissner. On a variational theorem in elasticity. *J. Math. Phys.*, 29:90–95, 1950.
- L. Rosenhead. The steady two-dimensional radial flow of viscous fluid between two inclined plane walls. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 175(963):436–467, 1940. URL <http://www.jstor.org/stable/97501>.
- J. Réthoré, A. Gravouil, and A. Combescure. An energy-conserving scheme for dynamic crack growth using the eXtended finite element method. *International Journal for Numerical Methods in Engineering*, 63(5):631–659, 2005.
- O. Sahni, J. Müller, K. Jansen, M. Shephard, and C. Taylor. Efficient anisotropic adaptive discretization of the cardiovascular system. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5634–5655, 2006.

- J. D. Sanders, J. E. Dolbow, and T. A. Laursen. On methods for stabilizing constraints over enriched interfaces in elasticity. *International Journal for Numerical Methods in Engineering*, 78(9):1009–1036, 2009.
- S. Sathe and T. Tezduyar. Modeling of fluid–structure interactions with the space–time finite elements: contact problems. *Computational Mechanics*, 43(1):51–60, 2008.
- M. Schäfer and S. Turek. *Flow Simulation with High-Performance Computers II, Notes on Numerical Fluid Mechanics*, volume 52, article Benchmark Computations of Laminar Flow Around a Cylinder, pages 547–566. Vieweg, 1996.
- J. Sethian. *Level set methods and fast marching methods - evolving interfaces in comp. geometry, fluid mechanics, computer vision, and materials sciences*. Cambridge University Press, 2 edition, 1999. ISBN 0-521-64557-3.
- X. Shi and N. Phan-Thien. Distributed Lagrange multiplier/fictitious domain method in the framework of lattice Boltzmann method for fluid-structure interactions. *Journal of Computational Physics*, 206:81–94, 2005.
- H. Si. Constrained Delaunay tetrahedral mesh generation and refinement. *Finite Elements in Analysis and Design*, 46(1-2):33–46, 2010a. ISSN 0168-874X.
- H. Si. Tetgen: A quality tetrahedral mesh generator and three-dimensional Delaunay triangulator. <http://tetgen.berlios.de/>, 2010b. URL <http://tetgen.berlios.de/>. as accessed on April 6th, 2010.
- F. L. Stazi, E. Budyn, J. Chessa, and T. Belytschko. An extended finite element method with higher-order elements for curved cracks. *Computational Mechanics*, 31(1):38–48, 2003.
- J. L. Steger, F. C. Dougherty, and J. A. Benek. A Chimera grid scheme. In K. N. Ghia and U. Ghia, editors, *Advances in Grid Generation*, volume 5, pages 59–69, New York, 1983. American Society of Mechanical Engineers, FED.
- E. Stein, R. de Borst, and T. J. R. Hughes, editors. *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd, 2004.
- R. Stenberg. On some techniques for approximating boundary conditions in the finite element method. *Journal of Computational and Applied Mathematics*, 63(1-3):139–148, 1995.
- T. Strouboulis, K. Copps, and I. M. Babuška. The generalized finite element method: an example of its implementation and illustration of its performance. *International Journal for Numerical Methods in Engineering*, 47(8):1401–1417, 2000.

- N. Sukumar, D. L. Chopp, N. Moës, and T. Belytschko. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering*, 190(46-47):6183–6200, 2001.
- M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187(1):110 – 136, 2003. ISSN 0021-9991.
- M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, 1994.
- A. Tabarraei and N. Sukumar. Extended finite element method on polygonal and quadtree meshes. *Computer Methods in Applied Mechanics and Engineering*, 197(5):425–438, 2008. ISSN 0045-7825.
- T. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: I. the concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94(3):339–351, 1992a. ISSN 0045-7825.
- T. Tezduyar, M. Behr, S. Mittal, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: II. computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Computer Methods in Applied Mechanics and Engineering*, 94(3):353–371, 1992b. ISSN 0045-7825.
- T. E. Tezduyar. Interface-tracking and interface-capturing techniques for finite element computation of moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 195(23-24):2983–3000, 2006.
- T. E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190(3-4):411–430, 2000. ISSN 0045-7825.
- T. E. Tezduyar and S. Sathe. Modelling of fluid-structure interactions with the space-time finite elements: Solution techniques. *International Journal for Numerical Methods in Fluids*, 54(6-8):855–900, 2007.
- T. E. Tezduyar, S. Mittal, S. E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95(2):221 – 242, 1992c. ISSN 0045-7825.

- Y.-H. Tseng and J. H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2):593–623, 2003. ISSN 0021-9991.
- R. van Loon, P. D. Anderson, F. P. Baaijens, and F. N. van de Vosse. A three-dimensional fluid-structure interaction method for heart valve modelling. *Comptes Rendus Mecanique*, 333(12):856–866, 2005.
- R. van Loon, P. Anderson, and F. van de Vosse. A fluid-structure interaction method with solid-rigid contact for heart valve dynamics. *Journal of Computational Physics*, 217(2):806–823, 2006. ISSN 0021-9991.
- J. A. Viecelli. A method for including arbitrary external boundaries in the mac incompressible fluid computing technique. *J. Comput. Phys.*, 4(4):543–551, 1969. ISSN 0021-9991.
- J. A. Viecelli. A computing method for incompressible flows bounded by moving walls. *Journal of Computational Physics*, 8(1):119–143, 1971. ISSN 0021-9991.
- G. J. Wagner, S. Ghosal, and W. K. Liu. Particulate flow simulations using lubrication theory solution enrichment. *International Journal for Numerical Methods in Engineering*, 56(9):1261–1289, 2003.
- W. A. Wall. *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen*. PhD thesis, Universität Stuttgart, Institut für Baustatik, 1999.
- W. A. Wall, D. P. Mok, and E. Ramm. Partitioned analysis approach of the transient coupled response of viscous fluids and flexible structures. In W. Wunderlich, editor, *Solids, Structures and Coupled Problems in Engineering, Proc. ECCM '99*, Munich, 1999.
- W. A. Wall, A. Gerstenberger, P. Gamnitzer, C. Förster, and E. Ramm. Large deformation fluid-structure interaction – advances in ALE methods and new fixed grid approaches. In H.-J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction: Modelling, Simulation, Optimisation*, volume 53 of *LNCSE*, pages 195–232. Springer Verlag, 2006.
- W. A. Wall, P. Gamnitzer, and A. Gerstenberger. Fluid-structure interaction approaches on fixed grids based on two different domain decomposition ideas. *International Journal of Computational Fluid Dynamics*, 22(6):411–427, 2008.
- X. Wang and W. K. Liu. Extended immersed boundary method using FEM and RKPM. *Computer Methods in Applied Mechanics and Engineering*, 193(12-14):1305–1321, 2004.

- Z. J. Wang and V. Parthasarathy. A fully automated Chimera methodology for multiple moving body problems. *International Journal for Numerical Methods in Fluids*, 33(7):919–938, 2000.
- N. Watanabe, H. Kataoka, T. Yasuda, and S. Takatani. Dynamic deformation and recovery response of red blood cells to a cyclically reversing shear flow: Effects of frequency of cyclically reversing shear flow and shear stress level. *Biophys. J.*, 91(5):1984–1998, 2006.
- N. Watanabe, Y. Arakawa, A. Sou, H. Kataoka, K. Ohuchi, T. Fujimoto, and S. Takatani. Deformability of human red blood cells exposed to a uniform shear stress as measured by a cyclically reversing shear flow generator. *Physiological Measurement*, 28(5):531, 2007.
- B. I. Wohlmuth. *Discretization Methods and Iterative Solvers Based on Domain Decomposition*. Springer, 2001.
- Z. Yu. A DLM/FD method for fluid/flexible-body interactions. *Journal of Computational Physics*, 207(1):1–27, 2005.
- L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193(21-22):2051–2067, 2004.
- H. Zhou and C. Pozrikidis. Deformation of liquid capsules with incompressible interfaces in simple shear flow. *Journal of Fluid Mechanics Digital Archive*, 283(-1):175–200, 1995.
- O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and *a posteriori* error estimates. part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, 1992a.
- O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and *a posteriori* error estimates. part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33(7):1365–1382, 1992b.
- A. Zilian and T.-P. Fries. A localized mixed-hybrid method for imposing interfacial constraints in the extended finite element method (XFEM). *International Journal for Numerical Methods in Engineering*, 79(6):733–752, 2009.
- A. Zilian and A. Legay. The enriched space-time finite element method (EST) for simultaneous solution of fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 75(3):305–334, 2008.
- S. Zlotnik and P. Díez. Hierarchical X-FEM for  $n$ -phase flow ( $n > 2$ ). *Computer Methods in Applied Mechanics and Engineering*, 198(30-32):2329–2338, 2009. ISSN 0045-7825.

- F. Örley. Analyse der numerischen Integration für XFEM-Anwendungen (Analysis of the numerical integration for XFEM applications). Technical report, Institute for Computational Mechanics, Technische Universität München, 2009.